1

Ordered Sets

Order, order, order – it permeates mathematics, and everyday life, to such an extent that we take it for granted. It appears in many guises: first, second, third, ... ; bigger versus smaller; better versus worse. Notions of progression, precedence and preference may all be brought under its umbrella. Our first task is to crystallize these imprecise ideas and to formalize the relationship of 'less-than-or-equal-to'. Besides presenting examples and basic properties of ordered sets, this chapter also introduces the diagrams which make order theory such a pictorial subject and give it much of its character.

Ordered sets

What exactly do we mean by order? More mathematically, what do we mean by an ordered set?

1.1 Order. Each of the following miscellany of statements has something to do with order.

(a) $0 < 1$ and $1 < 10^{23}$.

(b) Two first cousins have a common grandfather.

(c) $22/7$ is a worse approximation to π than 3.141592654.

(d) The planets in order of increasing distance from the sun are Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto.

(e) Neither of the sets $\{1,2,4\}$ and $\{2,3,5\}$ is a subset of the other, but $\{1,2,3,4,5\}$ contains both.

(f) Given any two distinct real numbers a and b, either a is greater than b or b is greater than a.

Order is not a property intrinsic to a single object. It concerns comparison between pairs of objects: 0 is smaller than 1; Mars is further from the sun than Earth; a seraphim ranks above an angel, etc. In mathematical terms, an ordering is a binary relation on a set of objects. In our examples, the relation may be taken to be 'less than' on $\mathbb{N}$ in (a), 'is a descendant of' on the set of all human beings in (b) and $\subseteq$ on the subsets of $\{1,2,3,4,5\}$ (or of $\mathbb{N}$) in (e).

What distinguishes an order relation from some other kind of relation? Firstly, ordering is transitive. From the facts that $0 < 1$ and $1 < 10^{23}$ we can deduce that $0 < 10^{23}$. Mars is nearer the sun than Saturn and Saturn is nearer than Neptune, so Mars is nearer than Neptune.

Secondly, order is antisymmetric: 5 is bigger than 3 but 3 is not bigger than 5. It is on these two properties – transitivity and antisymmetry – that the theory of order rests.

Order relations are of two types: strict and non-strict. Outside mathematics, the strict notion is more common. The statement 'Charles is taller than Bruce' is generally taken to mean 'Charles is strictly taller than Bruce', with the possibility that Charles is the same height as Bruce not included. Mathematicians usually allow equality and write, for instance, $3 \leqslant 3$ and $3 \leqslant 22/7$. We shall deal mainly with non-strict order relations.

Finally a comment about comparability. Statement (f) asserts that, for the ordering $<$ on the real numbers, any two distinct elements can be compared. This property is possessed by many familiar orderings, but it is not universal. For example, there certainly exist human beings A and B such that A is not a descendant of B and B is not a descendant of A. Non-comparability also arises in (e).

1.2 Definitions. Let P be a set. An **order** (or **partial order**) on P is a binary relation $\leqslant$ on P such that, for all $x, y, z \in P$,

(i) $x \leqslant x$,

(ii) $x \leqslant y$ and $y \leqslant x$ imply $x = y$,

(iii) $x \leqslant y$ and $y \leqslant z$ imply $x \leqslant z$.

These conditions are referred to, respectively, as **reflexivity**, **antisymmetry** and **transitivity**. A set P equipped with an order relation $\leqslant$ is said to be an **ordered set** (or **partially ordered set**). Some authors use the shorthand **poset**. Usually we shall be a little slovenly and say simply 'P is an ordered set'. Where it is necessary to specify the order relation overtly we write $\langle P; \leqslant \rangle$. On any set, $=$ is an order, the **discrete order**. A relation $\leqslant$ on a set P which is reflexive and transitive but not necessarily antisymmetric is called a **quasi-order** or, by some authors, a **pre-order**. An order relation $\leqslant$ on P gives rise to a relation $<$ of **strict inequality**: $x < y$ in P if and only if $x \leqslant y$ and $x \neq y$. It is possible to re-state conditions (i)–(iii) above in terms of $<$, and so to regard $<$ rather than $\leqslant$ as the fundamental relation; see Exercise 1.1.

Other notation associated with $\leqslant$ is predictable. We use $x \leqslant y$ and $y \geqslant x$ interchangeably, and write $x \nleqslant y$ to mean '$x \leqslant y$ is false', and so on. Less familiar is the symbol $\parallel$ used to denote non-comparability: we write $x \parallel y$ if $x \nleqslant y$ and $y \nleqslant x$.

We later deal systematically with the construction of new ordered sets from existing ones. However, there is one such construction which it is convenient to have available immediately. Let P be an ordered set and let Q be a subset of P. Then Q inherits an order relation from P; given $x, y \in Q$, $x \leqslant y$ in Q if and only if $x \leqslant y$ in P. We say in these circumstances that Q has the **induced order**, or, when we wish to be more explicit, the order **inherited from** P.

1.3 Chains and antichains. Let P be an ordered set. Then P is a **chain** if, for all $x, y \in P$, either $x \leqslant y$ or $y \leqslant x$ (that is, if any two elements of P are comparable). Alternative names for a chain are **linearly ordered set** and **totally ordered set**. At the opposite extreme from a chain is an antichain. The ordered set P is an **antichain** if $x \leqslant y$ in P only if $x = y$. Clearly, with the induced order, any subset of a chain (an antichain) is a chain (an antichain).

Let P be the n-element set $\{0, 1, \dots, n-1\}$. We write $\mathbf{n}$ to denote the chain obtained by giving P the order in which $0 < 1 < \cdots < n-1$ and $\overline{\mathbf{n}}$ for P regarded as an antichain. Any set S may be converted into an antichain $\overline{S}$ by giving S the discrete order.

1.4 Order-isomorphisms. We need to be able to recognize when two ordered sets, P and Q, are 'essentially the same'. We say that P and Q are (**order-**) **isomorphic**, and write $P \cong Q$, if there exists a map φ from P onto Q such that $x \leqslant y$ in P if and only if $\varphi(x) \leqslant \varphi(y)$ in Q. Then φ is called an **order-isomorphism**. Such a map φ faithfully mirrors the order structure. It is necessarily bijective (that is, one-to-one and onto): using reflexivity and antisymmetry of $\leqslant$, first in Q and then in P,

$$\begin{aligned} \varphi(x) = \varphi(y) &\iff \varphi(x) \leqslant \varphi(y) \ \&\ \varphi(y) \leqslant \varphi(x) \\ &\iff x \leqslant y \ \&\ y \leqslant x \\ &\iff x = y. \end{aligned}$$

On the other hand, not every bijective map between ordered sets is an order-isomorphism: consider, for example, $P = Q = \mathbf{2}$ and define φ by $\varphi(0) = 1$, $\varphi(1) = 0$.

Being a bijection, an order-isomorphism $\varphi \colon P \to Q$ has a well-defined inverse, $\varphi^{-1} \colon Q \to P$. It is easily seen that this is also an order-isomorphism.

We hinted in 1.1 at a variety of situations in which order is present. In 1.2 we developed the vocabulary for treating these examples systematically. We conclude this section by presenting formally the important orderings carried by some fundamental mathematical structures.

1.5 Number systems. The set $\mathbb{R}$ of real numbers, with its usual order, forms a chain. Each of $\mathbb{N}$ (the natural numbers $\{1, 2, 3, \dots\}$), $\mathbb{Z}$ (the integers) and $\mathbb{Q}$ (the rational numbers) also has a natural order making it a chain. In each case this order relation is compatible with the arithmetic structure in the sense that the sum and product of two elements strictly greater than zero is also greater than zero.

We denote the set $\mathbb{N} \cup \{0\}$ $(= \{0, 1, 2, \dots\})$ by $\mathbb{N}_0$. Endowed with the order in which $0 < 1 < 2 < \dots$, the set $\mathbb{N}_0$ becomes the chain known in set theory as ω. It is order-isomorphic to $\mathbb{N}$: the successor function $n \mapsto n^+ := n + 1$ from $\mathbb{N}_0$ to $\mathbb{N}$ is an order-isomorphism. A different order on $\mathbb{N}_0$ is defined as follows. Write $m \preccurlyeq n$ if and only if there exists $k \in \mathbb{N}_0$ such that $km = n$ (that is, m divides n). Then $\preccurlyeq$ is an order relation. Of course, $\langle \mathbb{N}_0; \preccurlyeq \rangle$ is not a chain. Yet another order on $\mathbb{N}_0$ is introduced in 1.22 for use in Chapters 8 and 9.

1.6 Families of sets. Let X be any set. The powerset $\wp(X)$, consisting of all subsets of X, is ordered by set inclusion: for $A, B \in \wp(X)$, we define $A \leqslant B$ if and only if $A \subseteq B$.

Any subset of $\wp(X)$ inherits the inclusion order. Such a family of sets might be specified set-theoretically. For example, it might consist of all finite subsets of an infinite set X. More commonly, families of sets arise where X carries some additional structure. For instance, X might have an algebraic structure – it might be a group, a vector space, or a ring. Each of the following is an ordered set under inclusion:

- the set of all subgroups of a group G (denoted $\operatorname{Sub} G$), and the set of all normal subgroups of G (denoted $\mathcal{N}$-$\operatorname{Sub} G$);
- the set of all subspaces of a vector space V (denoted $\operatorname{Sub} V$);
- the set of all subrings of a ring R, and the set of all ideals of R.

Families of sets also occur in other mathematical contexts. For example, let $(X; \mathcal{T})$ be a topological space. We may consider the families of open, closed, and clopen (meaning simultaneously closed and open) subsets of X as ordered sets under inclusion. Finally we note a more inbred member in this class of ordered sets which is of fundamental importance later. This is the family $\mathcal{O}(P)$ of down-sets of an ordered set P; it is introduced in 1.27.

Essentially the same ordered set as $\langle \wp(X); \subseteq \rangle$ manifests itself in a different form, as the set of predicates on X. A **predicate** is a statement taking value T (true) or value F (false). More precisely, a predicate on X is a function from X to $\{\mathsf{T}, \mathsf{F}\}$; here we don't distinguish between different ways of specifying the same function. For example, the map $p \colon \mathbb{R} \to \{\mathsf{T}, \mathsf{F}\}$ given by $p(x) = \mathsf{T}$ if $x \geqslant 0$ and $p(x) = \mathsf{F}$ if $x < 0$ is a predicate on $\mathbb{R}$, which can alternatively be specified by $p(x) = \mathsf{T}$ if $|x-1| \leqslant |x+1|$ and F otherwise. We write $\mathbb{P}(X)$ for the set of predicates on X and order it by implication: for $p, q \in \mathbb{P}(X)$,

$$p \Rrightarrow q \text{ if and only if } \{\, x \in X \mid p(x) = \mathbf{T} \,\} \subseteq \{\, x \in X \mid q(x) = \mathbf{T} \,\}.$$

Define a map $\varphi \colon \mathbb{P}(X) \to \wp(X)$ by $\varphi(p) := \{\, x \in X \mid p(x) = \mathbf{T} \,\}$. Then φ is an order-isomorphism between $\langle \mathbb{P}(X); \Rrightarrow \rangle$ and $\langle \wp(X); \subseteq \rangle$. The notion of a predicate is fundamental in logic and in computer science.

Examples from social science and computer science

Order and ordered structures enter into computer science, and also into social science, in many ways and on many different levels. Our aim in this section is to give a glimpse of *why* this should be so, rather than to explain in detail *how* order theory is employed in applications. This discussion supplies motivation for some of the theory we develop later on, but much of it is not used directly. We look first at ways in which ordered sets arise in social science.

1.7 Ordered sets in the humanities and social sciences. Below is a pot-pourri of examples to indicate how ordered sets occur in the social sciences and elsewhere. Each of these areas of application has led to the investigation of ordered sets of special types.

An **interval order** on a set X is an order relation such that there is a mapping φ of the points of X into subintervals of $\mathbb{R}$ such that, for $x < y$ in X, the right-hand endpoint of $\varphi(x)$ is less than the left-hand endpoint of $\varphi(y)$. Interval orders model, for example, the time spans over which animal species are found or the occurrence of styles of pottery in archaeological strata. A variant on the definition requires all the image intervals to be of the same length, with problems of inexact measurement in mind.

The problem of amalgamating the expressed preferences of a group of individuals to arrive at a consensus is of concern to selection committees, market researchers, psephologists and many others. More explicitly, given m objects and rankings of them by n individuals specified by n chains, how should a chain be constructed which best reflects the individuals' collective preferences? A **social choice function** assigns to any n–tuple of rankings a single ranking which defines a consensus, according to specified criteria. A famous theorem, due to K. Arrow, asserts that there is a set of criteria which are very natural but mutually incompatible. This paradoxical result set off an avalanche of research on social choice theory.

The problem of scheduling a collection of activities or events arises in many different contexts, such as manufacturing and conference planning. Many such problems involve precedence constraints. For example, certain stages in the assembly of a car must precede others and a conference organizer is likely to have to schedule certain lectures before others. The computational complexity of a scheduling problem depends critically on the order relation which describes the precedence constraints.

Order enters into the classification of objects on two rather different levels. The first is illustrated by our introductory example of the arrangement of the planets into a hierarchical list according to their distance from the sun and by Figure 9.1 which classifies certain ordered sets according to various criteria. On a deeper level, the rather new discipline of **concept analysis** provides a powerful technique for classifying and for analysing complex sets of data. From a set of objects (to take a simple example, the planets) and a set of attributes (for the planets, perhaps large/small, moon/no moon, near sun/far from sun), concept analysis builds an ordered set which reveals inherent hierarchical structure and thence natural groupings and dependencies among the objects and the attributes. Chapter 3 gives a brief introduction to concept analysis.

We now turn to order-theoretic ideas relating to computer science. Our focus in this book is limited to certain aspects of this burgeoning subject in which ordered structures provide useful mathematical models and in this introductory chapter we concentrate on the description of models for some particularly important datatypes. In each case, a relation $\geqslant$ serves to capture the notion of 'is at least as informative as', with the precise interpretation depending on the context. But before presenting examples of such information orderings we need to clarify how computations are to be viewed.

1.8 Programs. Speaking simplistically, a program to perform a computation takes a certain input and, the user hopes, returns a corresponding output. The input and output data may come from many different datatypes, such as natural numbers, strings, lists, sets, and so forth. The term **state** is used to denote an assignment, to the variables used by a program, of values drawn from the appropriate datatypes. The program **terminates** if it transforms any given state before its execution to a state afterwards; the initial and final states may be regarded as incorporating the input and output data. Frequently, the result of a computation will be generated step by step, with additional information being gained at each stage. Non-termination of a program naturally arises where only partial information towards the solution is output in finite time. A program is **deterministic** if, starting from a given initial state, it will terminate in the same final state each time it is run. Non-determinism can occur where the program's specification allows for more than one valid solution. For example, a program to compute an integer y such that $y^2 = x$ might start in the state $x = 9$ and terminate in either the state $y = 3$ or $y = -3$.

We now give three examples of order relations on datatypes. In 1.12 we look at the features these examples have in common.

1.9 Binary strings. Let Σ^* be the set of all finite binary strings, that is, all finite sequences of zeros and ones; the empty string is included. Adding the infinite sequences, we get the set of all finite or infinite sequences, which we denote by Σ^{**}. We order Σ^{**} by putting $u \leqslant v$ if and only if $u = v$ or u is a finite initial substring (the technical term is **prefix**) of v. Thus, for example, $0100 < 010011$, $010 \parallel 100$ and $10101 < 101010\ldots$ (the infinite string of alternating ones and zeros). Strings may be thought of as information encoded in binary form: the longer the string, the greater the information content. Further, given any string v, we may think of elements u with $u < v$ as providing approximations to v. In particular, any infinite string is, in a sense we shall later need to make precise, the limit of its finite initial substrings. Obviously this example can be generalized by considering strings whose elements are drawn from an arbitrary alphabet of symbols.

1.10 Partial maps. Let X and Y be non-empty sets and $f\colon X \to Y$ a map. Then f may be regarded as a recipe which assigns a member $f(x)$ of Y to each $x \in X$. Alternatively, and equivalently, f is determined by its **graph**, namely $\operatorname{graph} f := \{\, (x, f(x)) \mid x \in X \,\}$, a subset of $X \times Y$. If the values of f are given on some subset S of X, we have partial information towards determining f. Formally, we define a **partial map** from X to Y to be a map $\sigma\colon S \to Y$, where $\operatorname{dom}\sigma$, the domain of σ, is a subset S of X; here $S = \varnothing$ is allowed. If $\operatorname{dom}\sigma = X$, then σ is a map (or, for emphasis, a **total map**) from X to Y. The set of partial maps from X to Y is denoted $(X \multimap\!\!\to Y)$; it contains all total maps from X to Y and all partial determinations of them. The elements of $(X \multimap\!\!\to X)$ are called partial maps on X. We order $(X \multimap\!\!\to Y)$ as follows: given $\sigma, \tau \in (X \multimap\!\!\to Y)$, define $\sigma \leqslant \tau$ if and only if $\operatorname{dom}\sigma \subseteq \operatorname{dom}\tau$ and $\sigma(x) = \tau(x)$ for all $x \in \operatorname{dom}\sigma$. Equivalently, $\sigma \leqslant \tau$ if and only if $\operatorname{graph}\sigma \subseteq \operatorname{graph}\tau$ in $\wp(X \times Y)$. Note that a subset G of $X \times Y$ is the graph of a partial map if and only if

$$(\forall s \in X)\,\big(((s,y) \in G \;\&\; (s,y') \in G) \implies y = y'\big).$$

In a (non-terminating) computation to determine a map $f\colon X \to Y$, we may think of f as being built up from tokens of information, each of which is an element σ of $(X \multimap\!\!\to Y)$ with finite domain and which partially specifies f and where $\sigma < f$ in the ordering defined above on

$(X \multimap\!\!\rightarrow Y)$. In the other direction, suppose we are given a collection $\mathcal{F}$ of elements of $(X \multimap\!\!\rightarrow Y)$. Is there a map f such that we have $\sigma \leqslant f$ for each $\sigma \in \mathcal{F}$? Clearly, the tokens must not supply conflicting messages about the putative f. For example, f cannot exist if $\mathcal{F}$ contains elements σ and τ such that, for some $x \in X$, we have $(x, y) \in \operatorname{graph} \sigma$ and $(x, y') \in \operatorname{graph} \tau$, where $y \neq y'$. We say that a subset $\mathcal{F}$ of $(X \multimap\!\!\rightarrow Y)$ is **consistent** if, for any finite subset $\mathcal{G}$ of $\mathcal{F}$, there exists $\rho \in (X \multimap\!\!\rightarrow Y)$ (but not necessarily in $\mathcal{F}$) such that $\sigma \leqslant \rho$ for all $\sigma \in \mathcal{G}$. It is easy to see that, so long as $\mathcal{F}$ is a consistent subset of $(X \multimap\!\!\rightarrow Y)$, there exists a map $f\colon X \to Y$ such that $\sigma \leqslant f$ for all $\sigma \in \mathcal{F}$. Consistency is treated in a more general setting in Chapter 9.

Let us now compare two programs P and Q having a common set X of initial states and set Y of final states. Suppose first that they are deterministic but do not necessarily terminate. As above, we may view them as given by partial maps σ_P and σ_Q on X. Assume that $\sigma_P \leqslant \sigma_Q$, so that $\operatorname{dom} \sigma_P \subseteq \operatorname{dom} \sigma_Q$ and $\sigma_P(x) = \sigma_Q(x)$ for all $x \in \operatorname{dom} \sigma_P$. Thus from any input state from which P terminates, Q does too and in the same final state that P does; additionally Q may terminate from initial states from which P fails to do so. Thus Q can achieve everything that P can (more if $\sigma_P < \sigma_Q$, since then Q terminates from at least one state from which P fails to terminate). We write $P \sqsubseteq Q$ if $\sigma_P \leqslant \sigma_Q$. Widening this to (possibly) non-deterministic programs P and Q we say that Q **refines** P, and write $P \sqsubseteq Q$, if 'Q is at least as good as P' in the sense that Q achieves, at least, what P does. A particular situation in which this may arise is the progressive unfolding of a while-loop. Refinement of a non-deterministic program may result in one which is deterministic, or closer to being deterministic. Refinement of a non-terminating program may yield one which terminates more often.

1.11 Intervals in $\mathbb{R}$ and exact real arithmetic. The statement that some computed quantity r equals 1.35 correct to 2 decimal places may be re-expressed as the assertion that r lies in a particular interval in $\mathbb{R}$. We may accordingly treat the collection of all intervals $[\underline{x}, \overline{x}]$ (where $-\infty \leqslant \underline{x} \leqslant \overline{x} \leqslant \infty$) as a set P of approximations to the real numbers, with a smaller interval giving a tighter bound than a larger one and so being more informative. The intervals for which $\underline{x} = \overline{x}$ correspond to exact values. The set P carries a very natural order: for $x = [\underline{x}, \overline{x}]$ and $y = [\underline{y}, \overline{y}]$ define $x \leqslant y$ if and only if $\underline{x} \leqslant \underline{y}$ and $\overline{y} \leqslant \overline{x}$. Then $x \leqslant y$ means that y represents (or contains) at least as much information as x.

Traditionally, a floating-point representation of real numbers has been used in numerical computation. But this has inherent disadvantages: rounding errors are endemic and errors in the input data, resulting from its inexact representation, get propagated. An important goal, therefore, is to implement in a suitable high-level programming language the datatype for $\mathbb{R}$, and the basic arithmetic operations and elementary functions on it, in an efficient way and without rounding errors. (More precisely, the objective is to accomplish this within the framework of effective computation, in the sense of computability theory.) Building on various partially successful attempts, A. Edelat has developed an approach starting from the observation that a real number can be viewed as (being determined by) a shrinking nested sequence of intervals with rational endpoints. The move to rational numbers here is, of course, motivated by the fact that exact calculations can be performed with rationals. A survey of Edelat's work on this and other computational models can be found in [18].

1.12 Information orderings. In each of Examples 1.9–1.11 the order relation captures a notion of 'is more informative than': $x \leqslant y$ has an interpretation such as 'y is more defined than x' or 'y is a better approximation than x'. In each case, we have a notion of a **total object** (a completely defined, or idealized, element). These total objects are the infinite binary strings in the first example, the total maps in the second and the 1-point intervals in the third. An important feature of these examples from a computational point of view is that in each case the total objects may be realized in a natural way as limits of partial objects. Further, in Σ^{**} and in $(\mathbb{N} \multimap \mathbb{N})$, for example, we have approximations by partial objects which are in some sense 'finite': respectively, finite strings or partial maps which have finite domain. In general, a finite object should be one which encodes a finite amount of information.

We conclude our informal introduction to the occurrence of order in computer science with a few general remarks, to widen the perspective and to hint at themes picked up in later chapters.

1.13 Semantics and semantic domains. Running through our discussion of ordered sets in computer science is the idea of a semantic domain: a mathematical structure through which one can describe, analyse and reason about the behaviour of entities such as datatypes, programs and specifications. This use here of 'semantic domain' is generic, and very broad. In Chapter 9, the term 'domain' acquires a narrower, more technical meaning, as an ordered set of a special sort. In a domain, certain elements are to be viewed as partial, or incompletely specified, and each element is required to be the limit (in an appropriate order-theoretic sense) of special elements, designated finite. The ordered structures presented in 1.9 and 1.10 are examples of semantic domains. In fact, both these are domains in the sense of the formal definition in 9.7. Example 1.11 is a structure of a similar but more general type.

Domains can alternatively be viewed as logical structures. From this perspective, elements of domains are seen as being determined by assertions about them (or propositions they satisfy) and are modelled by sets of tokens of information. Defined in a precise way, either as a special class of ordered sets or, equivalently, as information systems, domains have a mathematical theory worthy of study in its own right, and not just for its computational significance.

In computer science, different styles of semantic modelling are favoured depending on what aspect of computing is being studied. As we hint in our discussion of Galois connections in Chapter 7, a semantics focussing on laws governing the actions of programs may be a useful view. In operational semantics, programs are modelled by the actions they perform on a computer (idealized rather than actual). With denotational semantics, by contrast, the emphasis is on *what* programs do, rather than on *how* a computer executes them. Programs are represented, for example, by functions or relations and can be studied through their representations without the distracting detail needed to describe their implementation.

Chapters 8 and 9 deal with classes of semantic domains rich enough to provide denotational models for complex computational processes, including recursion. The term **recursive** is used of an algorithm defined in terms of itself or a program which calls itself. An example is the specification of the factorial function on $\mathbb{N}_0$ via the recursive formula

$$\mathsf{fact}(k) = \begin{cases} 1 & \text{if } k = 0, \\ k\mathsf{fact}(k-1) & \text{if } k > 0. \end{cases}$$

Recursion is a very powerful computational tool and so a central issue in computer science has been the development of mathematical models of programming which can accommodate it. Specifically, what is required is a denotational framework within which theorems can be proved which assert the existence of computable solutions to (suitable) objects given by recursive definitions. The theory of fixpoints, which we introduce in a mathematical way in Chapter 8, and of domains, have in part been developed with this end in view.

Diagrams: the art of drawing ordered sets

One of the most useful and attractive features of ordered sets is that, in the finite case at least, they can be 'drawn'. To describe how to represent ordered sets diagrammatically, we need the idea of covering.