Lecture Notes:
Selected Topics in Discrete Structures

Ulf Nilsson

Dept of Computer and Information Science
Linköping University
581 83 Linköping, Sweden

ulfni@ida.liu.se

2004-03-09

Contents

CHAPTER 1

Ordered sets

1.1. Basic notions

We briefly summarize basic notions and notation used later on in these notes. For a more elaborate and verbose exposition, see e.g. Grimaldi [**Gri00**].

The set of all natural numbers $\{0, 1, 2, \ldots\}$ is denoted $\mathbb{N}$. The set of all integers is denoted $\mathbb{Z}$, and the subset of all positive integers is denoted $\mathbb{Z}^+$. The rational numbers are denoted by $\mathbb{Q}$ and the real numbers by $\mathbb{R}$. The cardinality of a set A is denoted $|A|$.

By $A \times B$ we mean the Cartesian product of two sets A and B. That is, the set $\{(a, b) \mid a \in A \wedge b \in B\}$. We generalize this to arbitrary finite products $A_1 \times \ldots \times A_n$. When all A_i equal A we write simply A^n $(n \geq 0)$. The elements of A^n are referred to as n-tuples. For instance, the triple $(1, 4, 3)$ is an element of $\mathbb{N}^3$.

By a (finite) *string* (or *word*) over some alphabet Σ we mean an element u in Σ^n, for some $n \in \mathbb{N}$. The length of $u \in \Sigma^n$ is n. The set of all finite-length strings is denoted Σ^* and is defined as

$$\Sigma^* := \bigcup_{i \in \mathbb{N}} \Sigma^i.$$

The *empty* string (the only element in Σ^0) is denoted ϵ. Given two strings $u, v \in \Sigma^*$ we write the *concatenation* of u and v as uv. The length of uv is the sum of the lengths of u and v.

A binary relation R on A and B is a subset of $A \times B$. When $A = B$ we say simply that R is a relation on A. If $(a, b) \in R$ we say that a is related to b. We usually write $R(a, b)$ or $a\ R\ b$ when a is related to b. A (non-binary) relation is simply a subset of the Cartesian product $A_1 \times \ldots \times A_n$ $(n \geq 0)$.

A binary relation $R \subseteq A \times A$ is said to be

- *reflexive* iff $R(x, x)$ for every $x \in A$.
- *irreflexive* iff $R(x, x)$ for no $x \in A$.
- *antisymmetric* iff $x = y$ whenever $R(x, y)$ and $R(y, x)$.
- *symmetric* iff $R(x, y)$ whenever $R(y, x)$.
- *transitive* iff $R(x, z)$ whenever $R(x, y)$ and $R(y, z)$.

The *identity relation* on A, i.e. the relation such that $R(x, y)$ iff $x = y$ and $x \in A$, is denoted $\textsc{id}_A$. The *composition* $R_1 \circ R_2$ of two binary relations $R_1 \subseteq A \times B$ and $R_2 \subseteq B \times C$ is a binary relation on $A \times C$ defined by

$$R_1 \circ R_2 := \{(a, c) \in A \times C \mid \exists b \in B\ (R_1(a, b) \text{ and } R_2(b, c))\}.$$

The identity relation acts as left and right identity for relational composition; if $R \subseteq A \times B$ then $\textsc{id}_A \circ R = R \circ \textsc{id}_B = R$. We adopt the standard notation for iterated composition of a relation $R \subseteq A \times A$. Hence[1]

$$\begin{array}{lll} R^0 & := & \mathrm{ID}_A, \\ R^{n+1} & := & R^n \circ R \quad (n \in \mathbb{N}), \\ R^+ & := & \bigcup_{n \in \mathbb{Z}^+} R^n, \\ R^* & := & \bigcup_{n \in \mathbb{N}} R^n. \end{array}$$

We refer to R^+ as the *transitive closure* of R, and R^* as the *reflexive and transitive closure* of R.

Example 1.1. A *transition system* is a pair $(C, \Rightarrow)$ where C is a set of *configurations*, and $\Rightarrow \subseteq C \times C$ is a so-called *transition relation*. Transition systems provide abstractions of computations; a step-wise process where we move from one configuration to the next as described by the transition relation: $c_0 \Rightarrow c_1 \Rightarrow c_2 \Rightarrow \ldots$ The reflexive and transitive closure of $\Rightarrow$, that is $\Rightarrow^*$, expresses reachability: if $c_0 \Rightarrow^* c_n$ then c_n is reachable, in zero or more steps, from c_0.

A transition system may also be equipped with a set of initial configurations, a set of terminal configurations, and occasionally also with labeled transitions (the label typically modeling an event firing the transition, or fired by the transition). In case of labeled transitions the transition relation is ternary, instead of binary, and often written as $c \overset{a}{\Rightarrow} c'$. □

By the notation $A \to B$ we mean the space of all (total) *functions* from A to B. A function $f \colon A \to B$ is a relation on $A \times B$ with the property that each $a \in A$ is related to exactly one element $f(a)$ in B. Instead of writing $(a, b) \in f$ we either write $(a \mapsto b) \in f$ or use the standard notation $f(a) = b$. The *graph* of a function is the set of all tuples belonging to the function (viewed as a set). For instance, the graph of the factorial function looks as follows

$$\{0 \mapsto 1, 1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 6, 4 \mapsto 24 \ldots\}.$$

We say that a set $B \subseteq A$ is *closed* under $f \colon A \to A$ iff $f(x) \in B$ for all $x \in B$, or put alternatively if $f(B) \subseteq B$. The notion of closedness extends in the natural way to n-ary functions $f \colon A^n \to A$.

Example 1.2. Consider subsets of Σ^*, i.e. all sets of finite strings over some alphabet Σ, or *languages* as we usually refer to them. The set of all regular languages is closed under complementation; for any regular language $L \subseteq \Sigma^*$ we have that its complement $\Sigma^* \setminus L$ is regular. Regular languages are also closed under intersection and union. □

A Cartesian product A^n may be seen as the space of all functions from $\{0, \ldots, n-1\}$ to A. Hence, the n-tuple $(a_0, \ldots, a_{n-1})$ may be seen as the function $\{0 \mapsto a_0, \ldots, n-1 \mapsto a_{n-1}\}$ and vice versa. For instance, $(5, 4, 2) \in \mathbb{N}^3$ is isomorphic to $\{0 \mapsto 5, 1 \mapsto 4, 2 \mapsto 2\}$. The function space $\mathbb{N} \to A$ can thus be thought of as an infinite product "A^∞", but for reasons to be explained later we usually denote this by A^ω.

The set of all subsets of a set A is called the *powerset* of A and is denoted 2^A. The elements of 2^A may be seen as functions from A to a binary set, e.g. $\{0, 1\}$. For instance, if A equals $\{a, b, c\}$ then $\{b, c\} \in 2^A$ can be seen as the Boolean function $\{a \mapsto 0, b \mapsto 1, c \mapsto 1\}$, and $\emptyset \in 2^A$ can be seen as the Boolean function

[1] Notice that the notation is ambiguous; A^n can denote both an n-fold Cartesian product of a set, or an n-fold composition of a relation A.

$\{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$. (It is sometimes customary to write B^A as an alternative to the function space $A \to B$ which explains in part the notation 2^A, i.e. functions from A to a set of cardinality 2.)[2]

EXAMPLE 1.3. A Boolean valuation (interpretation or model) is a mapping from an alphabet of propositional variables Var to a binary set $\mathbf{Bool} := \{0,1\}$. If $\mathit{Var} := \{x,y,z\}$ then $\mathit{Var} \to \mathbf{Bool}$ is the set of all Boolean functions from Var to $\mathbf{Bool}$. There are obviously $2^{|\mathit{Var}|} = 8$ such functions, for instance

$$\begin{array}{lll}
\sigma_0 & := & \{x \mapsto 0, y \mapsto 0, z \mapsto 0\} \\
\sigma_1 & := & \{x \mapsto 1, y \mapsto 0, z \mapsto 0\} \\
\sigma_2 & := & \{x \mapsto 0, y \mapsto 1, z \mapsto 0\} \\
\sigma_3 & := & \{x \mapsto 1, y \mapsto 1, z \mapsto 0\} \\
\text{etc.} & &
\end{array}$$

We may equivalently represent e.g. σ_3 by the set $\{x,y\} \in 2^{\mathit{Var}}$. We refer to the latter as the *set-representation* of a Boolean valuation or interpretation. □

1.2. Basic orderings

We next consider some well-known and useful relations. In particular relations which allow us to order (in an intuitive sense) elements.

DEFINITION 1.4. A relation $R \subseteq A \times A$ is called a *preorder* (or *quasi ordering*) if it is reflexive and transitive. □

EXAMPLE 1.5. The standard inequality $\leq$ on the natural numbers is a preorder. So is the standard subset relation $\subseteq$ on every powerset 2^A. □

EXAMPLE 1.6. Define $\leq_7 \subseteq \mathbb{N} \times \mathbb{N}$ as follows

$$(x \leq_7 y) \text{ iff } (x \bmod 7) \leq (y \bmod 7)$$

Then $\leq_7$ is a preorder since it is both reflexive and transitive. Note that $\leq_7$ is not antisymmetric since e.g. $6 \leq_7 13$ and $13 \leq_7 6$ but $6 \neq 13$. □

DEFINITION 1.7. A preorder $R \subseteq A \times A$ is called a *partial order* if it is also antisymmetric. □

EXAMPLE 1.8. The relation $\leq$ on the natural numbers is a partial order, and so is $\subseteq$ on 2^A. □

EXAMPLE 1.9. The relation “divides” on $\mathbb{Z}^+$ is a partial order; any positive integer divides itself (reflexivity); if x and y divide each other, then $x = y$ (antisymmetry), and if x divides y and y divides z, then by necessity x divides z (transitivity). □

EXAMPLE 1.10. Let Σ be an alphabet, and consider Σ^*; i.e. the set of all finite strings over Σ. If $u, v \in \Sigma^*$ then uv denotes the concatenation of u and v. Now let $\trianglelefteq \subseteq \Sigma^* \times \Sigma^*$ defined by

$$u \trianglelefteq v \text{ iff there is a } w \in \Sigma^* \text{ such that } uw = v.$$

Then $\trianglelefteq$ is a partial order, usually called the *prefix* order. □

[2] The other reason is of course that $|2^A| = 2^{|A|}$ for finite sets A. This holds also for function spaces; there are $|B|^{|A|}$ functions from A to B (for finite sets A, B).

Example 1.11. Let $A \twoheadrightarrow B$ denote the space of all partial maps from A to B. That is, any $f \subseteq A \times B$ such that if $(x, y) \in f$ and $(x, z) \in f$ then $y = z$. A partial map can be viewed as an underspecificed total function; in fact, we may order partial maps depending on how much *information* they convey. For instance, consider the function space $\mathbb{N} \twoheadrightarrow \mathbb{N}$ and the four partial functions

$$\begin{array}{lll} \sigma_1 & := & \{(0 \mapsto 1), (1 \mapsto 1)\} \\ \sigma_2 & := & \{(0 \mapsto 1), (1 \mapsto 1), (2 \mapsto 2)\}. \\ \sigma_3 & := & \{(0 \mapsto 1), (1 \mapsto 1), (2 \mapsto 2), (3 \mapsto 6)\}. \\ \sigma_4 & := & \{(0 \mapsto 1), (1 \mapsto 1), (2 \mapsto 1), (3 \mapsto 1)\}. \end{array}$$

Then σ_2 conveys more information than σ_1. Similarly σ_3 contains more information than both σ_2 and σ_1. Now if we compare σ_4 and σ_2 we see that σ_4 is more *defined* than σ_2, but it does not contain more *information* than σ_2; they convey *incomparable* information since $\sigma_2(2) = 2 \neq \sigma_4(2) = 1$. Formally we may define our ordering of partial maps (often refered to as the *information ordering*) simply as set inclusion on the graphs of the functions. That is, given $\sigma \colon A \twoheadrightarrow B$ and $\sigma' \colon A \twoheadrightarrow B$

$$\sigma \leq \sigma' \text{ iff } \sigma \subseteq \sigma'.$$

As we shall see later the information ordering is very important when formally defining e.g. functions with infinite domains; the partial maps $\sigma_1, \sigma_2, \sigma_3$ are examples of increasingly better approximations of the factorial function. The information ordering is also important when defining semantics of programming languages. □

A partial order is of course always a preorder, but the converse does not generally hold. However, a preorder $\preceq \subseteq A \times A$ which is not antisymmetric induces a partial order if lifted to a relation on equivalence classes. Let

$$x \equiv y \text{ iff } x \preceq y \wedge y \preceq x$$

and define

$$[x] \preceq_{\equiv} [y] \text{ iff } x \preceq y.$$

Then $\preceq_{\equiv}$ is a partial order (prove this). We sometimes say that $\preceq$ modulo $\equiv$ is a partial order.

Example 1.12. Consider the set of propositional formulas F induced by a finite set *Var* of propositional variables:

$$\begin{array}{lll} F & ::= & \mathit{Var} \\ F & ::= & \neg F \mid (F \wedge F) \mid (F \vee F) \mid (F \rightarrow F) \end{array}$$

We say that an interpretation (i.e. a Boolean valuation) σ is a model of a Boolean formula F if F is true in σ, and write $\mathrm{Mod}(F)$ for the set of all models of F.

Now consider F under the entailment ordering: $F_1 \models F_2$ iff every model of F_1 is also a model of F_2, or put equivalently iff $\mathrm{Mod}(F_1) \subseteq \mathrm{Mod}(F_2)$. The result is a preorder. The relation $\models$ is clearly reflexive and transitive, but not antisymmetric since e.g. $(\neg x \vee y) \models (x \rightarrow y)$ and $(x \rightarrow y) \models (\neg x \vee y)$. On the other hand, we have the following (logical) equivalence relation

$$\begin{array}{rll} F_1 \Leftrightarrow F_2 & \text{iff} & F_1 \text{ and } F_2 \text{ have the same set of models} \\ & \text{iff} & F_1 \models F_2 \text{ and } F_2 \models F_1. \end{array}$$

If we consider $\models$ modulo $\Leftrightarrow$ then we have a partial order. □

We sometimes encounter an alternative notion of partial order, sometimes called a *strict* partial order to distinguish it from the previous notion:

DEFINITION 1.13. A relation $R \subseteq A \times A$ which is irreflexive and transitive is called a *strict partial order.* □

If $R \subseteq A \times A$ is a partial order then $R \backslash \textsc{id}_A$ is a strict partial order (where $\textsc{id}_A$ is the identity relation on A). Note that a strict partial order is always antisymmetric (prove this).

EXAMPLE 1.14. The relation $<$ on $\mathbb{N}$ and the relation $\subset$ on 2^A are examples of strict partial orders. □

NOTATION: From now on we normally use relation symbols like $\leq$, $\preceq$, $\sqsubseteq$ for non-strict partial orders. In such cases we occasionally write $y \geq x$ as an alternative to $x \leq y$, and if $\leq$ is a partial order then $<$ refers to the strict version of $\leq$, i.e. $\leq \backslash \textsc{id}_A$, assuming that $\leq \subseteq A \times A$. As usual the notation $x \not\leq y$ means that x is not related to y. We say that two elements are *comparable* whenever $x \leq y$ or $y \leq x$; and *incomparable* otherwise. We write $x \mid\mid y$ when x and y are incomparable (assuming that the order is known).

DEFINITION 1.15. If $\leq \subseteq A \times A$ is a partial order then the pair $(A, \leq)$ is called a *partially ordered set*, or *poset.* □

By an *ordered set* we henceforth mean a poset (strict or non-strict).

DEFINITION 1.16. A poset $(A, \leq)$ is called a *total order* (or *chain*, or *linear order*) if either $a \leq b$ or $b \leq a$ for all $a, b \in A$. □

DEFINITION 1.17. A poset $(A, \leq)$ is called an *anti-chain* if $x \leq y$ implies $x = y$, for all $x, y \in A$. □

We use the terms chain and anti-chain also in the context of strict partial orders. A (strict) chain is a strict partial order $(A, <)$ where either $x < y$ or $y < x$ when $x \neq y$, for all $x, y \in A$. A (strict) anti-chain is a strict partial order $(A, <)$ where $x \mid\mid y$ for all $x, y \in A$.

1.3. Constructing orders

We survey some useful techniques for constructing posets from existing, usually simpler, posets. However first we consider the opposite; let $\mathcal{A} := (A, \leq)$ be a poset and let $B \subseteq A$. Then $\mathcal{B} := (B, \preceq)$ is called the *poset induced by* $\mathcal{A}$ if

$$x \preceq y \text{ iff } x \leq y \text{ for all } x, y \in B.$$

We prove that $\mathcal{B}$ is indeed a poset.

THEOREM 1.18. If $\mathcal{A}$ is a poset and $\mathcal{B}$ is induced by $\mathcal{A}$, then $\mathcal{B}$ is a poset. □

PROOF. First consider reflexivity: Let $x \in B$. Then $x \in A$ and $x \leq x$ in since $\mathcal{A}$ is a poset. Hence $x \preceq x$. Second, consider antisymmetry: Assume $x, y \in B$ and $x \preceq y \preceq x$; hence, $x \leq y \leq x$. Since $\mathcal{A}$ is antisymmetric $x = y$. Transitivity is shown similarly. □

In most cases we write simply that $(B, \leq)$ is the poset induced by $(A \;\leq)$ although $\leq$ in the former is different from $\leq$ in the latter (unless of course $A = B$).

We next consider so-called *componentwise orderings.*

THEOREM 1.19. Let $(A, \leq)$ be a poset, and consider a relation $\preceq$ on $A \times A$ defined as follows

$$(x_1, y_1) \preceq (x_2, y_2) \text{ iff } x_1 \leq x_2 \wedge y_1 \leq y_2.$$

Then $(A \times A, \preceq)$ is a poset. □

The proof is left as an exercise.

Componentwise orderings can be generalized to arbitrary (finite) Cartesian products. In principle it is possible to extend the notion also to infinite products, but we usually refer to them as *pointwise orderings*:

THEOREM 1.20. Let $(A, \leq)$ be a poset, and consider a relation $\preceq$ on $(B \to A)$ defined as follows

$$\sigma_1 \preceq \sigma_2 \text{ iff } \sigma_1(x) \leq \sigma_2(x) \text{ for all } x \in B.$$

Then $(B \to A, \preceq)$ is a poset. □

The proof is similar to componentwise orderings.

EXAMPLE 1.21. Given a set of Boolean variables *Var* a (Boolean) valuation is a mapping $\sigma\colon \mathit{Var} \to \{0, 1\}$ where we assume the natural ordering $\leq$ on Boolean values. In the pointwise ordering $\sigma_1 \preceq \sigma_2$ iff $\sigma_1(x) \leq \sigma_2(x)$, for all $x \in \mathit{Var}$. For instance, $\{x \mapsto 1, y \mapsto 0, z \mapsto 0\} \preceq \{x \mapsto 1, y \mapsto 0, z \mapsto 1\}$. □

We finally consider so-called *lexicographical orderings*. Let $\Sigma = \{a_1, \ldots, a_n\}$ be a finite alphabet under some strict total ordering $a_1 < \ldots < a_n$. Let Σ^* be the set of all finite (possibly empty) strings over Σ and define $x_1 \ldots x_i \sqsubset y_1 \ldots y_j$ to hold iff

- $i < j$ and $x_1 \ldots x_i = y_1 \ldots y_i$, or
- there is some $k < i$ such that $x_{k+1} < y_{k+1}$ and $x_1 \ldots x_k = y_1 \ldots y_k$.

This is the standard total ordering of words that we encountered e.g. in dictionaries. Note that $\sqsubset$ is a strict total order.

EXAMPLE 1.22. Let $\Sigma = \{a, b, c\}$ with the standard total ordering. Then e.g.

$$\epsilon \sqsubset a \sqsubset aa \sqsubset ab \sqsubset abb \sqsubset ac \sqsubset \ldots$$

As usual ϵ denotes the empty string. □

1.4. Well-founded relations and well-orders

We next introduce the notion of well-founded relations which provides the basis of many computer science notions; both in the formalization of computation and as a means of proving properties of programs. Last but not least, it also provides a basis for unambiguous definition of (infinite) sets, functions and relations.

We first introduce the following auxiliary notions.

DEFINITION 1.23. Consider a relation $R \subseteq A \times A$. An element $a \in A$ is called *R-minimal* (or simply minimal when R is clear from the context) if there is no $b \in A$ such that $b\ R\ a$. Similary, $a \in A$ is called *maximal* if there is no $b \in A$ such that $a\ R\ b$. □

DEFINITION 1.24. An element $a \in A$ is called *least* if $a\ R\ b$ for all $b \in A$; it is called *greatest* if $b\ R\ a$ for all $b \in A$. □

The least element of a set (if it exists) is sometimes denoted $\bot$ and the greatest element is denoted $\top$.

It should be pointed out that minimal and least elements do not coincide. In particular, a partial order can have no minimal element since it is reflexive, but it can have 0 or 1 least elements; that is, least elements are unique if they exist. For strict partial orders the situation is almost the opposite; there can be no least element, since the relation is irreflexive, but there can be any number of minimal elements (including 0).

Example 1.25. The poset $(\mathbb{N}, \leq)$ has no greatest element, but it does have a least element (namely 0). Note that $(\mathbb{N}, \leq)$ has neither a maximal nor, more surpringly, a minimal element since $\leq$ is reflexive (and $0 \leq 0$). Note also that $(\mathbb{N}, <)$ has a minimal element (namely 0), but no least (nor maximal, nor greatest) element. □

Example 1.26. The poset $(2^A, \subseteq)$ has a least and greatest element, namely $\emptyset$ and A. But it does not have any minimal or maximal elements. For $(2^A, \subset)$ there is a unique minimal element, $\emptyset$, and a unique maximal element, A.

The strict poset $(2^A \setminus \{\emptyset\}, \subset)$ has $|A|$ minimal elements; namely all singleton subsets of A. □

Example 1.27. Consider the poset $(\mathbf{Bool}, \leq)$. The set $\mathit{Var} \to \mathbf{Bool}$ under the pointwise ordering is a poset; the valuation σ such that $\sigma(x) = 1$ for all $x \in \mathit{Var}$ is the greatest element and the valuation such that $\sigma(x) = 0$ for all $x \in \mathit{Var}$ is the least element. □

Note that in some books an element $a \in A$ is said to be minimal if there is no $b \neq a$ in A such that $b\ R\ a$. In that case a least element is always minimal if (A, R) is a poset. Similarly, an element $a \in A$ is sometimes said to be least if $a\ R\ b$ for all $b \neq a$ in B. In the following we rely on the first definitions unless otherwise stated.

Definition 1.28. A relation $R \subseteq A \times A$ is said to be *well-founded* if every non-empty subset of A contains an R-minimal element. □

If R is a well-founded relation on A we sometimes say that (A, R) is a *well-founded set*. And when R is clear from the context, we sometimes say simply that A is a well-founded set. Note that we make no special assumptions about R; it has to be irreflexive since otherwise there must be some singleton set $\{x\} \subseteq A$ which has no minimal element, but we do not require R to be transitive (although in practice it often is).

We have the following important instance of well-founded relations:

Definition 1.29. A strict total order $(A, <)$ which is well-founded is called a well-order. □

Example 1.30. The relation $<$ on $\mathbb{N}$ is a well-order, while $<$ on $\mathbb{Z}$ is not, since $\mathbb{Z}$ has no minimal element. □

It follows that every subset of a well-order, including the set itself, has a unique minimal element. Moreover, the following theorem is easy to prove.

Theorem 1.31. Any subset $(B, <)$ of a well-order $(A, <)$ is a well-order. □

We will often write $x_0 < x_1 < x_2 < \ldots$ for a well-order $(A, <)$ where x_0 is the (unique) minimal element in A, and x_1 is the (unique) minimal element of $A \setminus \{x_0\}$ etc.

EXAMPLE 1.32. The following are examples of well-orders

- The natural numbers under $<$.
- The set Σ^* under the lexicographical order $\sqsubset$ on a finite alphabet Σ.

The following relations are not well-ordered

- The rational numbers $\mathbb{Q}$ under $<$, since e.g.
$$\ldots < \frac{1}{4} < \frac{1}{3} < \frac{1}{2} < 1$$
has no minimal element.
- The set Σ^* under the inverse lexicographical ordering $\sqsubset^{-1}$ (written $\sqsupset$), since e.g.
$$\ldots \sqsupset aaa \sqsupset aa \sqsupset a$$
has no minimal element.

□

LEMMA 1.33. If every non-empty subset of $(A, <)$ has a unique minimal element then $<$ is transitive. □

PROOF. Assume that $x < y$ and $y < z$. Assume also that $x \not< z$. Since $\{x, z\}$ must contain a unique minimal element it follows that $z < x$. By the inital assumption $x < y < z < x$, which means that $\{x, y, z\}$ contains no minimal element, contradicting the antecendent of the lemma. Hence $x < z$. □

THEOREM 1.34. The structure $(A, <)$ is a well-order iff every non-empty subset of A has a unique minimal element. □

PROOF. The direction $\Rightarrow$ follows trivially. To prove $\Leftarrow$ we assume that every non-empty subset of A has a unique minimal element. We prove first that $<$ is a strict total order. Since $<$ is transitive by Lemma 1.33, it remains to be shown that it is also irreflexive and a total order.

- Assume that $<$ is not irreflexive. Then there is some $x \in A$ such that $x < x$. But then $\{x\}$ contains no minimal element. Contradiction!
- Assume that $<$ is not a total order. That is, that there are $x \neq y$ such that $x \mid\mid y$. But then $\{x, y\}$ contains two minimal elements. Contradiction!

Hence, $<$ is a strict total order and since every non-empty subset of a contains a minimal element $(A, <)$ is a well-order. □

The following notions are important e.g. in the verification of termination of programs and discrete dynamic systems:

DEFINITION 1.35. Let $(A, \leq)$ be a poset. A well-order $x_0 < x_1 < \ldots$ where $\{x_0, x_1, \ldots\} \subseteq A$ is called an *ascending chain in A*. □

EXAMPLE 1.36. The well-order $\emptyset \subset \{0\} \subset \{0, 1\} \subset \{0, 1, 2\} \subset \ldots$ is an ascending chain in $2^{\mathbb{N}}$. □

DEFINITION 1.37. A poset in which every non-empty chain has a maximal element is called *Noetherian*, and is said to satisfy the *ascending chain condition.* Dually, a poset where every non-empty chain has a minimal element (i.e. where every chain is a well-order) is said to satisfy the descending chain condition. □

We have the following equivalent characterization of well-founded relations.

Theorem 1.38. A relation $< \subseteq A \times A$ is well-founded iff $(A, <)$ contains no infinite descending chains $\ldots < x_2 < x_1 < x_0$ (i.e. $(A, <)$ satisfies the descending chain condition). □

Proof. $(\Rightarrow)$: Assume that $(A, <)$ is well-founded and that there exists an infinite descending chain $\ldots < x_2 < x_1 < x_0$. Then $\{x_0, x_1, x_2, \ldots\}$ contains no minimal element contradicting the assumption that $<$ is well-founded.

$(\Leftarrow)$: First assume that $(A, <)$ contains no infinite descending chain. Secondly assume that $(A, <)$ is not well-founded. Hence there is some non-empty $B \subseteq A$ which contains no minimal element. Now B must contain a descending chain $y_n < \ldots < y_0$ where $(n \geq 0)$; since B contains no minimal element there must be some $y_{n+1} \in B$ such that $y_{n+1} < y_n$ which implies that all finite descending chains can be extended in infinity. Hence there must be an infinite descending chain contradicting our initial assumption. □

The following examples illustrate some uses of well-founded sets.

Example 1.39. Consider an inductive definition of a language, e.g. the set of all propositional formulas over some finite alphabet of propositional variables *Var*:

$$\begin{array}{lll} F & ::= & \mathit{Var} \\ F & ::= & \neg F \mid (F \wedge F) \mid (F \vee F) \mid (F \rightarrow F) \end{array}$$

Let $\prec$ be the "proper subformula" relation; $G \prec F$ iff G is a proper subformula of F. For instance, x, y, $\neg x$, $\neg y$ and $\neg x \vee y$ are all proper subformula of $(\neg x \vee y) \vee \neg y$. Then $\prec$ is a well-founded relation. (Prove this!) □

Example 1.40. Consider a transition system $(C, \Rightarrow, I)$ with an initial set $I \subseteq C$ of configurations. Let $\prec \subseteq C^+ \times C^+$ be defined as follows[3]

$$c_1 \ldots c_n \prec c_1 \ldots c_n c_{n+1} \text{ iff } c_n \Rightarrow c_{n+1}$$

Now let the set of traces T of $(C, \Rightarrow, I)$ be the smallest set of words such that

- if $c \in I$ then $c \in T$,
- if $t \in T$ and $t \prec t'$ then $t' \in T$.

Then $\prec$ is a well-founded relation on T. □

We conclude this chapter defining the notions of down-sets, or order ideals, and the dual notions of up-sets, and order filters.

Definition 1.41. Let $(A, \leq)$ be a poset. A set $B \subseteq A$ is called a *down-set* (or an *order ideal*) iff

$$y \in B \text{ whenever } x \in B \text{ and } y \leq x.$$

A set $B \subseteq A$ induces a down-set, denoted $B{\downarrow}$,

$$B{\downarrow} := \{x \in A \mid \exists y \in B, x \leq y\}.$$

By $\mathcal{O}(A)$ we denote the set of all down-sets in A,

$$\{B{\downarrow} \mid B \subseteq A\}.$$

□

A notion of *up-set*, also called *order filter*, is defined dually.

[3] As usual C^+ denotes the set of all non-empty and finite words (i.e. sequences) of configurations.

Exercises

1.1 Draw the Hasse diagram of $\{x, y, z\} \rightarrow$ **Bool** under the ordering in Example 1.3. Compare the diagram to the Hasse diagram of the poset $(2^{\{x,y,z\}}, \subseteq)$.

1.2 Show that the prefix ordering in Example 1.10 is a partial order.

1.3 Let $(A, \leq)$ be a preorder, and let $x \equiv y$ iff $x \leq y$ and $y \leq x$. Prove that $\leq$ lifted to the equivalence classes of $\equiv$ is a partial order.

1.4 Prove that a strict partial order is always antisymmetric.

1.5 Let (A, R_1) be a well-founded set, and let $R_2 \subseteq R_1$. Show that (A, R_2) is well-founded.

1.6 Let (A, R) be well-founded set. Prove that (A, R^+) also is well-founded.

1.7 Prove Theorem 1.19.

1.8 Prove Theorem 1.20.

1.9 Prove that the subformula relation $\prec$ in Example 1.39 is well-founded.

CHAPTER 2

Algebraic structures

In this chapter we study two types of partially ordered sets which are extensively used in many areas of computer science; we first consider the more general notion of lattices and complete lattices, followed by the more specialized notion of complete partial orders, or cpo's.

2.1. Lattices and complete lattices

We survey basic definitions and and fundamental properties of lattices. For more elaborate expositions, see Birkhoff [**Bir67**] or Grätzer [**Grä78**].

We first introduce the auxiliary notion of (least) upper bound and (greatest) lower bound.

Definition 2.1. Let $(A, \leq)$ be a poset and $B \subseteq A$. Then $x \in A$ is called an *upper bound* of B iff $y \leq x$ for all $y \in B$ (often written $B \leq x$ by abuse of notation). The notion of *lower bound* is defined dually. □

Note that the set of all lower bounds of $\{x\}$, or simply x, is identical to $\{x\}\!\downarrow$, i.e. the down-set of x. More generally, the set of all lower bounds of $B \subseteq A$ equals

$$\bigcap_{x \in B} \{x\}\!\downarrow.$$

Definition 2.2. Let $(A, \leq)$ be a poset and $B \subseteq A$. Then $x \in A$ is called a *least upper bound* of B iff $B \leq x$ and $x \leq y$ whenever $B \leq y$. The notion of *greatest lower bound* is defined dually. □

Least upper bounds (and greatest lower bounds) are unique, if they exist.

Definition 2.3. A *lattice* is a poset $(A, \leq)$ where every pair of elements $x, y \in A$ has a least upper bound, denoted $x \vee y$, and greatest lower bound, denoted $x \wedge y$. □

The least upper bound (abbr. lub) $x \vee y$ is sometimes called the *join* or *supremum* of x and y, and the greatest lower bound (glb) $x \wedge y$ is sometimes called the *meet* or *infimum* of x and y. Alternative notations are $\sup(x, y)$ and $\inf(x, y)$.

Figure 1 depicts two posets as Hasse diagrams. The leftmost poset is a lattice, while the rightmost is not. (Why?)

Example 2.4. The following are examples of lattices

- The set 2^A under $\subseteq$ is a lattice with least upper bound $\cup$ and greatest lower bound $\cap$.
- The set $\mathbb{Z}$ under $\leq$ is a lattice with the function *min* as greatest lower bound, and the function *max* as least upper bound.

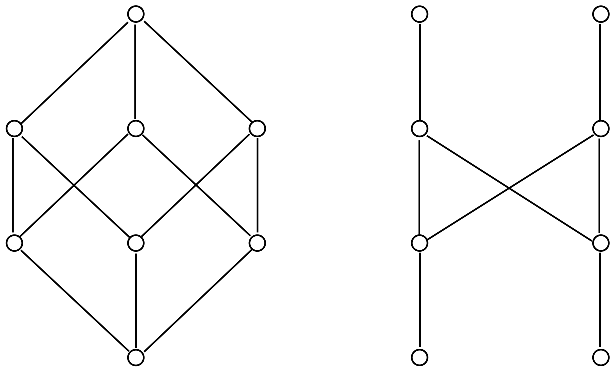


FIGURE 1. A lattice (left) and a poset which is not a lattice (right)

- The set of regular languages over some alphabet Σ is a lattice with intersection as greatest lower bound and union as least upper bound (recall that regular languages are closed under both intersection and union).

While the set of all regular languages over some alphabet is a lattice, the set of all context-free languages is not. Context-free languages *are* closed under union (if L_1 and L_2 are context-free languages then there are context-free grammars G_1 and G_2 describing them, and $L_1 \cup L_2$ can be obtained by taking the union of G_1 and G_2 after renaming of the non-terminals so that G_1 and G_2 have disjoint nonterminal alphabets apart from the start symbol). However, context-free languages are not closed under intersection; for example, both $L_1 = \{a^ib^ic^j \mid i,j \geq 1\}$ and $L_2 = \{a^ib^jc^j \mid i,j \geq 1\}$ are context-free (both can be described by context-free grammars), but their intersection $L_1 \cap L_2 = \{a^ib^ic^i \mid i \geq 1\}$ is the standard example of a language which is not context-free. □

A lattice which is closed under only one of $\wedge$ and $\vee$ is called a *semi-lattice* (join-semi-lattice or meet-semi-lattice).

A lattice involves a poset and two operations. Hence a lattice really is a structure $(A, \leq, \wedge, \vee)$. However, the two operations actually follow from $\leq$ (if they exist) and vice versa. That is, a lattice is given unambiguously either by the partial order or the two bounds. (We will discuss this in some detail in the next section.) As a consequence we sometimes say that $(A, \leq)$ is a lattice assuming implicitly the existence also of $\wedge$ and $\vee$; sometimes we say instead that $(A, \wedge, \vee)$ is a lattice assuming tacitly the ordering $\leq$.

DEFINITION 2.5. Let $(A, \leq)$ be a lattice. An element $a \in A$ is said to *cover* an element $b \in A$ iff $a > b$ and there is no $c \in A$ such that $a > c > b$. □

EXAMPLE 2.6. The element $\{0, 1\}$ covers $\{0\}$ in the lattice $(2^{\{0,1,2\}}, \subseteq)$. But it is not the only element covering $\{0\}$, since $\{0\}$ is also covered by $\{0, 2\}$. □

DEFINITION 2.7. The *length* of a poset $(A, \leq)$ (and hence lattice) is $|C| - 1$ where C is the longest chain in A. □

A poset/lattice is *finite length* (or height) if $|C|$ is a natural number.

EXAMPLE 2.8. The length of $(2^{\{0,1,2\}}, \subseteq)$ is 3, since e.g. $\emptyset \subset \{0\} \subset \{0, 1\} \subset \{0, 1, 2\}$. The length of the lattice $(2^{\mathbb{N}}, \subseteq)$ is infinite. □

We next define the notion of complete lattice, which provides an important instance of ordinary lattices.

DEFINITION 2.9. A *complete lattice* is a poset $(A, \leq)$ where every subset $B \subseteq A$ (finite or infinite) has a least upper bound $\bigvee B$ and a greatest lower bound $\bigwedge B$. The element $\bigvee A$ is called the *top* element and is usually denoted $\top$. The element $\bigwedge A$ is called the *bottom* element and is denoted $\bot$. □

Every complete lattice is a lattice since every pair of elements has a least upper and greatest lower bound, but the converse does not hold in general as illustrated by the following example.

EXAMPLE 2.10. The set of all natural numbers $\mathbb{N}$ under the standard non-strict ordering $\leq$ is a lattice; any pair of natural numbers has a least upper bound (namely the supremum of the two), and a greatest lower bound (namely the infimum of the two). However, it is not a complete lattice; any finite subset has a least upper, and greatest lower bound, but the set of all natural numbers does *not* have a least upper bound. (However, it does have a greatest lower bound.) On the other hand, if we add a top element $\top$ to the natural numbers we have a complete lattice. □

EXAMPLE 2.11. The powerset 2^A of any set A is a complete lattice under standard set inclusion $\subseteq$. Let $\{A_i\}_{i \in I} \subseteq 2^A$, then we have the least upper bound

$$\bigcup_{i \in I} A_i := \{a \mid a \text{ is a member of some } A_i\}.$$

The greatest lower bound is defined dually

$$\bigcap_{i \in I} A_i := \{a \mid a \text{ is a member of every } A_i\}.$$

□

EXAMPLE 2.12. The set of all regular languages is not a complete lattice; there is a least and greatest element, namely $\emptyset$ and Σ^*, and the union of finitely many regular languages is regular, but the infinite union is in general not regular. For instance, all of the following singleton languages are trivially regular

$$\begin{array}{lcl} L_0 & = & \{\epsilon\} \\ L_1 & = & \{ab\} \\ L_2 & = & \{aabb\} \\ L_3 & = & \{aaabbb\} \\ \text{etc.} & & \end{array}$$

Moreover, the union of any finite subset of $\{L_0, L_1, \ldots\}$ is also regular, but the infinite union, i.e. $\{a^n b^n \mid n \geq 0\}$ is a standard example of a language which is not regular (but rather context-free). □

In the following special case a lattice is trivially complete. The proof is by induction proving that any finite set of elements has a lub (glb) if any pair of elements has.

THEOREM 2.13. Any finite lattice is a complete lattice. □

We finally survey some special lattices that enjoy additional algebraic properties.

Definition 2.14. Let $(A, \leq)$ be a lattice with $\bot$ and $\top$. We say that $a \in A$ is the *complement* of $b \in A$ iff $a \vee b = \top$ and $a \wedge b = \bot$. □

It follows that the complement of $\bot$ is $\top$, and vice versa (provided that they exist, of course).

Definition 2.15. We say that a lattice is *complemented* if every element has a complement. □

The lattice of regular languages over some alphabet Σ is a complemented lattice; in this particular case each complement is unique. However, the complement of an element in a complemented lattice need not be unique (see exercises). If the the complement of all elements x is unique, it is denoted x'; hence, $x \wedge x' = \bot$ and $x \vee x' = \top$.

Definition 2.16. A lattice $(A, \leq)$ is said to be *distributive* iff $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for all $a, b, c \in A$. □

It can be shown that $\vee$ distributes over $\wedge$ iff $\wedge$ distributes over $\vee$; hence, in a distributive lattice we also have that $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ (see exercises). One can also show that in a complemented, distributive lattice the complement of each element is unique. Suppose that both b and c are complements of a, then

$$b = b \wedge \top = b \wedge (a \vee c) = (b \wedge a) \vee (b \wedge c) = \bot \vee (b \wedge c) = b \wedge c$$

Hence $b \leq c$. By an analogous argument $c \leq b$, in which case by necessity $b = c$, hence the complement of a must be unique.

Definition 2.17. A lattice $(A, \leq)$ is said to be *Boolean* iff it is complemented and distributive. □

Example 2.18. The set of all regular languages over some alphabet Σ has a top and bottom element (namely Σ^* and $\emptyset$). Moreover, every regular language has a complement (recall that regular languages are closed under complementation). Finally it can be shown that the lattice of regular languages is distributive, and hence Boolean. □

Example 2.19. Not surprisingly, Boolean algebras and Boolean lattices coincide; that is, a Boolean algebra $(B, +, \cdot, ', 0, 1)$ is a Boolean lattice with least upp bound $+$, greatest lower bound $\cdot$, complement $'$, bottom element 0 and top element 1. Recall also that a Boolean algabra is an algebraic structure satisfying the following laws,

Commutative laws:	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributive laws:	$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	$a + (b \cdot c) = (a + b) \cdot (a + c)$
Identity laws:	$a + 0 = a$	$a \cdot 1 = a$
Inverse laws:	$a + a' = 1$	$a \cdot a' = 0$

for all $a, b, c \in B$. □

Definition 2.20. Let A be a set and $B \subseteq 2^A$. If $(B, \subseteq)$ is a lattice, then we refer to it as a *lattice of sets*. If it is a complete lattice we call it a complete lattice of sets. □

Theorem 2.21. We have the following results for lattices of sets:

(1) Any lattice of sets is distributive.

(2) $(2^A, \subseteq)$ is distributive, and Boolean.

□

The proofs are left as exercises.

2.2. Lattices as algebras

Our definition of lattice is based on partially ordered sets. However, there is an equivalent algebraic definition. Consider an algebra $(A, \otimes, \oplus)$ with operations $\otimes\colon A \times A \to A$ and $\oplus\colon A \times A \to A$. The algebraic structure $(A, \otimes, \oplus)$ is a lattice if the operations satisfy the following laws, for all $a, b, c \in A$.

(L_1) Idempotency: $a \otimes a = a \oplus a = a$
(L_2) Commutativity: $a \otimes b = b \otimes a$ and $a \oplus b = b \oplus a$
(L_3) Associativity: $a \otimes (b \otimes c) = (a \otimes b) \otimes c$ and $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
(L_4) Absorption: $a \otimes (a \oplus b) = a$ and $a \oplus (a \otimes b) = a$

Now let $a \leq b$ iff $a \otimes b = a$ (or $a \oplus b = b$); then it follows that the ordered set $(A, \leq)$ is a lattice, i.e. every pair of elements $a, b \in A$ has a least upper bound (namely $a \oplus b$) and a greatest lower bound (namely $a \otimes b$). The proof of this equivalence is left as an exercise.

2.3. Complete partial orders

Lattices possess many appealing properties, but the requirement that any pair of elements, or any subset of elements, has both a lub and glb is often an unnecessarily strong requirement. A number of results in computer science rely only on the fact that all ascending chains have least upper bounds.

DEFINITION 2.22. A partial order $(A, \leq)$ is said to be *complete* if it has a bottom element $\bot$ and if each ascending chain

$$a_0 < a_1 < a_2 < \ldots$$

has a least upper bound $\bigvee\{a_0, a_1, a_2, \ldots\}$. □

EXAMPLE 2.23. The set $\mathbb{N}$ of natural numbers under $\leq$ is not a complete partial order. It does have a least element, namely 0, but no infinite ascending chain

$$n_1 < n_2 < n_3 < \ldots$$

has a least upper bound. However, $\mathbb{N}$ extended with a top element ω where $n < \omega$ for all $n \in \mathbb{N}$ is a complete partial order. (The reason for using the symbol ω should be clear in Chapter 3. For now, think of it simply as the set of all natural numbers; i.e. a synonym of $\mathbb{N}$.) □

EXAMPLE 2.24. Consider the set of finite strings Σ^* over some alphabet Σ, and the prefix ordering

$$u \trianglelefteq v \text{ iff there is a } w \in \Sigma^* \text{ such that } uw = v.$$

Every $S \subseteq \Sigma^*$ has a greatest lower bound $\bigwedge S$; namely the longest string which is a prefix of all $w \in S$ (possibly ϵ). However, not all $S \subseteq \Sigma^*$ have a least upper bound, i.e. the shortest string which every $w \in S$ is a prefix of. Even worse, not even every ascending chain $w_1 \triangleleft w_2 \triangleleft \ldots$ has a least upper bound. All finite chains have; namely the longest string in the set, but if we consider an infinite ascending chain

$$a \triangleleft aa \triangleleft aaa \triangleleft aaaa \triangleleft \ldots$$

then there is no finite string which has all strings in the chain as prefixes. Hence $(\Sigma^*, \trianglelefteq)$ is not a complete partial order. On the other hand, if we "throw in" an extra top element $\top$ greater than all finite strings then we have a complete partial order.

Instead of throwing in just *any* top element $\top$, we may define a notion of infinite strings by viewing strings as functions: for instance, consider a finite string $w := 010101 \in \mathbf{Bool}^*$. This may be viewed as a function in $\{0, \ldots, 5\} \to \mathbf{Bool}$ where $w(0) = w(2) = w(4) = 0$ and $w(1) = w(3) = w(5) = 1$. More conveniently, we may view strings as partial maps in $\mathbb{N} \twoheadrightarrow \mathbf{Bool}$, defined for a prefix of the natural numbers $0 < 1 < 2 < \ldots$ An infinite string is simply a total map.

We use the notation Σ^ω for the set of all infinite strings over some alphabet Σ, and Σ^∞ for the set of all finite or infinite strings over Σ. Note that the notation Σ^ω is consistent in that Σ^ω is a map from natural numbers, i.e. $\omega = \mathbb{N}$, to Σ. Note also that u is a prefix of v iff $u \subseteq v$, and that this extends to infinite chains of prefixes. Hence if we have a prefix chain of strings

$$w_0 \lhd w_1 \lhd w_2 \lhd \ldots$$

then the least upper bound is simply the infinite union of $w_0, w_1, w_2, \ldots$ viewed as (partial) maps. The result is an infinite string (i.e. a total map) if the chain is infinite, and a finite string (partial map) if the chain is finite. For instance, the least upper bound of

$$0 < 01 < 010 < 0101 < 01010 < 010101 < \ldots$$

is the infinite string of alternating ones and zeros starting with zero. That is the function w where

$$w(n) = \begin{cases} 0 \text{ if } n \text{ is even} \\ 1 \text{ if } n \text{ is odd.} \end{cases}$$

Infinite strings (or traces) have attracted a great deal of attention in the field of concurrency theory, and we return to them in Chapter 6 □

Exercises

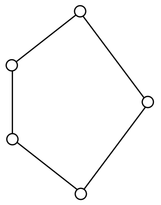
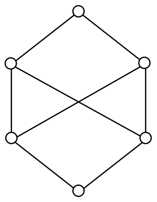
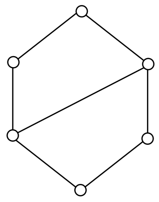
2.1 Which of the following structures are complete lattices? Give a counter-example if not, otherwise describe the operations of least-upper-bound and greatest-lower-bound and the top and bottom elements:

(1) The set of all finite strings Σ^* under the (non-strict) lexicographical order, where e.g. $a \sqsubseteq ab \sqsubseteq abb \sqsubseteq ac$.

(2) The set $\mathbb{N}$ under the partial order $m \preceq n$ iff there exists an $i \in \mathbb{N}$ such that $m \cdot i = n$.

(3) The set of all equivalence relations on a set A, ordered by set inclusion.

2.2 Which of the following Hasse diagrams represent lattices?

2.3 Prove that the following definitions of a lattice are equivalent:

- A poset $(A, \leq)$ where all $x, y \in A$ have a lub and glb.
- A poset $(A, \leq)$ where every finite and non-empty subset has a lub and glb.

2.4 Consider a lattice $(A, \leq)$ with lub $\vee$ and glb $\wedge$. Show that the following conditions are equivalent:
(1) $a \leq b$,
(2) $a \wedge b = a$,
(3) $a \vee b = b$.

2.5 If possible give an example of a 5-element lattice which is distributive, and one which is not.

2.6 Prove that if a lattice $(A, \leq)$ satisfies $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for all $a, b, c \in A$, then it also satisfies $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.

2.7 If possible give an example of a 5-element complemented lattice which has unique complements, and one which has not.

2.8 Use the algebraic laws $L_1 - L_4$ to prove that $a \otimes b = a$ iff $a \oplus b = b$.

2.9 Prove that the componentwise product of two complete lattices is a complete lattice as well.

2.10 Let $(A, \leq)$ be a non-empty poset. Prove that the following are equivalent:
(1) $(A, \leq)$ is a complete lattice,
(2) $\bigwedge B$ exists for every $B \subseteq A$,
(3) $\bigvee A$, and $\bigwedge B$ exist for every non-empty $B \subseteq A$.

2.11 Prove that $(\mathcal{O}(A), \subseteq)$ is a complete lattice.

2.12 Prove that the two definitions of a lattice – the one based on an ordered set $(A, \leq)$ with least upper and greatest lower bounds, and the algebraic one $(A, \otimes, \oplus)$ – are equivalent. That is
(1) Given a lattice $(A, \leq)$ where $x \leq y$ iff $x \otimes y = x$ (or $x \oplus y = y$), prove that the algebra $(A, \otimes, \oplus)$ satisfies $L_1 - L_4$.
(2) Given an algebra $(A, \otimes, \oplus)$ satisfying $L_1 - L_4$, prove that $(A, \leq)$, where $x \leq y$ iff $x \otimes y = x$ (or $x \oplus y = y$), is a poset and that each $x, y \in A$ has a least upper and greatest lower bound.

2.13 Prove Theorem 2.21.

2.14 Give an example of a cpo with a top element which is not a complete lattice.

CHAPTER 3

Ordinal numbers

The natural numbers are often viewed as abstractions of finite sets. For instance, 7 is an abstraction of the set of weekdays or the set of mortal sins. But natural numbers can also be used to describe the *position* of an element in a sequence or a chain. For instance, July is the 7th month in the total order of all months ordered in the standard manner

$$\text{January}<\text{February}<\text{March}<\text{April}< \ldots <\text{December}.$$

Actually instead of saying that 7 is the position of July, we may view it as the length of the (strict) chain which includes July and all its predecessors

$$\text{January}<\text{February}<\text{March}<\text{April}<\text{May}<\text{June}<\text{July}.$$

In the very first example 7 denotes the equivalence class of all sets of size 7, in which case we talk of the *cardinal* number 7. In the second case 7 denotes the equivalence class of all strict chains of length 7. In this case we refer to 7 as an *ordinal number*, or simply *ordinal*. Hence, ordinal numbers carry more information than cardinal numbers, namely order in addition to size. This idea is fairly straightforward for finite sets and finite chains, but intuition is usually blurred when considering infinite sets and chains.

For cardinal numbers we say that two (possibly infinite) sets A and B have the same cardinality iff there exists a bijective mapping $f\colon A \to B$ (and hence a bijective mapping $f^{-1}\colon B \to A$). If there exists a bijection we also say that A and B are *isomorphic* (from Greek "of the same form") or *similar*. We often write $A \sim B$ in this case. We know for instance that $\mathbb{N} \sim \mathbb{Z} \sim \mathbb{Q}$ but $\mathbb{N} \not\sim \mathbb{R}$. It is easy to prove that $\sim$ is an equivalence relation, and each equivalence class of this relation is a cardinal number: the least cardinal number, written $\mathbf{0}$, is the equivalence class that contains $\emptyset$ (and nothing else); the next cardinal number, written $\mathbf{1}$, is the equivalence class of all singleton sets, etc. The least infinite cardinal number is written $\aleph_0$ and contains the set of all natural numbers, as well as all other infinite, denumerable sets.

We can extend this notion of structural similarity to ordered sets (the same idea applies to reflexive as well as irreflexive orders).

Definition 3.1. A function f from $(A, <)$ to $(B, \prec)$ is called *monotonic* (*isotone*, *order-preserving*) iff $x < y$ implies $f(x) \prec f(y)$ for all $x, y \in A$. □

We sometimes say that f is an order-homomorphism (or order-morphism) from $(A, <)$ into $(B, \prec)$ when f is monotonic.

Definition 3.2. An order-homomorphism f from $(A, <)$ into $(B, \prec)$ is called

- a *monomorphism* if f is injective;
- an *epimorphism* if f is onto (surjective);
- an *isomorphism* if f is bijective (injective and onto).

□

Definition 3.3. Two ordered sets $\mathcal{A} := (A, <)$ and $\mathcal{B} := (B, \prec)$ are said to be (order-)*isomorphic* if there exists an order-isomorphism $f\colon A \to B$. We write $\mathcal{A} \simeq \mathcal{B}$ when $\mathcal{A}$ and $\mathcal{B}$ are isomorphic. □

One can easily show that $\simeq$ must be an equivalence relation by using elementary properties of bijective mappings.

Consider the following examples of strict chains:

(1) $\mathbb{Z}$ ordered as $\ldots < -2 < -1 < 0 < 1 < 2 < \ldots$.
(2) $\mathbb{Z}$ ordered as $0 < -1 < 1 < -2 < 2 < \ldots$
(3) $\mathbb{N}$ ordered as $0 < 1 < 2 < 3 < 4 < \ldots$
(4) $\mathbb{N}$ ordered as $1 < 2 < 3 < 4 < \ldots < 0$

Abstracting away from order it is evident that all four orders have the same cardinality, namely $|\mathbb{N}| = |\mathbb{Z}| = \aleph_0$. However, structurally they are pair-wise different with except of (2) and (3). The first order has neither a minimal nor a maximal element. Hence it cannot be isomorphic to a chain with a minimal (or maximal) element. The last order has both a minimal and a maximal element and is therefore not isomorphic to any of the other three; the two middle orderings have only a minimal element. In fact, they are isomorphic – there exists a bijective and order-preserving mapping from $\mathbb{Z}$ to $\mathbb{N}$ (and hence also in the other direction), namely

$$f(n) = \begin{cases} 2n \text{ if } n \geq 0, \\ -2n - 1 \text{ if } n < 0. \end{cases}$$

Yet another important difference between (1) and the other ones is that the former has no minimal element, while the latter three all do. In fact, any subset of the last three has a minimal element. That is, the last three ones are *well-orders*. Just as a cardinal number represents the equivalence class of all isomorphic sets, an *ordinal number*, or simply *ordinal*, is an equivalence class of all isomorphic well-orders. For instance, the well-order

$$\text{January}<\text{February}<\text{March}<\text{April}$$

represents the same ordinal number as

$$0 < 1 < 2 < 3 \text{ or } 1 < 2 < 3 < 4 \text{ or } aa \ll ab \ll abc \ll aca$$

This ordinal is usually denoted **4**, and it is the *equivalence class* of all isomorphic well-orders with four elements. Although the equivalence class is infinite the ordinal **4** is said to be *finite* since each well-order in the class is finite.

The least ordinal number is clearly $\emptyset$ (i.e. the empty well-order). We usually write this ordinal as **0**. This ordinal is special since it is the only ordinal that is a singleton. The second ordinal is the set of all well-orders of cardinality one; that is, all singleton sets, for instance $\{\emptyset\}$ or $\{0\}$ or $\{\text{April}\}$ (of course, in a singleton set the ordering is trivial). This ordinal is usually written **1**. The third ordinal, written **2**, is any well-ordered binary set. For instance, $\{0, 1\}$ or $\{1, 3\}$ under the standard ordering, or $\{\emptyset, \{\emptyset\}\}$ under set membership (since $\emptyset \in \{\emptyset\}$). The fourth ordinal is any well-ordered set of cardinality 3. For instance, $0 < 1 < 2$, or $\emptyset \in \{\emptyset\} \in \{\emptyset, \{\emptyset\}\}$. Not surprisingly, this ordinal is usually written **3**. Like cardinal numbers, finite ordinals are not very exciting; the ordinal of a finite well-ordered set is simply the cardinality of the well-ordered set.

Based on ordinary set theory John von Neumann gave an elegant definition of the class of all ordinals, or to be more precise: a definition of a *representative* of all ordinals. That is, every von Neumann ordinal is a well-order, and every well-order is isomorphic to a von Neumann ordinal. The definition of von Neumann is based on a technique called *transfinite induction* which will be discussed in some detail at the end of this chapter. For now, we note that the smallest ordinal is the empty set $\emptyset$ which is the unique representative of $\mathbf{0}$. This is also the smallest von Neumann ordinal. The basic idea is to order sets under set membership, hence since $\{\emptyset\}$ is a singleton and since $\emptyset \in \{\emptyset\}$, the set $\{\emptyset\}$ represents the ordinal $\mathbf{1}$. To represent the third ordinal (i.e. $\mathbf{2}$) we pick $\{\emptyset, \{\emptyset\}\}$ since

$$\emptyset \in \{\emptyset, \{\emptyset\}\} \text{ and } \{\emptyset\} \in \{\emptyset, \{\emptyset\}\}.$$

More generally if A is a von Neumann ordinal, then $A \cup \{A\}$ is the next von Neumann ordinal. Moreover if α is the ordinal number of the von Neumann ordinal A then we often write $\alpha + \mathbf{1}$ for the ordinal number of $A \cup \{A\}$. (We will introduce a general notion of addition involving ordinals in the next section.) While abusing notation it is customary to identify a von Neumann ordinal with its ordinal number; thus, instead of writing correctly that $\{\emptyset, \{\emptyset\}\} \in \mathbf{2}$ we often write instead that $\{\emptyset, \{\emptyset\}\} = \mathbf{2}$. Hence, by abuse of notation we will write incorrectly that $\mathbf{1} = \{\mathbf{0}\}$, and $\mathbf{2} = \mathbf{1} \cup \{\mathbf{1}\} = \{\mathbf{0}, \mathbf{1}\}$ etc. That is, a von Neumann ordinal is the set of all of its predecessors. The finite ordinal numbers thus look as follows (recall that the ordinal $\mathbf{3}$ also can be written as $\mathbf{2} + \mathbf{1}$)

Ordinal	von Neumann representation
$\mathbf{0}$	$\emptyset$
$\mathbf{1}$	$\{\emptyset\} = \mathbf{0} \cup \{\mathbf{0}\} = \{\mathbf{0}\}$
$\mathbf{2}$	$\{\emptyset, \{\emptyset\}\} = \mathbf{1} \cup \{\mathbf{1}\} = \{\mathbf{0}, \mathbf{1}\}$
$\mathbf{3}$	$\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} = \mathbf{2} \cup \{\mathbf{2}\} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$
$\mathbf{4}$	$\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\} = \mathbf{3} \cup \{\mathbf{3}\} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}$
etc.	

Note that e.g. $\mathbf{3} \in \mathbf{4}$ since $\mathbf{4} = \mathbf{3} \cup \{\mathbf{3}\}$ (i.e. $\mathbf{4} = \mathbf{3} + \mathbf{1}$). The canonical representation of finite ordinals is thus well-ordered by set membership $\in$.

More generally, if $\{A_i\}_{i \in I}$ is a set of von Neumann ordinals then

$$\bigcup_{i \in I} A_i$$

is also a von Neumann ordinal. For example, $\mathbf{0} \cup \mathbf{1} = \mathbf{1}$ and $\mathbf{1} \cup \mathbf{3} = \mathbf{3}$. Next consider the infinite set $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$ of all finite von Neumann ordinals. Since

$$\mathbf{0} \cup \mathbf{1} \cup \mathbf{2} \cup \ldots = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$$

it follows that $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$ is a von Neumann ordinal. Since, all finite von Neumann ordinals are members in this set it also follows that this ordinal is strictly greater than all finite von Neumann ordinals. The ordinal number of this set is usually denoted ω, which is the least infinite ordinal. Another example of a representative of the ordinal ω is the well-order $0 < 1 < 2 < 3 < \ldots$ on the natural numbers. The ordinal ω is often viewed as a synonym of $\mathbb{N}$ but it is more accurate to think of it as the *well-ordered* set $(\mathbb{N}, <)$.

The ordinal ω differs from all previous ordinals (except $\mathbf{0}$) in that it is not a successor of any previous ordinal, but rather the limit of all finite ordinals. It is therefore called a *limit* ordinal. Note that e.g. $\mathbf{1} \in \mathbf{2} \in \mathbf{3} \in \mathbf{4} \in \ldots$ is isomorphic to $\mathbf{0} \in \mathbf{1} \in \mathbf{2} \in \mathbf{3} \in \ldots$, so its ordinal number is also ω. In fact, if we have a bijective map $f\colon A \to \mathbb{N}$ (i.e. $|A| = \aleph_0$) then any ordered set $(A, \prec)$, where $x \prec y$ iff $f(x) < f(y)$, has the ordinal number ω.

Now ω has a successor ordinal represented e.g. by $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots, \omega\}$ which is denoted $\omega + \mathbf{1}$ since it is the successor ordinal of ω and this ordinal also has a successor, e.g. represented by $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots, \omega, \omega + \mathbf{1}\}$ sometimes denoted $\omega + \mathbf{2}$ or $(\omega + \mathbf{1}) + \mathbf{1}$. Continuing we eventually obtain the ordinal

$$\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots, \omega, \omega + \mathbf{1}, \omega + \mathbf{2}, \omega + \mathbf{3}, \ldots\}$$

This (limit) ordinal is written $\omega + \omega$ or $\omega \cdot \mathbf{2}$. The *cardinality* of this ordinal is the same as the cardinality of ω but they are not the same ordinals.

The complete definition of von Neumann ordinals can now be formulated as follows using definition by transfinite induction:

- $\emptyset$ is a von Neumann ordinal,
- if A is a von Neumann ordinal then so is its successor $A \cup \{A\}$,
- if $\{A_i\}_{i \in I}$ is a set of von Neumann ordinals then so is
$$\bigcup_{i \in I} A_i.$$

From now on we usually refer to von Neumann ordinals simply as ordinals unless stated otherwise.

3.1. Ordinal arithmetic

Assume that we have two disjoint well-ordered sets A and B. Then it is easy to create a new well-ordered set $A \cup B$ where $a < b$ for all $a \in A$ and $b \in B$. This idea can be used to define a notion of addition of ordinal numbers. To add two ordinals α and β we pick two disjoint well-orders A resp. B from α resp. β and create a new well-order where all $a < b$ for all $a \in A$ and all $b \in B$. The ordinal number of the resulting well-order is denoted $\alpha + \beta$.

Example 3.4. Let A be $0 < 1 < 2 < 3$ and B be $4 < 5 < 6$. Then the ordinal number of the well-order $0 < 1 < 2 < 3 < 4 < 5 < 6$ is $\mathbf{7}$ and so is the ordinal number of $4 < 5 < 6 < 0 < 1 < 2 < 3$. Hence, $\mathbf{4} + \mathbf{3} = \mathbf{3} + \mathbf{4} = \mathbf{7}$. □

It is easy to see that the ordinal number of the new well-order is independent of the choice of A and B as long as they are disjoint. (If they are not disjoint the result is not even a well-order.)

Contrary to what is suggested by the example the cummutative law does *not* hold in general for addition of ordinals. Consider the ordinals ω and $\mathbf{1}$, and consider first $\omega + \mathbf{1}$. The well-order $1 < 2 < 3 < 4 < \ldots$ has ordinal number ω, and the well-order 0 has ordinal number $\mathbf{1}$. But the well-order $1 < 2 < 3 < 4 < \ldots < 0$ is isomorphic to $0 < 1 < 2 < \ldots < \omega$ which has ordinal number $\omega + \mathbf{1}$; which hopefully explains why we denote the successor ordinal of ω by $\omega + \mathbf{1}$. If we instead consider $\mathbf{1} + \omega$ we note that the well-order $0 < 1 < 2 < 3 < 4 < \ldots$ has ordinal number ω and is not isomorphic to $1 < 2 < 3 < 4 < \ldots < 0$. Hence, $\mathbf{1} + \omega \neq \omega + \mathbf{1}$.

Note also that the following holds (in contrast to addition of cardinal numbers).

Theorem 3.5. If $\beta \neq \mathbf{0}$ then $\alpha < \alpha + \beta$ for all ordinals α. □

We have outlined how to do addition of two ordinals, thus obtaining a new ordinal. It is also possible to multiply ordinals: Let α and β be two ordinals, and assume that A resp. B are well-orders in α resp. β. We then define $\alpha \cdot \beta$ to be the ordinal number of the Cartesian product of the sets A and B, under the reverse lexicographical order. More precisely, $\alpha \cdot \beta$ is the ordinal number of set $\{(a, b) \mid a \in A \text{ and } b \in B\}$ under the ordering

$$(a_1, b_1) \prec (a_2, b_2) \text{ iff either } b_1 < b_2, \text{ or } b_1 = b_2 \text{ and } a_1 < a_2.$$

Hence, $\mathbf{2} \cdot \omega$ is

$$(0,0) \prec (1,0) \prec (0,1) \prec (1,1) \prec (0,2) \prec (1,2) \prec \dots$$

which is isomorphic to ω, that is $\mathbf{2} \cdot \omega = \omega$. If we instead consider $\omega \cdot \mathbf{2}$ we get

$$(0,0) \prec (1,0) \prec (2,0) \prec (3,0) \prec \dots \prec (0,1) \prec (1,1) \prec (2,1) \prec (3,1) \prec \dots$$

which is isomorhic to $\omega + \omega$. Hence, $\omega \cdot \mathbf{2} = \omega + \omega \neq \mathbf{2} \cdot \omega$. The finite n-fold product of α is often written $\alpha^{\mathbf{n}}$. More generally, it is also possible to define α^{β} for all ordinals α and β (see e.g. [**Hal61**] for an extensive exposition of ordinal numbers and ordinal arithmetic).

3.2. Ordinal powers of functions

An inductive definition of a set $S \subseteq A$ consists of one or more base cases defining a set $B \subseteq S \subseteq A$; an inductive definition also consists of one or more inductive cases, saying how to induce new members in the set from existing ones. We may view this as a function $\mathcal{R} \colon 2^A \to 2^A$ which given $X \subseteq S \subseteq A$ induces the new members $\mathcal{R}(X)$ into S, i.e. $\mathcal{R}(X) \subseteq S$. An inductively defined set S is then the least set S satisfying,

(1) $B \subseteq S$, and
(2) $\mathcal{R}(S) \subseteq S$.

In fact, by requiring S to be the *least* set satisfying (1) and (2) we prefer to have $S = B \cup \mathcal{R}(S)$. For example, consider the standard inductive definition of the set E of even natural numbers.

(1) $\{0\} \subseteq E$, and
(2) if $X \subseteq E$ then $\{n + 2 \mid n \in X\} \subseteq E$

The inductive case can be formulated as a function $\mathcal{R} \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ where $\mathcal{R}(X) := \{n + 2 \mid n \in X\}$ and we can define a function $f \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ where

$$f(X) := \{0\} \cup \mathcal{R}(X),$$

The set E can be obtained by the following infinite union

$$E := \bigcup_{n \geq 0} f^n(\emptyset).$$

That is, the least upper bound of the ascending chain $\emptyset \subset \{0\} \subset \{0, 2\} \subset \{0, 2, 4\} \subset \dots$. This is a general scheme for constructing inductively defined sets which will be discussed further in Chapter 5.

We next outline some notions useful for generalized definition of infinite and transfinite (beyond infinity) sets. Consider a function $f \colon A \to A$ on a complete lattice $(A, \leq)$. The (ascending) ordinal powers of f are defined as follows:

$$\begin{array}{lcl} f^0(x) & := & x \\ f^{\alpha+\mathbf{1}}(x) & := & f(f^{\alpha}(x)) \text{ for successor ordinals } \alpha + \mathbf{1} \\ f^{\alpha}(x) & := & \bigvee_{\beta < \alpha} f^{\beta}(x) \text{ for limit ordinals } \alpha \end{array}$$

When x equals $\bot$ we simply write f^α instead of $f^\alpha(\bot)$. That is:

$$\begin{array}{lll} f^0 & := & \bot \\ f^{\alpha+\mathbf{1}} & := & f(f^\alpha) \text{ for successor ordinals } \alpha+\mathbf{1} \\ f^\alpha & := & \bigvee_{\beta<\alpha} f^\beta \text{ for limit ordinals } \alpha \end{array}$$

The definition of $f^\alpha(x)$ applies also when A is a cpo if f is monotonic and $x \leq f(x)$. For complete lattices we also have a corresponding dual notion of descending ordinal powers:

$$\begin{array}{lll} f^0(x) & := & x \\ f^{\alpha+\mathbf{1}}(x) & := & f(f^\alpha(x)) \text{ for successor ordinals } \alpha+\mathbf{1} \\ f^\alpha(x) & := & \bigwedge_{\beta<\alpha} f^\beta(x) \text{ for limit ordinals } \alpha \end{array}$$

EXAMPLE 3.6. Consider our inductive definition of the even natural numbers again. The finite ordinal powers of

$$f(X) := \{0\} \cup \{n+2 \mid n \in X\}$$

look as follows,

$$\begin{array}{lll} f^0(\emptyset) & = & \emptyset \\ f^1(\emptyset) & = & \{0\} \\ f^2(\emptyset) & = & \{0,2\} \\ f^3(\emptyset) & = & \{0,2,4\} \\ \ldots & & \\ f^n(\emptyset) & = & \{0,2,4,\ldots,2(n-1)\} \end{array}$$

Consequently

$$f^\omega(\emptyset) \quad = \quad \{0,2,4,6,8,\ldots\}.$$

Note also that

$$f^{\omega+\mathbf{1}}(\emptyset) \quad = \quad \{0,2,4,6,8,\ldots\}.$$

In fact, for any ordinal α we have

$$f^{\omega+\alpha}(\emptyset) = f^\omega(\emptyset).$$

□

EXAMPLE 3.7. Consider a transition system $(C, \to, I)$ with initial configurations I; i.e. $I \subseteq C$. Let $\textsc{step}\colon 2^C \to 2^C$ be a step function

$$\textsc{step}(x) := I \cup \{c \in C \mid \exists c' \in x \text{ such that } c' \to c\}$$

Then

$$\begin{array}{lll} \textsc{step}^0(\emptyset) & = & \emptyset \\ \textsc{step}^1(\emptyset) & = & \textsc{step}(\emptyset) \;=\; I \\ \textsc{step}^2(\emptyset) & = & \textsc{step}(\textsc{step}^1(I)) \;=\; \textsc{step}(I). \end{array}$$

That is, for finite ordinals n, $\textsc{step}^n(\emptyset)$ is the set of all configurations reachable in fewer than n steps from some initial configuration. Moreover, the least infinite ordinal power, corresponding to the limit ordinal ω is

$$\textsc{step}^\omega(I) = \bigcup_{n<\omega} \textsc{step}^n(I).$$

The limit ordinal is the set of all configurations reachable in a finite (but unbounded) number of steps from an initial configuration. □

EXAMPLE 3.8. Consider a relation $R \subseteq A \times A$. Let $(R\circ)(S)$ be defined as $R \circ S$, i.e. the function which composes the relation R with the relation $S \subseteq A \times A$. That is, $(R\circ)\colon 2^{A\times A} \to 2^{A\times A}$. The ordinal powers of $(R\circ)$ are as follows

$$\begin{array}{lcl}
(R\circ)^0(S) & = & S \\
(R\circ)^{\alpha+\mathbf{1}}(S) & = & (R\circ)((R\circ)^\alpha(S)) = R \circ (R\circ)^\alpha(S) \\
(R\circ)^\alpha(S) & = & \bigcup_{\beta<\alpha}(R\circ)^\beta(S) \text{ for limit ordinals } \alpha.
\end{array}$$

And

$$\begin{array}{lcl}
(R\circ)^0(\textsc{id}_A) & = & \textsc{id}_A \\
(R\circ)^1(\textsc{id}_A) & = & R \circ \textsc{id}_A = R \\
(R\circ)^2(\textsc{id}_A) & = & R \circ R \circ \textsc{id}_A = R \circ R \\
(R\circ)^3(\textsc{id}_A) & = & R \circ R \circ R \\
\text{etc.} & &
\end{array}$$

It follows that $(R\circ)^\omega(\textsc{id}_A)$ is the reflexive and transitive closure of R. Similarly $(R\circ)^\omega(R)$ is the transitive closure of R. □

Exercises

3.1 Prove that $\sim$ (having the same cardinality as) is an equivalence relation on the class of all sets.

3.2 Give an example of a well-order of $\mathbb{N}$ with the ordinal number $\omega + \omega + \omega$ (or $\omega \cdot 3$).

3.3 Give an example of a well-order of $\mathbb{N}$ with the ordinal number $\omega \cdot \omega$ (i.e. intuitively an infinite sequence of infinite sequences of natural numbers).

CHAPTER 4

Principles of induction

STANDARD MATHEMATICAL INDUCTION ON the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ states that if a property P is true of 0, and if whenever P is true of $n \geq 0$ then P is true also of $n+1$, then P is true of all natural numbers. Formulated as a derivation rule the principle can be stated as follows

$$\frac{P(0) \qquad \forall n \in \mathbb{N}\ (P(n) \Rightarrow P(n+1))}{\forall n \in \mathbb{N}\ P(n)}\ .$$

The principle of mathematical induction actually applies to any well-ordered set isomorphic to ω. However, the induction hypothesis $P(n)$ is sometimes not sufficiently strong to prove $P(n+1)$ for all $n \in \mathbb{N}$. In that case we may generalize the induction scheme to so-called *strong* induction, where we strengthen the induction hypothesis as follows

$$\frac{P(0) \qquad \forall n \in \mathbb{N}\ (P(0) \wedge \ldots \wedge P(n) \Rightarrow P(n+1))}{\forall n \in \mathbb{N}\ P(n)}$$

or stated more economically

$$\frac{\forall n \in \mathbb{N}\ (P(0) \wedge \ldots \wedge P(n-1) \Rightarrow P(n))}{\forall n \in \mathbb{N}\ P(n)}\ .$$

Again, this principle applies to any well-ordered set isomorphic to ω. However, what happens if we want to prove a property of an infinite set which is not isomorphic to $(\mathbb{N}, <)$? For instance

- the set $2^{\mathbb{N}}$ under $\subset$
- $(\mathbb{N}, <)$ extended with a top element $\top$.

In this chapter we consider two induction schemes which extend standard induction and which are often more adequate in computer science applications; so-called *well-founded induction* and *transfinite induction.*

4.1. Well-founded induction

We first consider an induction scheme which subsumes most of the standard inductions; namely *well-founded induction.* While standard induction applies to well-ordered sets isomorphic to the natural numbers, well-founded induction applies to any well-founded set. If this motivation is not convincing enough we first show informally that any (standard) inductive definition gives rise to a well-founded set. Hence, well-founded induction can be used to prove properties of *any* inductively defined set, including functions and relations.

Recall that a well-founded relation $\prec \subseteq A \times A$ is a relation where each non-empty subset of A has a minimal element. A straightforward consequence of this is that A can contain no infinite descending chain $\ldots x_2 \prec x_1 \prec x_0$ since $\{x_0, x_1, x_2, \ldots\}$ must contain a minimal element.

A well-founded set (A, R) is typically defined by inductive definitions (see e.g. Aczel [**Acz77**]); and an (unambiguous) inductive definition typically gives rise to a well-founded set: As discussed in previous chapters, an inductive definition of A typically consists of three (types of) statements

- one or more *base cases*, B, saying that $B \subseteq A$,
- one or more *inductive cases*, saying schematically that if $x \in A$ and y is in some relation with x, i.e. $R(x, y)$, then $y \in A$,
- an *extremal* condition stating that A contains no more elements than those given by the base and inductive case. One way of stating this is to say that A is the *least* set satisfying the base and inductive case. (This presumes of course that there is such a least set; a problem that we address in Chapter 5.)

Now let us write $\mathcal{R}(X)$ for $\{y \mid \exists x \in X, R(x, y)\}$, i.e. the set of all elements that can be induced from X. Then an inductive definition of A states that A is the least set X such that

$$B \subseteq X \text{ and } \mathcal{R}(X) \subseteq X, \text{ that is, } B \cup \mathcal{R}(X) \subseteq X$$

Provided that R contains no cycles $x_0 \; R \; x_1 \; R \ldots R \; x_n = x_0$ $(n \geq 1)$ the result is a well-founded set (A, R) with minimal elements B. One way of guaranteeing this is to make sure that R is a strict partial order. And if R is not a strict poset, it is always possible to find a subrelation $R' \subseteq R$ which is well-founded, and which índuces the same set.

Example 4.1. Consider the definition of the factorial function. We define *fact*: $\mathbb{N} \to \mathbb{N}$ to be the least set closed under the following

(1) $(0 \mapsto 1) \in \mathit{fact}$,
(2) if $(n \mapsto m) \in \mathit{fact}$ and $n' = n + 1, m' = (n + 1)m$ then $(n' \mapsto m') \in \mathit{fact}$

Since $(0 \mapsto 1) \in \mathit{fact}$ it follows that $(1 \mapsto 1) \in \mathit{fact}$; hence $(2 \mapsto 2) \in \mathit{fact}$; hence $(3 \mapsto 6) \in \mathit{fact}$ etc. Actually, claiming this to be the definition of the factorial *function* is a bit bold; we have only defined a set and it remains to be shown that *fact* is a function, but it is easy to see that *fact* must be a function on $\mathbb{N} \to \mathbb{N}$.

The inducing relation R is a relation between pairs of natural numbers where

$$R((n, m), (n', m')) \text{ iff } n' = n + 1 \text{ and } m' = (n + 1)m.$$

The relation R must be acyclic since if $R((n, m), (n', m'))$ then $n < n'$. Hence, *fact* is well-founded under R.[1] □

We now state the principle of well-founded induction, which applies to any well-founded set. Let $(A, \prec)$ be a well-founded set and P a property of A.

(1) If P holds of all minimal elements of A, and
(2) whenever P holds of all x such that $x \prec y$ then P holds of y,

[1] The inductive case obviously could have been stated more succinctly as: if $(n, m) \in \mathit{fact}$ then $(n + 1, (n + 1)m) \in \mathit{fact}$.

then P holds of all $x \in A$. Expressed in the form of a derivation rule the principle can be expressed as follows (note that the condition that P holds for all minimal elements actually is a special case of the inductive case, as in strong induction)

$$\frac{\forall y \in A\ (\forall x \in A\ (x \prec y \Rightarrow P(x)) \Rightarrow P(y))}{\forall x \in A\ P(x)}\ .$$

As pointed out well-founded induction encompasses most other forms of induction including "standard" induction and strong induction over subsets of the integers, and the notion of structural induction illustrated by the following example:

EXAMPLE 4.2. Consider the standard definition of a language of propositional logic given a finite set *Var* of propositional variables.

$$\begin{array}{lll} F & ::= & \mathit{Var} \\ F & ::= & \neg F \mid (F \wedge F) \mid (F \vee F) \mid (F \rightarrow F) \mid \ldots \end{array}$$

This inductive definition induces an irreflexive subformula ordering: if G is a direct subformula of F then $G \prec F$. For instance, $\neg A \prec (B \wedge \neg A)$ and $B \prec (B \wedge \neg A)$, but $A \not\prec (B \wedge \neg A)$. The set $(F, \prec)$ of formulas is well-founded since any subset of formulas by necessity contains some minimal formulas.

Now, let $\sigma_\top$ be the valuation such that $\sigma_\top(x) = 1$ for all $x \in \mathit{Var}$. A propositional formula F is said to be *positive* iff F is true in $\sigma_\top$, written $\sigma_\top \models F$. For instance, given $\Sigma = \{x, y\}$ then $x \wedge y$ is positive and so is $x \vee y$, but neither $\neg x$ nor $\neg(x \wedge y)$ are positive.

We now prove, using well-founded induction, that the following subset of formulas are positive

$$\begin{array}{lll} P & ::= & \mathit{Var} \\ P & ::= & (P \wedge P) \mid (P \vee P) \mid (P \rightarrow P) \end{array}$$

BASE CASE: All minimal elements of P (the propositional variables) are clearly positive. That is, if $x \in \mathit{Var}$ then $\sigma_\top \models x$.

INDUCTIVE CASE: Assume that we have a non-atomic formula $F \in P$ all of whose proper subformulas are positive. If F is of the form $F_1 \rightarrow F_2$ then by assumption $\sigma_\top \models F_1$ and $\sigma_\top \models F_2$. Hence, $\sigma_\top \models F_1 \rightarrow F_2$, i.e. F is positive. The remaining two cases are analogous.

NOTE: Being a member in P is a sufficient, but not necessary, condition for being a positive formula, since e.g. $\neg x \vee y$ is positive but not contained in P. □

Well-founded sets are used extensively when proving convergence, in particular termination, of computations.

EXAMPLE 4.3. One can show that $\mathbb{Z}^+ \times \mathbb{Z}^+$ is well-founded under the ordering

$$(m_1, n_1) \ll (m_2, n_2) \text{ iff } \mathit{max}(m_1, n_1) < \mathit{max}(m_2, n_2).$$

Now consider the standard recursive algorithm for computing the greatest common divisor (assuming an initial configuration $\mathit{gcd}(m, n)$ with $m, n > 0$):

$$\begin{array}{lll} (T_1) & \mathit{gcd}(m, m) \Rightarrow m & \\ (T_2) & \mathit{gcd}(m, n) \Rightarrow \mathit{gcd}(m-n, n) & \text{if } m > n \\ (T_3) & \mathit{gcd}(m, n) \Rightarrow \mathit{gcd}(m, n-m) & \text{if } m < n \end{array}$$

We prove termination by well-founded induction on $\ll$. The only minimal element is $(1, 1)$, and $\mathit{gcd}(1, 1)$ clearly terminates due to T_1.

Next assume that we have $gcd(m,n)$ with $m+n>2$ and that $gcd(i,j)$ terminates for all $(i,j) \ll (m,n)$. There are three cases to consider:

(1) If $m=n$ then $gcd(m,n)$ terminates trivially due to T_1.
(2) if $m>n>0$ then clearly $(m-n,n) \ll (m,n)$. That is, $gcd(m-n,m)$ terminates by the induction hypothesis. Hence, $gcd(m,n)$ must terminate since only T_2 is applicable.
(3) If $n>m>0$ then clearly $(m,n-m) \ll (m,n)$. Since $gcd(m,n-m)$ terminates by the induction hypothesis, then also $gcd(m,n)$ must terminate since only T_3 is applicable.

Hence, $gcd(m,n)$ terminates for all $m,n \in \mathbb{Z}^+$. □

Remark: Well-founded relations are often used indirectly e.g. in termination proofs. Consider a transition system $(C, \Rightarrow)$ and assume that we want to prove that there are no infinite computations $x_0 \Rightarrow x_1 \Rightarrow \ldots \Rightarrow x_n \Rightarrow \ldots$. We may prove this by exhibiting a so-called *norm function* f from C to some well-founded relation $(A, \prec)$, often the natural numbers under $<$, or a lexicographical well-order. Now computations starting with a terminal configuration are trivially finite. Moreover, if we can prove that $f(x_{i+1}) \prec f(x_i)$ whenever $x_i \Rightarrow x_{i+1}$, then we know for sure that there can be no infinite computations, since a well-founded set can have no infinite descending chain.

Example 4.4. Consider the transitions $T_1 - T_3$ again and define the following norm function $sum\colon (\mathbb{N} \times \mathbb{N}) \to \mathbb{N}$.

$$sum(m,n) := m+n.$$

As pointed out above, T_1 can be applied at most once; hence all infinite computations are due to T_2 and/or T_3. First consider T_2: assuming that $m>n>0$, then

$$sum(m-n,n) = m-n+n = m < m+n = sum(m,n).$$

Next consider T_3: assuming that $n>m>0$, then

$$sum(m,n-m) = m+(n-m) = n < m+n = sum(m,n).$$

Hence, whenever $(m_i,n_i) \Rightarrow (m_{i+1},n_{i+1})$ then $sum(m_{i+1},n_{i+1}) < sum(m_i,n_i)$. Thus, there can be no infinite computations assuming an initial configuration $gcd(m,n)$ where $m,n>0$. (A norm-function based on the measure max in the previous example works equally well.) □

4.2. Transfinite induction

Ordinals are well-orders which are well-founded sets. Hence, well-founded induction applies to properties of the ordinal numbers. However, the induction scheme can be somewhat simplified when reasoning about properties of ordinals; and especially, sets defined by ordinal powers. The resulting induction scheme is called the principle of *transfinite induction.*

Let P a property of ordinals, then P is true of every ordinal if

- P is true of 0,
- P is true of $\alpha+1$ whenever P is true of α,
- P is true of β whenever β is a limit ordinal and P is true of every $\alpha<\beta$.

(The first case is actually a special case of the last case.)

We illustrate the transfinite induction scheme by proving the following theorem.

THEOREM 4.5. Let $(A, \leq)$ be a complete lattice and assume that $f\colon A \to A$ is monotonic. We prove that $f^\alpha \leq f^{\alpha+1}$ for all ordinals α. □

Before giving the proof we give the following lemma whose proof is left as an exercise.

LEMMA 4.6. Let $(A, \leq)$ be a complete lattice and assume that $f\colon A \to A$ is monotonic. If $B \subseteq A$ then $f(\bigvee B) \geq \bigvee\{f(x) \mid x \in B\}$. □

We now prove Theorem 4.5.

PROOF. The property clearly holds for $\alpha = 0$ since $f^0 = \bot \leq f^1 = f(\bot)$.

Now assume that $f^\beta \leq f^{\beta+1}$. Since f is monotonic it follows that $f(f^\beta) \leq f(f^{\beta+1})$, that is $f^{\beta+1} \leq f^{\beta+2}$.

Finally assume that β is a limit ordinal and assume that the property holds for all $\alpha < \beta$; that is $f^\alpha \leq f^{\alpha+1}$ for all $\alpha \leq \beta$. Now using the Lemma 4.6 we get

$$\begin{array}{rcl} f^{\beta+1} & = & f(\bigvee\{f^\alpha \mid \alpha < \beta\}) \\ & \geq & \bigvee\{f(f^\alpha) \mid \alpha < \beta\} \text{ (by Lemma 4.6)} \\ & \geq & \bigvee\{f^\alpha \mid \alpha < \beta\} \\ & = & f^\beta. \end{array}$$

Hence, $f^\alpha \leq f^{\alpha+1}$ for all ordinals α. □

Exercises

4.1 Give a sufficient and necessary syntactic condition for being a positive formula and prove this.

4.2 Prove that $\mathbb{N} \times \mathbb{N}$ under the ordering

$$(x_1, x_2) \ll (y_1, y_2) \text{ iff } max(x_1, x_2) < max(y_1, y_2)$$

is well-founded.

4.3 Prove that the following program for concatenation of strings of a's is associative. That is, $\mathsf{aconc}(x, \mathsf{aconc}(y, z)) = \mathsf{aconc}(\mathsf{aconc}(x, y), z)$ for all strings $x, y, z \in \{a\}^*$.

$$\begin{array}{lcl} \mathsf{aconc}(\epsilon, y) & \Rightarrow & y \\ \mathsf{aconc}(a.x, y) & \Rightarrow & \mathsf{aconc}(x, a.y). \end{array}$$

4.4 Prove termination of Ackermann's function on $\mathbb{N}$.

$$\begin{array}{lcl} \mathsf{ack}(0, y) & := & y + 1 \\ \mathsf{ack}(x, 0) & := & \mathsf{ack}(x - 1, 1) \quad (x > 0) \\ \mathsf{ack}(x, y) & := & \mathsf{ack}(x - 1, \mathsf{ack}(x, y - 1)) \quad (x > 0, y > 0). \end{array}$$

Hint: Exhibit a well-founded order on $\mathbb{N} \times \mathbb{N}$ and prove by well-founded induction that each recursive call $\mathsf{ack}(m, n) \in \mathbb{N} \times \mathbb{N}$ must terminate.

4.5 Does the greatest common divisor program in Example 4.3 terminate if the domain is extended to $\mathbb{N} \times \mathbb{N}$?

4.6 Prove, using well-founded induction, that every natural number $n \geq 2$ is a product of prime numbers. (Well-founded induction reduces to strong mathematical induction in this case).

4.7 Prove Lemma 4.6.

4.8 A set/class A is said to be transitive iff $C \in A$ whenever $B \in A$ and $C \in B$. Prove, using transfinite induction, that the class of von Neumann ordinals is transitive.

4.9 Let $(A, \leq)$ be a cpo and let $f \colon A \to A$ be monotone. Prove, using transfinite induction, that $f^{\alpha}(\bot) \leq f^{\alpha+1}(\bot)$ for any ordinal α.

4.10 Let $(A, \leq)$ be a complete lattice and assume that $f \colon A \to A$ is monotonic. (That is, $f(x) \leq f(y)$ whenever $x \leq y$). Let $a \in A$ and assume that $a \leq f(a)$. Show that there must be some ordinal α such that $f^{\alpha+1}(a) = f^{\alpha}(a)$.

REMARK: We call x a fixed point of f iff $f(x) = x$. Now, since $\bot \leq f(\bot)$ all monotonic maps on complete lattices must have at least one fixed point. (We prove an even stronger result for monotonic maps on complete lattices in the next chapter.)

CHAPTER 5

Fixed points

In this chapter we consider the problem of finding solutions to equations of the form

$$f(x) = x$$

where $f\colon A \to A$. An element $a \in A$ such that $f(a) = a$ is called a *fixed point* of f. Note that our problem is more far-reaching than this. If we want to solve an equation $f(x) = 0$, this can be reformulated as the problem of solving the equation $f(x)+x = x$; hence, if we define $g(x) := f(x)+x$ we have again the problem $g(x) = x$. Of course, our function may not have any fixed points (e.g. $f(n) := n+1$), or it may have more than one fixed point (e.g. $f(n) := n$). The problem of determining if a function has a fixed point is undecidable in general and in this chapter we focus on two sufficient conditions under which we can guarantee the existence of fixed points, and in some cases even computer them (or at least approximate them arbitrarily well).

For a historic account of the use of fixed points in semantics of programming languages and logics, see Lassez et al. [**LNS82**].

5.1. Basic notions

We summarize some basic properties of functions on ordered sets.

Definition 5.1. Let $(A, \leq)$ be a poset. A function $f\colon A \to A$ is said to be

- *monotone* (order-preserving) iff $f(x) \leq f(y)$ whenever $x \leq y$.
- *antimonotone* iff $f(x) \geq f(y)$ whenever $x \leq y$.
- *inflationary* iff $x \leq f(x)$ for all $x \in A$.
- *idempotent* iff $f(f(x)) = f(x)$ for all $x \in A$.

□

A trivial case where we can always guarantee the existence of fixed points is when the map is idempotent.

Theorem 5.2. Let $(A, \leq)$ be a non-empty cpo and $f\colon A \to A$ idempotent. Then f has a (not necessarily unique) fixed point. □

We also introduce the notion of *continuous* maps. We first attempt a generic definition which is subsequently adapted to complete lattices and cpo's.

Definition 5.3. A function $f\colon A \to A$ is *continuous* if it preserves existing least upper bounds; i.e. if $B \subseteq A$ and $\bigvee B$ exists, then $\bigvee \{f(x) \mid x \in B\}$ exists and equals $f(\bigvee B)$. □

In the case of a complete partial order – which only guarantees the existence of suprema for ascending chains – we may specialize this as follows.

Definition 5.4. Let $(A, \leq)$ be a cpo. A function $f\colon A \to A$ is called (chain-) *continuous* if

$$f(\bigvee\{x_0, x_1, x_2, \ldots\}) = \bigvee\{f(x_0), f(x_1), f(x_2), \ldots\}$$

for every ascending chain $x_0 < x_1 < x_2 < \ldots$ in A. □

In the case of a complete lattice – where every subset has a supremum – we get:

Definition 5.5. Let $(A, \leq)$ be a complete lattice. A function $f\colon A \to A$ is *continuous* if

$$f(\bigvee B) = \bigvee\{f(x) \mid x \in B\}$$

for every $B \subseteq A$. □

A continuous map is always monotone (prove this), but the converse is not true in general. However, if $(A, \leq)$ is finite the two concepts coincide.

Theorem 5.6. If $f\colon A \to A$ is monotone and A is finite, then f must be continuous. □

For a cpo or a lattice we may weaken the condition that A is finite to the condition that A contains no infinite ascending chains (i.e. is finite-length in case of a lattice).

5.2. Knaster-Tarski's theorem

In this section we prove a classic fixed point theorem due to Knaster-Tarski (see [**Tar55**]) which concerns the existence of (least and greatest) fixed points of monotone maps on complete lattices. We also give, without proof, a similar result concerning the existence of (least) fixed points of monotone maps on cpo's.

Definition 5.7. Let $(A, \leq)$ be a poset and consider a map $f\colon A \to A$. An $x \in A$ such that $f(x) \leq x$ is called a *pre-fixed point* of f. Similarly $x \in A$ is called a *post-fixed point* of f iff $x \leq f(x)$. □

Example 5.8. As defined previously a set $B \subseteq A$ is closed under the map $f\colon A \to A$ iff $f(x) \in B$ whenever $x \in B$. We may "lift" f to $F\colon 2^A \to 2^A$ as follows

$$F(X) := \{f(x) \mid x \in X\}.$$

Then $B \subseteq A$ is closed under f iff B is a pre-fixed point of F, i.e. if $F(B) \subseteq B$. (In many cases the same symbol f is used for both functions, by abuse of notation.) □

We now state and prove the Knaster-Tarski fixed point theorem.

Theorem 5.9. Let $(A, \leq)$ be a complete lattice and $f\colon A \to A$ monotone. Then $\bigwedge\{x \in A \mid f(x) \leq x\}$ is the least fixed point of f, and $\bigvee\{x \in A \mid x \leq f(x)\}$ is the greatest fixed point of f. □

Proof. We prove the first part of the theorem. The second result can be shown dually. Consider the set $\mathcal{S} := \{x \in A \mid f(x) \leq x\}$ of all pre-fixed points of f. The set $\mathcal{S}$ clearly is non-empty since at least $f(\top) \leq \top$. Now let $y := \bigwedge \mathcal{S}$. Then by definition $y \leq x$ for every $x \in \mathcal{S}$. By montonicity of f, $f(y) \leq f(x) \leq x$, for every $x \in \mathcal{S}$. Hence $f(y) \leq x$ for every $x \in \mathcal{S}$, i.e. $f(y)$ is a lower bound of $\mathcal{S}$. But since y is the greatest lower bound we must have

$$f(y) \leq y. \tag{5.1}$$

Moreover, $f(f(y)) \leq f(y)$ (by monotonicity) which means that $f(y) \in \mathcal{S}$. Hence

$$y \leq f(y). \tag{5.2}$$

Our two inequalities imply not only that y is a fixed point but also that y is least, since all fixed points (including y) must be contained in $\mathcal{S}$. □

That is, the least pre-fixed point of f is the least fixed point of f. Dually, the greatest post-fixed point of f is the greatest fixed point of f.

The Knaster-Tarski theorem concerns complete lattices, but the existence of a least fixed point holds also for monotone maps on cpo's; however, the proof is more complicated, and we give the theorem without a proof.

Theorem 5.10. Let $(A, \leq)$ be a cpo and $f\colon A \to A$ monotone. Then f has a least fixed point. □

The least fixed point of f is denoted $\textsc{lfp}(f)$. Alternatively it is written

$$\mu x.f(x)$$

with reading: the least x such that $f(x) = x$. The greatest fixed point of f is often denoted $\textsc{gfp}(f)$, alternatively

$$\nu x.f(x).$$

Example 5.11. In many areas of computer science we model computations by transition systems, $(C, \Rightarrow)$, where C is a set of *configurations* and where $\Rightarrow \subseteq C \times C$ is a so-called *transition relation*. The configurations are snap-shots of the state of a computation and the transition relation describes the atomic steps of a system. Hence a computation is a finite, or possibly infinite, sequence of configurations

$$c_0 \Rightarrow c_1 \Rightarrow c_2 \Rightarrow \ldots$$

Now given a set of configurations $X \subseteq C$ we may define a function $\textsc{step}\colon 2^C \to 2^C$ mapping $X \subseteq C$ to the set of all configurations reachable in one step from some configuration in X. That is

$$\textsc{step}(X) := \{c \in C \mid \exists c' \in X \text{ such that } c' \Rightarrow c\}$$

This function is clearly monotone with respect to $\subseteq$; if $X \subseteq Y$ then $\textsc{step}(X) \subseteq \textsc{step}(Y)$. Hence, $\textsc{step}$ has a least and greatest fixed point. Of course, knowing that $\textsc{step}$ has a fixed point is of limited value, but we will soon see not only that it can be computed (or at least approximated) but also that such fixed points often convey useful information. □

We conclude this section with a return to inductive definitions of sets. Recall from Chapter 4 that an inductive definition of a set A schematically states that A is the least set X such that

$$B \subseteq X \text{ and } \mathcal{R}(X) \subseteq X,$$

or put alternatively

$$B \cup \mathcal{R}(X) \subseteq X.$$

In other words, A is the least pre-fixed point of the map $\Phi(X) := B \cup \mathcal{R}(X)$. Of course, there is no guarantee that there is a *least* pre-fixed point, but if Φ is monotonic on a cpo (or a complete lattice) then the inductive definition must be sound and

$$A = \mu X.B \cup \mathcal{R}(X) = \textsc{lfp}(\Phi).$$

Hence, if we make sure that inductive definitions are monotonic, then the definition must be well-defined. (It defines something and that something is unique).

EXAMPLE 5.12. We continue Example 5.11: Assume that we have a set $I \subseteq C$ of initial configurations and want to characterize the set $R \subseteq C$ of all configurations reachable from some $c \in I$. The set $R \subseteq C$ then is the least set such that

(1) $I \subseteq R$ (i.e. all initial configurations are reachable),
(2) if $X \subseteq R$ then $\textsc{step}(X) \subseteq R$.

Hence, R is the least set X such that

$$I \cup \textsc{step}(X) \subseteq X.$$

Now, the function $\Phi_I(X) := I \cup \textsc{step}(X)$ is clearly monotonic and R therefore must be the least fixed point of Φ_I. That is, the least fixed point of Φ_I is the set of all configurations reachable, in 0 or more steps, from I. □

In the next section we shall see how to characterize the set R constructively under the additional assumption that Φ_I is continuous.

5.3. Kleene's fixed point theorem

Knaster-Tarski's theorem concerns the existence of least and greatest fixed points. We now turn to the problem of computing, or at least approximating, fixed points. The following theorem, due to Kleene, provides a hint on how to compute the least fixed point, in case of a continuous map.

THEOREM 5.13. Let $(A, \leq)$ be a cpo (or a complete lattice) and assume that $f : A \to A$ is continuous. Then $f^\omega(\bot)$ is the least fixed point of f. □

PROOF. We first recall that $f^n(\bot) \leq f^{n+1}(\bot)$ for all $n < \omega$ according to Theorem 4.5. We next prove that $f^\omega(\bot)$ is a fixed point of f, and finally show that it must be the least fixed point. By definition

$$f^\omega(\bot) = \bigvee_{n<\omega} f^n(\bot). \tag{5.3}$$

Hence

$$f(f^\omega(\bot)) = f(\bigvee_{n<\omega} f^n(\bot)).$$

Now since f is continuous, and $f^n(\bot) \leq f^{n+1}(\bot)$ for all $n \leq \omega$

$$\begin{aligned} f(\bigvee_{n<\omega} f^n(\bot)) &= \bigvee_{n<\omega} f(f^n(\bot)) \\ &= \bigvee_{1\leq n<\omega} f^n(\bot) \\ &= f^\omega(\bot) \text{ (since } f^0(\bot) = \bot). \end{aligned}$$

We finally demonstrate that $f^\omega(\bot)$ must be the least fixed point of f. Let x be an arbitrary fixed point of f. Clearly $\bot \leq x$. By monotonicity $f^n(\bot) \leq f^n(x) = x$ for all $n < \omega$. That is, x is an upper bound of $\{f^n(\bot) \mid n < \omega\}$. But by (5.3) $f^\omega(\bot)$ is the *least* upper bound of $\{f^n(\bot) \mid n < \omega\}$; whence $f^\omega(\bot) \leq x$, and $f^\omega(\bot)$ therefore is the least fixed point. □

From Chapter 3 we know that

$$f^{\omega}(\bot) = \bigvee_{n<\omega} f^n(\bot).$$

That is, $\textsc{lfp}(f)$ is the least upper bound of the so-called Kleene sequence,

$$\bot, f(\bot), f^2(\bot), \ldots, f^n(\bot), \ldots$$

In fact, since f is monotonic this is an ascending chain, which means that $\textsc{lfp}(f) = \lim_{n\to\infty} f^n(\bot)$.

Example 5.14. We consider a class of logic formulas which are particularly suited for computation and which provides the basis of the field of *logic programming.* (It also is a subclass of the positive Boolean formulas introduced earlier.) Logic programs typically consist of predicate logic formulas, but for simplicity we consider only the propositional fragment here. A logic program is a set of *definite clauses* (sometimes called Horn clauses) which are logic formula on the form

$$x_1 \wedge \ldots \wedge x_n \to x_0 \quad \text{(with } n \geq 0\text{)}$$

where (in our restricted case) $x_0, \ldots, x_n$ are propositional variables. To emphasize the programming aspect definite clauses are usually written

$$x \leftarrow x_1, \ldots, x_n.$$

A definite clause with $n = 0$ is called a *fact.* A definite program (or simply program) is a set of definite clauses. For instance, let P be the definite program

$$\begin{aligned} &a \leftarrow b, c \\ &b \leftarrow \\ &c \leftarrow b, d \\ &d \leftarrow e \\ &e \leftarrow e \\ &d \leftarrow f \\ &f \leftarrow . \end{aligned}$$

Restricting attention to definite programs has several important consequences which are not possessed by propositional formulas in general. First of all, definite programs cannot be inconsistent; there is always at least one interpretation which is a model of the program (which one?). Moreover, definite programs always have a *least* model called the least Herbrand model (least under set inclusion or the pointwise ordering depending if we represent interpretations as sets or Boolean maps).

Let $\mathit{Var} := \{a, b, c, d, e, f\}$ and consider the space of all interpretations 2^{Var} of Var. In order for an interpretation $\mathfrak{I}$ to be a model of a clause $x_0 \leftarrow x_1, \ldots, x_n \in P$ we require that $x_0 \in \mathfrak{I}$ if $\{x_1, \ldots, x_n\} \subseteq \mathfrak{I}$. Hence the least Herbrand model is the least interpretation satisfying this for all clauses in P. We may formalize this observation by means of an operator $T_P \colon 2^{\mathit{Var}} \to 2^{\mathit{Var}}$ usually called the *immediate consequence operator*,

$$T_P(I) := \{x_0 \mid x_0 \leftarrow x_1, \ldots, x_n \in P \wedge \{x_1, \ldots, x_n\} \subseteq I\}. \tag{5.4}$$

In order for $\mathfrak{I}$ to be a model of P it is required that $T_P(\mathfrak{I}) \subseteq \mathfrak{I}$; i.e. $\mathfrak{I}$ must be a pre-fixed point of T_P, and $\textsc{lfp}(T_P)$ then is the least model of P.

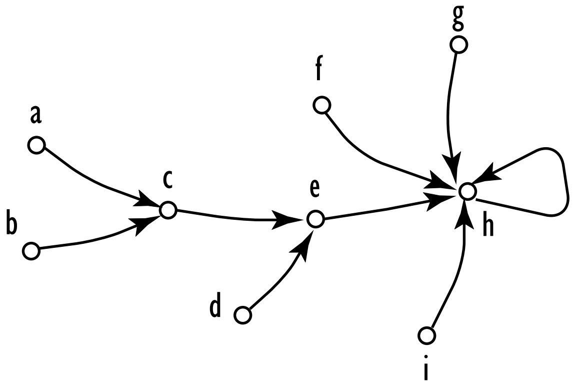


Figure 1. Transition system $(C, \Rightarrow)$

The space of valuations 2^{Var} is a finite complete lattice of sets with bottom element $\emptyset$, and T_P clearly is monotonic; hence continuous since 2^{Var} is finite. (Actually, one can show that it is continuous also when the set of all interpretations is infinite.) Hence $T_P^{\omega}(\emptyset)$ is the least fixed point of T_P.

In this particular example we may compute $\text{LFP}(T_P)$ by a finite prefix of the ordinal powers of T_P.

$$\begin{array}{rcl} T_P^0(\emptyset) & = & \emptyset \\ T_P^1(\emptyset) & = & \{b, f\} \\ T_P^2(\emptyset) & = & \{b, d, f\} \\ T_P^3(\emptyset) & = & \{b, c, d, f\} \\ T_P^4(\emptyset) & = & \{a, b, c, d, f\} \\ T_P^5(\emptyset) & = & T_P^4(\emptyset) \end{array}$$

Hence $\{a, b, c, d, f\}$ is the least model of P. For more information on fixed point semantics of logic programs see e.g. Lloyd [**Llo87**]. □

Example 5.15. Consider the transition system $(C, \Rightarrow)$ depicted in Figure 1. In Example 5.12 we showed that the function

$$\Phi_I(X) := I \cup \text{STEP}(X)$$

was monotonic. Since, 2^C is finite it follows trivially that Φ_I is also continuous. Hence, $\Phi_I^{\omega}(\emptyset)$ must be the least fixed point of Φ_I. The ascending Kleene sequence looks as follows, assuming that $I = \{b, d\}$,

$$\begin{array}{rcl} \Phi_I^0(\emptyset) & = & \emptyset \\ \Phi_I^1(\emptyset) & = & \{b, d\} \\ \Phi_I^2(\emptyset) & = & \{b, c, d, e\} \\ \Phi_I^3(\emptyset) & = & \{b, c, d, e, h\} \\ \Phi_I^4(\emptyset) & = & \Phi_I^3(\emptyset). \end{array}$$

Hence, $\{b, c, d, e, h\}$ are the only configurations reachable from $\{b, d\}$. □

Example 5.16. Consider a transition system $(C, \Rightarrow)$, and let $\text{GOOD} \subseteq C$ be a set of configurations that we hope to reach eventually (i.e. they are good configurations). Now consider the problem of computing the set of all configurations where it is possible to eventually reach a good configuration. We first introduce the operation $\textsc{ex}\colon 2^C \to 2^C$, defined as

$$\textsc{ex}(X) := \{c \in C \mid \exists c' \in X \text{ such that } c \Rightarrow c'\}\,.$$

That is, $\textsc{ex}(X)$ is the set of all configurations where it is possible (but not necessary) to reach X in one step. Hence, $\textsc{ex}(\textsc{good})$ is the set of all configurations that may move to a good configuration in one step, and $\textsc{ex}(\textsc{ex}((\textsc{good}))$ is the set of all configurations that may reach a good configuration in 2 steps, and so forth.

The set $\textsc{ef}(\textsc{good})$ of configurations which may eventually reach $\textsc{good}$ is the least set $X \subseteq C$ such that

- $\textsc{good} \subseteq X$, and
- $\textsc{ex}(X) \subseteq X$.

That is, $\textsc{ef}(\textsc{good})$ is the least set X such that $\textsc{good} \cup \textsc{ex}(X) \subseteq X$. Now $\Phi(X) := \textsc{good} \cup \textsc{ex}(X)$ is monotonic, hence

$$\textsc{ef}(\textsc{good}) = \mu X.\textsc{good} \cup \textsc{ex}(X).$$

If we consider the transition system depicted in Figure 1 again, and if we assume e.g. that $\textsc{good} = \{e, f\}$, then we get the following ascending Kleene sequence

$$\begin{array}{lcl}
\Phi^0(\emptyset) & = & \emptyset \\
\Phi^1(\emptyset) & = & \{e, f\} \\
\Phi^2(\emptyset) & = & \{c, d, e, f\} \\
\Phi^3(\emptyset) & = & \{a, b, c, d, e, f\} \\
\Phi^4(\emptyset) & = & \Phi^3(\emptyset).
\end{array}$$

So far we focused almost entirely on least fixed points, but a monotonic map also has a greatest fixed point. For the sake of simplicity we make the common assumption that our transition system contains no *sinks*, i.e. configurations without outgoing transitions.[1] Consider the following map,

$$\textsc{ax}(X) := \{c \in C \mid \forall c' \text{ if } c \Rightarrow c' \text{ then } c' \in X\}\,.$$

Hence, $\textsc{ax}(X)$ is the set of all configurations where we, by necessity, reach X in a single step no matter what transition is taken.

Assume that we have a set $\textsc{bad}$ of illegitimate configurations, and let $\overline{\textsc{bad}}$ be its complement; i.e. the ok configurations. We now want to characterize a set $\textsc{ag}(\overline{\textsc{bad}})$ of configurations where it is impossible to reach $\textsc{bad}$. That is, if we are in $\textsc{ag}(\overline{\textsc{bad}})$ then we remain in $\textsc{ag}(\overline{\textsc{bad}})$ no matter what transition we make. Hence, we require that

(1) $\textsc{ag}(\overline{\textsc{bad}})$ must be contained in $\overline{\textsc{bad}}$, and
(2) $\textsc{ag}(\overline{\textsc{bad}})$ must be contained in $\textsc{ax}(\overline{\textsc{bad}})$, and
(3) $\textsc{ag}(\overline{\textsc{bad}})$ must be contained in $\textsc{ax}(\textsc{ax}(\overline{\textsc{bad}}))$, etc

That is, $\textsc{ag}(\overline{\textsc{bad}})$ is the largest set X, such that

(1) $X \subseteq \overline{\textsc{bad}}$, and
(2) $X \subseteq \textsc{ax}(X)$.

That is, X should be the largest satisfying

$$X \subset \overline{\textsc{bad}} \cap \textsc{ax}(X).$$

[1] This is actually no restriction; if there are configurations without outgoing transitions we can always add a cyclic transition from the configuration to itself.

Ideally we want to establish an equality, that is

$$\textsc{ag}(\overline{\textsc{bad}}) = \nu X.\overline{\textsc{bad}} \cap \textsc{ax}(X).$$

Now let $\Phi(X) := \overline{\textsc{bad}} \cap \textsc{ax}(X)$. The greatest fixed point of Φ can be constructed in the same fashion as the least fixed point via a *descending* Kleene sequence starting from the top element; in our case C. Assume that $\textsc{bad} = \{c, f\}$. Then

$$\begin{array}{rcl}
\Phi^0(C) & = & C \\
\Phi^1(C) & = & \{a, b, d, e, g, h, i\} \\
\Phi^2(C) & = & \{d, e, g, h, i\} \\
\Phi^3(C) & = & \Phi^2(C).
\end{array}$$

Hence, $\{d, e, g, h, i\}$ is the set of all configurations where it is impossible to reach an illegitimate configuration.

When reasoning about transition systems least fixed points typically express that something (may) eventually happen, while greatest fixed points typically express that something (may) hold forever. The operations EX and AX (as well as EF and AG) are operators of CTL (Computation Tree Logic); a temporal logic used to reason about temporal properties of transition systems (see e.g. Clarke et al. [**CGP99**] for details). □

All of the previous examples involved finite lattices in which case it is always possible to reach the least (and greatest) fixed points after a finite prefix of the ascending (resp. descending) Kleene sequence. However, this is not always the case, as illustrated in the following example.

EXAMPLE 5.17. Consider the following equation over strings of some alphabet Σ with $1 \in \Sigma$

$$w = 1w. \tag{5.5}$$

Such equations occur frequently when studying semantics of perpetual processes; for instance, in reactive, distributed and concurrent systems. The equation may for example model the behavior of a process w which sends a message 1 and then behaves like w again, in infinity.

It should be clear that no finite string satisfies (5.5) since the righthand side is always one character longer than the lefthand side if w is finite. That is, there are no solutions in Σ^*. Hence, consider instead the set of all possibly infinite strings $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. We may rewrite (5.5) as a fixed point equation

$$w = \textsc{one}(w) \tag{5.6}$$

where

$$\textsc{one}(x) := \{(0 \mapsto 1)\} \cup \{(n + 1 \mapsto a) \mid (n \mapsto a) \in x\}.$$

Then

$$\begin{array}{rcl}
\textsc{one}^0(\emptyset) & = & \emptyset \\
\textsc{one}^1(\emptyset) & = & \{0 \mapsto 1\} \\
\textsc{one}^2(\emptyset) & = & \{0 \mapsto 1, 1 \mapsto 1\} \\
\textsc{one}^3(\emptyset) & = & \{0 \mapsto 1, 1 \mapsto 1, 2 \mapsto 1\} \\
 & \vdots &
\end{array}$$

with the limit

$$\textsc{one}^\omega(\emptyset) = \bigcup_{n<\omega} \textsc{one}^n(\emptyset) = \{0 \mapsto 1, 1 \mapsto 1, 2 \mapsto 1, 3 \mapsto 1, \ldots\}.$$

Now $\textsc{one}\colon \Sigma^\infty \to \Sigma^\infty$ can be shown to be chain-continuous; for every chain $w_0 < w_1 < w_2 < \ldots$ we have that,

$$\textsc{one}(\bigcup_{i\geq 0} w_i) = \bigcup_{i\geq 0} \textsc{one}(w_i) \tag{5.7}$$

Hence, $\textsc{one}^\omega(\emptyset)$ is the (unique) least solution to (5.5). □

Exercises

5.1 Give an example of a function which is monotone but not inflationary and vice versa.

5.2 Prove that functional composition preserves monotonicity. That is, if $f\colon A \to A$ and $g\colon A \to A$ are monotone, then so is $f \circ g$. Note: $(f \circ g)(x) := f(g(x))$.

5.3 Prove that functional composition preserves continuity.

5.4 Define a map Φ such that $\Phi^\omega(\bot)$ is the Fibbonacci function,

$$\{0 \mapsto 1, 1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3, 4 \mapsto 5, 5 \mapsto 8, \ldots\}.$$

Show that the map is chain-continuous.

5.5 Give an example of a complete lattice $(A, \leq)$ and a monotonic map $f\colon A \to A$ such that $f^\omega(\bot)$ is not a fixed point of f.

5.6 Prove that a continuous map on a complete lattice (or a cpo) is always monotone.

5.7 Prove that if $f\colon A \to A$ is monotone and A is finite, then f must be continuous.

5.8 Show that if there exists an $n \in \mathbb{N}$ such that x is a unique fixed point of f^n (i.e. for every y such that $f^n(y) = y$ it holds that $x = y$), then x must be a fixed point of f.

5.9 Let $(A, \leq)$ be a complete lattice (or a cpo) and $f\colon A \to A$ monotone. Show, using transfinite induction, that $f^\alpha(\bot) \leq \textsc{lfp}(f)$ for every ordinal α.

CHAPTER 6

Finite automata on infinite words

In this chapter we study extensions of classical automata theory; more precisely automata defining languages of infinite words. Such automata have gained considerable interest in the study of systems that typically exhibit infinite behaviors, e.g. reactive systems such as operating systems or telephone switches.

The automata that we consider here have applications in the verification of properties of reactive systems such as safety, liveness and fairness. A system exhibits a set of behaviours which can be abstracted into an infinite sequence of observations; typically an infinite sequence of events or values of (some) state variables. A system is then identified with the set of all its possible behaviours, i.e. typically an infinite set of infinite words. Now a property of the system can likewise be represented as the set of all allowed behaviours; also typically an infinite set of infinite words. To verify that the system exhibits a property amounts to showing that all behaviors of the system are contained in the specification. That is, we want to show that the language L_1 representing all possible system behaviors is contained in the language L_2 representing all *allowed* system behaviors; that is $L_1 \subseteq L_2$. The main focus of the chapter is formalisms for representing infinite words, and effective procedures for checking language containment. For a more elaborate presentation see e.g. Thomas [**Tho90**].

6.1. Basic notation

In what follows A denotes a finite alphabet, and A^* denotes the set of all finite words over A; that is, the set of all functions $w\colon \{0,\ldots,n\} \to A$, including the case $n < 0$, which is called the empty word (and usually written simply ϵ). Finite words are typically denoted by u, v, w, while sets of finite words are denoted U, V, W. The standard operations on sets of finite words (concatenation $U.V$, union $U + V$ and finite repetition U^*) will be used.

By A^ω we denote the set of all countably infinite words over A; that is the set of functions $\alpha\colon \omega \to A$. In what follows we refer to countably infinite words as ω-words, and use symbols such as α, β, γ to denote them. If U is a set of finite words over A then U^ω denotes the set of infinite repetitions of finite strings in U, that is

$$U^\omega = \{\alpha \in A^\omega \mid \exists w_1 w_2 \ldots \in U, \alpha = w_1 w_2 \ldots\}.$$

Sets of ω-words are called ω-languages and we use L to denote such languages.

6.2. Büchi automata

Finite automata for ω-words are almost identical to the finite word counterpart. The only difference is the notion of *final*, or *accepting*, state, which (obviously)

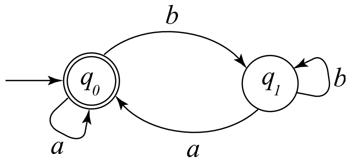


FIGURE 1. Büchi automaton recognizing $(b^*a)^\omega$

cannot be the same. There are different types of automata with different acceptance conditions, the simplest of which is Büchi automata [**Büc60**].

DEFINITION 6.1. A Büchi automaton $\mathcal{B}$ over an alphabet A is a tuple (Q, q_0, Δ, F) where Q is a finite set of *states*, $q_0 \in Q$ an *initial state*, $\Delta \subseteq Q \times A \times Q$ a *transition relation* and $F \subseteq Q$ a set of *accepting*, or *final*, states. □

A Büchi automaton is called deterministic iff for any $q \in \Delta$ and $a \in A$ there is a unique q' such that $(q, a, q') \in \Delta$.

A *run* of a Büchi automaton $\mathcal{B} = (Q, q_0, \Delta, F)$ on an ω-word α is an infinite word of states $\sigma \in Q^\omega$ such that $\sigma(0) = q_0$ and $(\sigma(i), \alpha(i), \sigma(i+1)) \in \Delta$ for all $i \geq 0$.

Let $\inf(\sigma)$ be the set of all states that occur infinitely often in the ω-word σ. Then an ω-word α is *accepted* by a Büchi automaton $\mathcal{B}$ iff there is a run σ on α such that $F \cap \inf(\sigma) \neq \emptyset$. That is, if there is a run where some accepting state occurs infinitely often. The language of a Büchi automaton $\mathcal{B}$ is defined as the set of all such words,

$$\mathcal{L}(\mathcal{B}) = \{\alpha \mid \mathcal{B} \text{ accepts } \alpha\}.$$

An ω-language definable by some Büchi automaton is said to be *Büchi recognizable*.

EXAMPLE 6.2. Figure 1 depicts a (deterministic) Büchi automaton that accepts ω-words of a:s and b:s in which each occurrence of b is eventually followed by an a. That is, it accepts words with infinitely many a:s, or using regular expression syntax, the ω-language $(b^*a)^\omega$. □

Like finite regular languages, Büchi recognizable languages are closed under union as well as intersection.

THEOREM 6.3. If $L_1, L_2 \subseteq A^\omega$ are Büchi recognizable languages, then so are $L_1 \cup L_2$ and $L_1 \cap L_2$. □

PROOF. The proof (or rather construction) is similar to that of finite regular languages; that is, we construct a product automaton. However, some care has to be exercised to handle the accepting states in the case of intersection. We show how to construct an automaton accepting the intersection of Büchi recognizable languages L_1 and L_2.

Let $\mathcal{B}_1 = (Q_1, q_1, \Delta_1, F_1)$ and $\mathcal{B}_2 = (Q_2, q_2, \Delta_2, F_2)$ be Büchi automata with ω-languages L_1 and L_2. We construct a product automaton

$$\mathcal{B} = (Q_1 \times Q_2 \times \{0,1,2\}, (q_1, q_2, 0), \Delta, Q_1 \times F_2 \times \{2\})$$

where $((x_1, y_1, n_1), a, (x_2, y_2, n_2)) \in \Delta$ iff

- $(x_1, a, x_2) \in \Delta_1$ and $(y_1, a, y_2) \in \Delta_2$,
- and where in addition

- $n_2 = 1$ if $n_1 = 0$ and $x_2 \in F_1$,
- $n_2 = 2$ if $n_1 = 1$ and $y_2 \in F_2$,
- $n_2 = 0$ if $n_1 = 2$,
- $n_2 = n_1$ otherwise.

The third component of the state ensures that the accepting states of $\mathcal{B}$ correspond to runs where accepting states of *both* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ are visited infinitely often. The third component is initially 0; then becomes 1 when we reach a state corresponding to an accepting state of $\mathcal{B}_1$ and is incremented again when reaching a state corresponding to an accepting state of $\mathcal{B}_2$ after which it is reset to 0. □

The following propositions can be verified using techniques similar to those for finite regular languages, and illustrate two techniques of constructing ω-languages.

PROPOSITION 6.4. If $U \subseteq A^*$ is regular, then U^ω is Büchi recognizable. □

PROPOSITION 6.5. If $U \subseteq A^*$ is regular and $L \subseteq A^\omega$ is Büchi recognizable then so is $U.L$. □

Based on these results we can now state and prove the following theorem which also provides a characterization of Büchi recognizable languages in terms of finite regular languages.

THEOREM 6.6. An ω-language L is Büchi recognizable iff there is some $n \geq 0$ and regular languages of finite words, U_i and V_i where $1 \leq i \leq n$, such that

$$L = \bigcup_{i=1}^{n} U_i.(V_i)^\omega.$$

□

PROOF. Let $W(q, q')$ denote the language of finite (non-empty) words accepted in any run starting in state q and ending in state q'. The language $W(q, q')$ clearly is regular for all q, q' of a Büchi automaton.

($\Rightarrow$) Assume that $\alpha \in L$. Since L is Büchi recognizable there must be some Büchi automaton (Q, q_0, Δ, F) and some run σ where some $q \in F$ occurs infinitely often. That is $\alpha \in W(q_0, q).(W(q, q))^\omega$. Since there are only finitely many (accepting) states, it follows that L can be expressed as a finite union of ω-languages.

($\Leftarrow$) By Propositions 6.4 and 6.5 and Theorem 6.3 it follows that L must be Büchi recognizable. □

Hence any Büchi recognizable language can be written in the form

$$U_1.(V_1)^\omega + \ldots + U_n.(V_n)^\omega$$

where $U_1, \ldots, U_n, V_1, \ldots, V_n$ are regular expressions. Such expressions are called *ω-regular expressions.*

The so-called *non-emptiness* problem is decidable for Büchi automata; that is, the problem of checking whether the language defined by a Büchi automaton contains some string. It is just a matter of finding a path in the automaton from the initial state to one of the final states, and then find a non-empty path from the final state to itself.

THEOREM 6.7. The non-emptiness problem for Büchi automata is solvable in O(m+n) time, where m is the number of states, and n the number of transitions. □

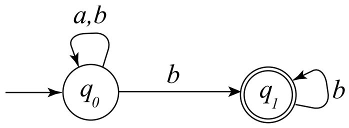


FIGURE 2. Büchi automata recognizing $(a+b)^*b^\omega$

PROOF. Proceed as follows: First determine the set of all states Q' reachable from the initial state. Then find all strongly connected components (SCCs) in Q'. Finally check if some component contains an accepting state. All steps can be solved in time O(m+n). □

The non-emptiness-problem may first appear insignificant. However, it is an important component in checking language containment. Let L_1 and L_2 be ω-languages over an alphabet A, and let $\overline{L_2}$ denote the complement of L_2, that is $A^\omega \setminus L_2$. Then

$$L_1 \subseteq L_2 \text{ iff } L_1 \cap \overline{L_2} = \emptyset.$$

Now Büchi-recognizable languages are closed under intersection (see Theorem 6.3) and we have already shown an effective way of intersecting two Büchi automata. We have also just shown that the non-emptiness-problem is decidable. Hence, if Büchi automata are closed under complement, and if we can find some effective way of computing a Büchi automaton recognizing the complement of another Büchi automaton, then we we have an effective way of checking language containment. Now Büchi recognizable languages are in fact closed under complementation, as shown by Büchi.

THEOREM 6.8. If L is Büchi recognizable, then so is $A^\omega \setminus L$. □

However, complementing a Büchi automaton is a tricky matter in contrast to computing the complement of automata over finite words. For a deterministic finite automaton accepting a language W of finite words it is straightforward to compute a new DFA that accepts W's complement, $A^* \setminus W$; simply change all non-accepting states into accepting states and vice versa. Moreover, since nondeterministic automata can be effectively transformed into deterministic finite automata it is “straightforward” to take the complement also of nondeterministic finite automata.[1] For Büchi automata the situation is more intricate. As will be seen, there are nondeterministic Büchi automata that have no equivalent deterministic counterpart. Hence, nondeterministic Büchi automata are strictly more expressive than deterministic Büchi automata. Moreover, it turns out that deterministic Büchi automata are not closed under complementation, as illustrated by the following example.

EXAMPLE 6.9. Figure 2 depicts a nondeterministic Büchi automaton that accepts the language $(a+b)^*b^\omega$; i.e. ω-words with only a finite number of a:s; or in other words, the complement of the automaton in Figure 1. □

[1] The translation from a nondeterministic to a deterministic automaton may of course lead to an exponential blow-up in the size of the automaton.

We prove that there is no deterministic Büchi automaton that accepts the language $(a+b)^*b^\omega$ from our example. Let $W \subseteq A^*$ be a regular language and let

$$\lim W = \{\alpha \in A^\omega \mid \forall m \geq 0\ \exists n > m \text{ such that } \alpha(0)\ldots\alpha(n) \in W\}.$$

That is, $\lim W$ is the set of all ω-words that have an infinite number of prefixes in W. For example

- $\lim b(ab)^+ = b(ab)^\omega$
- $\lim a^*b = \emptyset$
- $\lim a(b+c)^* = a(b+c)^\omega$

We can now prove the following characterization of deterministically Büchi recognizable languages.

Theorem 6.10. An ω-language L is deterministically Büchi recognizable iff there is some regular language $W \subseteq A^*$ such that $L = \lim W$. □

Proof. Let $\mathcal{B} = (Q, q_0, \Delta, F)$ be a deterministic Büchi automaton, and let $W \subseteq A^*$ be the language recognized by the DFA $\mathcal{A} = (Q, q_0, \Delta, F)$. Then,

$$\begin{array}{lll}\mathcal{B} \text{ accepts } \alpha & \text{iff} & \text{some run of } \mathcal{B} \text{ on } \alpha \text{ enters } F \text{ infinitely often} \\ & \text{iff} & \forall m \geq 0\ \exists n > m \text{ such that } \mathcal{B} \text{ reaches } F \text{ on } \alpha(0)\ldots\alpha(n) \\ & \text{iff} & \forall m \geq 0\ \exists n > m \text{ such that } \mathcal{A} \text{ reaches } F \text{ on } \alpha(0)\ldots\alpha(n) \\ & \text{iff} & \forall m \geq 0\ \exists n > m \text{ such that } \alpha(0)\ldots\alpha(n) \in W \\ & \text{iff} & \alpha \in \lim W\end{array}$$

□

Theorem 6.11. The language $(a+b)^*b^\omega$ is not deterministically Büchi recognizable. □

Proof. Assume that there is some deterministic Büchi automaton that recognizes $(a+b)^*b^\omega$. If so, there must be some regular language W such that $\lim W = (a+b)^*b^\omega$. Since $b^\omega \in \lim W$ we know that there must be some $n_1 \geq 0$ such that $b^{n_1} \in W$. Now for this n_1 there must be some n_2 such that $b^{n_1}ab^{n_2} \in W$, since $b^{n_1}ab^\omega \in \lim W$. Proceeding in the same way we see that $b^{n_1}ab^{n_2}a\ldots ab^{n_i} \in W$ for all $i \geq 1$. Hence, by the definition of lim, $\lim W$ must contain the infinite word $b^{n_1}ab^{n_2}a\ldots$ which contains infinitely many occurrences of a contradicting the fact that $\lim W = (a+b)^*b^\omega$. □

6.3. Muller automata

In this section we consider an alternative to Büchi automata called Muller automata [**Mul63**]. We show that the Büchi recognizable languages are equivalent to the deterministically Muller recognizable languages, and that there is an effective procedure for transforming Büchi automata into deterministic Muller automata and vice versa.[2] Moreover, deterministic Muller automata are effectively closed under complementation (as well as union and intersection), thus providing an indirect way of complementing Büchi recognizable languages.

Muller automata differ from Büchi automata only in the power of the acceptance condition.

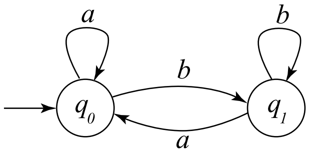
[2] One can also show that Büchi recognizable languages are equivalent to nondeterministically Muller recognizable languages. (That is, deterministic and nondeterministic Muller automata are equally expressive.) But this equivalence is of minor importance for our purposes.

Definition 6.12. A Muller automaton $\mathcal{B}$ over an alphabet A is a tuple (Q, q_0, Δ, F) where Q is a finite set of *states*, $q_0 \in Q$ an *initial state*, $\Delta \subseteq Q \times A \times Q$ a *transition relation* and $F \subseteq 2^Q$ a set of sets of *accepting states*. □

An ω-word α is accepted by a Muller automaton $\mathcal{B}$ iff there exists a run σ on α such that $\inf(\sigma) \in F$. That is, if the set of all states that occur infinitely often in σ is contained in F.

A Muller automaton is called deterministic iff for any $q \in \Delta$ and $a \in A$ there is a unique q' such that $(q, a, q') \in \Delta$. To emphasize this we usually write $\Delta(q, a) = q'$ when the automaton is deterministic.

Example 6.13. Consider the following deterministic automaton,



With the acceptance condition $\{\{q_1\}\}$ the automaton accepts the language $(a + b)^*b^\omega$. That is, words with only finitely many occurrences of a (and hence infinitely many occurrences of b). Choosing instead the acceptance condition $\{\{q_0\}, \{q_0, q_1\}\}$ we obtain the complement of $(a+b)^*b^\omega$. That is, all words containing infinitely many occurrences of a, and hence either finitely many, or infinitely many, occurrences of b. □

Every deterministic Muller automaton, has an equivalent (possibly nondeterministic) Büchi automaton, as first shown by McNaughton [**McN66**].

Theorem 6.14. If L is deterministically Muller recognizable, then L is (possibly non-deterministically) Büchi recognizable. □

Proof. Let $\mathcal{B} = (Q, q_0, \Delta, F)$ be a deterministic Muller automaton accepting the ω-language L. We show how to construct an equivalent non-deterministic Büchi automaton.

First note that every accepting run of $\mathcal{B}$ must be of the form

$$w_0 w_1 w_2 w_3 \ldots,$$

where $w_0 \in Q^*$ and where there is some $F_j \in F$ such that w_k contains exactly the states in F_j for all $k > 0$. The basic idea in the construction of the Büchi automaton accepting L is to "guess" when we enter w_1; that is, when all subsequent states are from F_j and all states in F_j are visited infinitely often. For this purpose we introduce a set of additional states of the form (q, P, j) where $q \in F_j$ and $P \subseteq F_j$. Whenever $\mathcal{B}$ contains a transition from p to q where $q \in F_j \in F$, we introduce an additional transition (with the same label) from p to the new state $(q, \emptyset, j)$. Now if our guess is correct there has to be a path in $\mathcal{B}$ that starts with q and contains all states in F_j and leads back to q. The second component P of a state (q, P, j) is used to record the states that we have visited "so far" in such a cycle. If we reach a state $(q', F_j \setminus \{q\}, j)$ and $\mathcal{B}$ contains a transition from q' to q, we "close the cycle" by a transition from $(q', F_j \setminus \{q\}, j)$ to $(q, \emptyset, j)$ (with the same label). The accepting states of this extended (Büchi) automaton are the states of the form $(q, \emptyset, j)$, for all $j \in \{1, \ldots, |F|\}$.

Hence let $\mathcal{A} = (Q^B, q_0^B, \Delta^B, F^B)$ be a Büchi automaton with the state set $Q^B = Q \cup (Q \times 2^Q \times \{1, \dots, |F|\})$, with initial state $q_0^B = q_0$, where $F^B = \{(q, \emptyset, j) \mid q \in F_j \in F\}$, and where Δ^B is the least relation such that,

- $(p, a, q) \in \Delta^B$ if $\Delta(p, a) = q$,
- $(p, a, (q, \emptyset, j)) \in \Delta^B$ if $q \in F_j$ and $\Delta(p, a) = q$,
- $((p, P, j), a, (q, P \cup \{q\}, j)) \in \Delta^B$ if $\Delta(p, a) = q$ and $P \cup \{q\} \subset F_j$, and
- $((p, P, j), a, (q, \emptyset, j)) \in \Delta^B$ if $\Delta(p, a) = q$ and $P \cup \{q\} = F_j$.

□

Constructing a (possibly non-deterministic) Muller automaton that accepts the same language as a Büchi automaton is easy; the automaton will have the same states (including initial state), and the same transitions. The only difference is the acceptance condition: if the Büchi automaton contains an accepting state q then the acceptance condition of the Muller automaton must contain any F_i such that $q \in F_i$. However, constructing a *deterministic* Muller automaton is more difficult. The standard approach (powerset construction) used to transform an NFA (for finite words) into a DFA does not work, since Muller automata have a much more involved acceptance condition. The following result is due to McNaughton but the proof (which builds on a generalized powerset construction) is due to Safra [**Saf88**].

THEOREM 6.15. If L is (nondeterministically) Büchi recognizable, then L is deterministically Muller recognizable. □

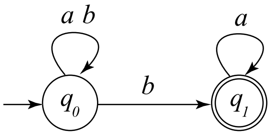
Rather than actually proving the theorem we give the construction, due to Safra, that produces an equivalent deterministic Muller automaton. In what follows, let $\mathcal{A} = (Q, q_0, \Delta, F)$ be a Büchi automaton. By a *Safra tree* over Q we mean a finite, ordered tree with nodes from the set $\{1, \dots, 2 \cdot |Q|\}$, where each node is labeled by some $R \subseteq Q$, and where leaves may be marked as final. Siblings in the tree are assumed to have disjoint labels and the union of their labels should be a proper subset of the parent's label.

PROOF OF THEOREM 6.15. The idea of the Safra construction is to build a deterministic automaton $(Q^S, q_0^S, \Delta^S, F^S)$ where Q^S are Safra trees over Q and where q_0^S is the singleton tree 1 labeled $\{q_0\}$. The transitions Δ^S of the new automaton are obtained in four steps. Let s be a Safra tree and $a \in \Sigma$, then $s' = \Delta^S(s, a)$ is obtained as follows

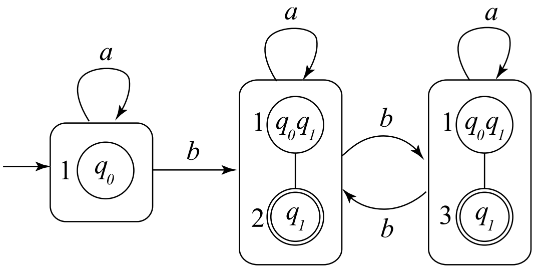
(1) For each node n (labeled S_n) in s, apply the powerset construction on the input a, i.e. relabel n by $\{q' \mid \exists q \in S_n, (q, a, q') \in \Delta\}$. Unmark n if it was marked as final.
(2) For each node n in the new tree, add a new child (picking a free node from $\{1, \dots, 2|Q|\}$) labeled by all accepting states in S_n. Mark these nodes as final,
(3) Remove the state q from a node (and all its descendants) if q appears in an existing sibling node. Remove the whole node(s) if labeled by the empty set (unless it is the root node),
(4) For each node n, remove all of its descendants if the union of labels of the children equals S_n. If so, mark n as final. Let s' be the resulting tree.

A set F of Safra trees is in F^S iff there exists a node name that appears in every tree of F and at least one such node is marked as final. □

EXAMPLE 6.16. Consider the following nondeterministic automaton.

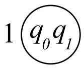


The Safra construction will contain an initial state labeled $\{q_0\}$. The resulting deterministic Muller automaton looks as follows:

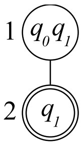


The acceptance condition contains two singleton sets, containing the middle and the rightmost Safra tree.

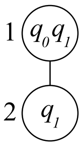
We inspect in some detail some of the transitions of the resulting Muller automaton. Consider the initial state (labeled $\{q_0\}$) on input b. Step (1) of the construction yields the intermediate tree



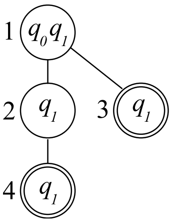
Expanding this tree according to step (2) yields



Step (3) and (4) impose no further changes, so we introduce this as a new state (reachable from the initial state via b). Now consider this new state on input a. Step (1) yields

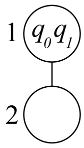


Step (2) yields:

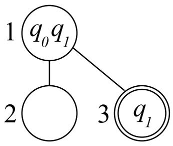


In step (3) node 3 is removed since q_1 already appears in a sibling node, and in step (4) node 4 is removed since its labeling equals the labeling of node 2 (which should be marked as final).

Finally consider the second node on input b instead. Step (1) yields



Step (2) yields



Finally node 2 is removed (since it is labeled by the empty set). The transition back from the rightmost Safra tree to the middle one is completely analogous except that nodes 2 and 3 are exchanged. □

The fact that a nondeterministic Büchi automata can be transformed into an equivalent deterministic Muller automaton, and back again, facilitates an alternative way of constructing the complement of a Büchi automaton. Taking the complement of a deterministic Muller automaton is namely straightforward (like complementing a DFA).

Theorem 6.17. If (Q, q_0, Δ, F) is a deterministic Muller automaton accepting $L \subseteq A^\omega$, then $(Q, q_0, \Delta, 2^Q \setminus F)$ accepts $A^\omega \setminus L$, i.e. the complement of L. □

For instance, the complement of the automaton in Example 6.13 with the acceptance condition $\{\{q_1\}\}$ is the same automaton with the new acceptance condition $\{\emptyset, \{q_0\}, \{q_0, q_1\}\}$. Actually, we may always drop $\emptyset$ since any infinite run must involve at least one state.

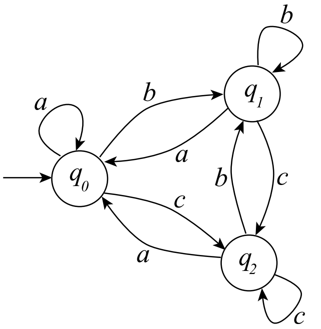
Exercises

6.1 One important application of ω-regular languages is in the verification of properties of discrete event systems. Consider a finite alphabet $\{a, b, c\}$ of possible observations from a system, i.e. the system behavior is a subset of $\{a, b, c\}^\omega$. Express the following properties of ω-languages by Büchi automata (deterministic if possible)

(1) b happens eventually;
(2) a never happens;
(3) b happens infinitely often;
(4) whenever a happens b happens eventually;
(5) b never happens twice in a row;
(6) the subsequence ab happens only finitely many times.

6.2 Give ω-regular expressions for the ω-regular languages in the previous exercise.

6.3 Consider the following deterministic automaton,

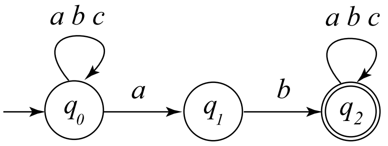


What Muller acceptance condition is required to express the property "if a occurs infinitely often, then b occurs infinitely often"?

6.4 Muller recognizable ω-languages are closed under intersection. Consider two deterministic Muller automata $\mathcal{A}_1$ and $\mathcal{A}_2$ on the finite alphabet A, and assume that $\mathcal{A}_1$ accepts L_1 and $\mathcal{A}_2$ accepts L_2. Describe how to construct a deterministic Muller automaton that accepts $L_1 \cap L_2$?

6.5 In the same setting as the previous exercise: describe how to construct a Muller automaton that accepts $L_1 \cup L_2$?

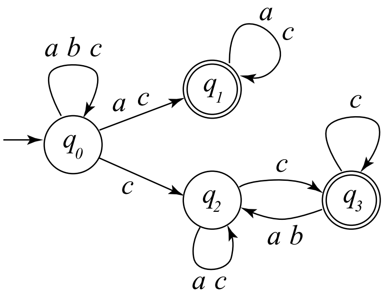
6.6 Consider the following Büchi automaton:



Translate it into a deterministic Muller automaton by the Safra construction, and finally complement this automaton. What is the resulting language?

6.7 Translate the resulting deterministic Muller automaton from the previous exercise into a Büchi automaton.

6.8 Consider the following Büchi automaton.



Construct, by means of the Safra construction, an equivalent deterministic Muller automaton.

Bibliography

[Acz77] P. Aczel. An introduction to inductive definitions. In *Handbook on Mathematical Logic*, pages 739–782. North-Holland, 1977.

[Büc60] J.R. Büchi. Weak Second-order Arithmetic and Finite Automata. *Z. Math. Logik Grundlag. Math.*, 6:66–92, 1960.

[Bir67] G. Birkhoff. *Lattice Theory.* American Mathematical Society, 3rd edition, 1967.

[CGP99] E. Clarke, O. Gumberg, and D. Peled. *Model Checking.* MIT Press, 1999.

[Grä78] G. Grätzer. *General Lattice Theory.* Academic Press, 1978.

[Gri00] R. Grimaldi. *Discrete and Combinatorial Mathematics.* Addison-Wesley, 4th edition, 2000.

[Hal61] P. Halmos. *Naive Set Theory.* van Nostrand, 1961.

[Llo87] J. Lloyd. *Foundations of Logic Programming.* Springer Verlag, 2nd edition, 1987.

[LNS82] J.-L. Lassez, V.L. Nguyen, and E.A. Sonenberg. Fixed point theorems and semantics: A folk tale. *Inform. Process. Lett.*, 14:112–116, 1982.

[McN66] R. McNaughton. Testing and Generating Infinite Sequences by a Finite Automaton. *Inform. and Control*, 9:521–530, 1966.

[Mul63] D.E. Muller. Infinite Sequences and Finite Machines. In *Proc. 4th Ann. IEEE Symp. on Switching Circuit Theory and Logical Design*, pages 3–16, 1963.

[Saf88] S. Safra. On the Complexity of ω-automata. In *Proc 29th Ann. IEEE Symp. on Foundations of Computer Science*, pages 319–327, 1988.

[Tar55] A. Tarski. A lattice-theoretic fixpoint theorem and it´s application. *Pacific J. Math.*, 5:285–309, 1955.

[Tho90] W. Thomas. Automata on infinite objects. In *Handbook on Theoretical Computer Science*, pages 135–191. Elsevier, 1990.