Chapter 1

Sets and Maps

1.1 Sets

1.1.1 Sets, Arrays, Records, etc.

Consider the digits 0,1,2,3,4. In how many ways can we choose two of them?

(a) 25 (b) 20 (c) 15 (d) 10 (e) 4 (f) 1.

The answer depends on how the choices are to be organized. In fact we can obtain any of these answers by organizing the choices as (a) an array, (b) a chain, (c) a multiset, (d) either a set or a cycle, (e) an interval, or (f) either an initial or a final segment.

When the choices are made sequentially and we "sample with replacement," that is, permit multiple choices, we have the notion of an *array*. For brevity of enumeration we shall denote the array "3 followed by 2" as simply 32. The 25 two-element arrays of the digits 0-4 are all two-digit numerals in radix 5 without zero-suppression, namely $00, 01, 02, \ldots, 43, 44$.

There are two noteworthy aspects here. First, an element can appear more than once, as in 00, 11, 22, 33, and 44. A structure with this property is called a "multiset." Second, order of elements matters: 01 is a distinct choice from 10. A structure with this property is called *ordered*. Thus an array is an ordered multiset.

A *chain* is can be viewed either as an array with no repetitions ("sampling without replacement"), or as an ordered set. Deleting the five arrays containing repetitions leaves twenty chains, namely 01, 02, 03, 04, 10, 12, 13, 14, 20, 21, 23, 24, 30, 31, 32, 34, 40, 41, 42, and 43. In combinatorics and statistics a chain is also called a *permutation* of m things chosen from n, or of n things taken m at a time.

A *multiset*, sometimes called a bag, can be viewed either as an array whose order is immaterial, or as a set with repetitions permitted. The fifteen multisets are 00, 01, 02, 03, 04, 11, 12, 13, 14, 22, 23, 24, 33, 34, and 44. This time we did not delete the unmentioned arrays but instead combined pairs of them, writing 01 for what we could equally well have written as 10.

Another description of multiset, one that matches up with how we wrote the 15 multisets in the example, is that a multiset is an array chosen with replacement from a chain, preserving the order of the chain. That is, we go along the chain from beginning to end and as we come to each element we take it as often as we please. In this description the array 10 was deleted rather than identified with 01. A multiset drawn from a set X can also be described as a function $n : X \to \mathcal{N}$ mapping each element x of X to a natural number $n(x)$ giving the number of instances of x. The number of elements of the multiset is then given by $\sum_{x \in X} n(x)$.

A *set* is an array for which order is immaterial and no repetitions are allowed. (So we *eliminate* 00 but we *identify* 01 with 10.) The ten two-element *subsets* of 0-4 are 01, 02, 03, 04, 12, 13, 14, 23, 24, and 34.

A *cycle* is a chain looped back on itself. No repetitions are allowed, and although order matters going around the cycle there is no start (no element is distinguished as the first). Cycles with at most two elements are the same as sets, whence we obtain answer (d), 10, for either sets or cycles. But while there are only 10 3-element subsets of 0-4, each of these, e.g. 134, can be organized as a chain in 6 ways, namely 134, 143, 314, 341, 413, and 431, and as a cycle in 2 ways, namely 134 and 143. So we have twenty 3-element cycles, namely 234, 243, 134, 143, 124, 142, 123, 132, 034, 043, 024, 042, 023, 032, 014, 041, 013, 031, 012, 021.

The last three organizations of choices assume the source is a chain, here the chain 01234. An *interval* is a consecutive set. The four intervals of length 2 are 01, 12, 23, and 34. There are $n-m+1$ m-element intervals in an n-element chain. An *initial segment* of X is an interval starting at the beginning of X; there is one of each length, the length 2 initial segment of 01234 being 01. Dually a *final segment* ends at the end, here 34. If we require that the chosen and unchosen sets each form an interval then we have two ways to choose two elements, namely the initial or the final segment.

There is one more kind of entity that we will find useful, the *record* or *family*. Whereas an array is *homogeneous* in that it draws all its elements from the same set, a record is *heterogeneous* in that it draws the element at any given location $i \in L$ from a set X_i specific to that location. We denote the set of all possible such records as $\Pi_{i\in L}X_i$, called the *product* of the X_i's. When $L = \{1, 2, \ldots, n\}$ the product may be written as $X_1 \times X_2 \times \ldots \times X_n$, and we then call such records n-tuples. For example the elements of $\mathbf{Z} \times \mathbf{R}$ are all those 2-tuples (also called *pairs*) $\langle i, x\rangle$ for which i is an integer and x is a real, $\mathbf{Z}$ and $\mathbf{R}$ being the standard notations for those sets. When all X_i's are equal to X we can write $\Pi_{i\in L}X_i$ as simply X^L.

The universally used notation for 01 is $\{0, 1\}$, the braces indicating that it is a set. Arrays, at least when considered as tuples, are often notated $\langle 0, 1\rangle$ or just 0,1; the same notation applies to records. APL writes such an array as 0 1 while LISP writes (01) for the corresponding list. Cycles are commonly notated (021) or $(0, 2, 1)$. Intervals are denoted $[2, 5]$, their first and last elements inclusive. If the last element is to be excluded we write $[2, 5)$ which for the integers is the same as $[2, 4]$ and $(1, 4]$, while for the reals $(2, 5]$ denotes the *half-open* interval $\{x \mid 2 < x \leq 5\}$. Context is required to distinguish the open interval $(2, 5)$ from the similarly notated cycle.

The arrays we have been speaking of are one-dimensional arrays. Synonyms for such are vector, row, column, list, string, and (not necessarily infinite) sequence. In computer science strings usually consist of characters, while lists are commonly taken to behave like the lists of the programming language LISP.

1.1.2 Properties of Aggregation

(This and the remaining subsections of this section are not essential but may be of interest to some readers.)

The notion of array can be correlated with properties of *and* in the expression "choose 3 *and* 2," which we can think of as the operation of *aggregation*. This expression denotes a different choice of array from "2 *and* 3", the array 23, so "*and*" is not commutative. It also differs from "3 *and* 2 *and* 2", the array 322, so "*and*" is not idempotent either. On the other hand "(3 *and* 2) *and* 4" denotes the same array as does "3 *and* (2 *and* 4)," namely 324. That is, "*and*" is associative and so the parentheses are redundant.

The notion of chain modifies "*and*" to make it idempotent while leaving it noncommutative. With multisets it is the other way round: "*and*" now becomes commutative but not idempotent, the behavior of arithmetic +. With sets we have both, which along with associativity make *and* the binary operation of a *semilattice*, a notion we shall take up again in the chapter on ordered sets.

The least abstract organization is the array, which can be defined as a function $a : L \to X$ where L is a set of locations taken to be an initial segment $0, 1, 2, \ldots$ of the natural numbers and X is the set being selected from. An interesting consequence of using functions for representing aggregates is that although the "*and*" operation combining choices ("choose 3 *and* 2") is neither commutative nor idempotent, it is associative. If we drop even associativity we obtain a data structure no longer representable as a function in this way, namely ordered binary trees with the choices installed at the leaves.

In the absence of associativity it becomes a little trickier to enumerate the possible choices in the presence of either or both of commutativity and idempotence. Assuming both, there are altogether fifty choices of

3 digits from 0-4, thirty choices of which are 0(12), (01)2, (02)1, and similarly for the other nine ways to choose 3 things from 5, and the remaining twenty choices of which are 1(12), (12)2, and similarly for the other nine ways to choose 2 things from 5. We do not have 1(22) because idempotence rules out 22, but 1 and 12 are different so idempotence does not rule out 1(12).

1.1.3 Enumeration Formulas

In forming an array there are n ways to make each of the m choices. These ways multiply to yield a total of n^m arrays of m choices from n things. In our example $m = 2$ and $n = 5$, whence $5^2 = 25$ arrays.

There are $n!/(n-m)! = n(n-)(n-2)\ldots(n-m+1)$ chains of m choices from n, notated P^n_m. The total number of chains that can be formed from n things is $\sum_{m=0}^{m=n} P^n_m = K_n$ where $K_0 = 1$ and $K_n = 1 + nK_{n-1}$ (almost the recurrence for $n!$); starting from $n = 0$, $K_n = 1, 2, 5, 16, 65, 326, \ldots$.

Counting the six possible sizes of subset, all told there are $1 + 5 + 10 + 10 + 5 + 1 = 32 = 2^5$ subsets of the five digits. There are $\frac{n!}{m!(n-m)!}$ sets of m choices from n, notated $\binom{n}{m}$, pronounced "n choose m," the number of *combinations* of n things taken m at a time. The total number of subsets of n things is given by $\sum_{m=0}^{m=n} \binom{n}{m} = 2^n$.

Counting the six possible sizes of cycle, $1 + 5 + 10 + 20 + 30 + 24 = 90$ cycles can be formed from the the five digits. There are P^n_m/m cycles of m choices from n. From $n = 0, 1, 2, 3, 4, 5, 6, \ldots$ things we obtain respectively $1, 2, 4, 9, 25, 90, 416, \ldots$ cycles.

The number of ordered binary trees with $m + 1$ leaves is given by C_m, the m-th *Catalan number*, which for $m = 0, 1, 2, 3, 4$ is respectively $m = 1, 1, 2, 5, 14$. The number of ordered binary trees of m items drawn from n is $n^m C_{m-1}$, or $5^3.C_2 = 250$ when $m = 3$ and $n = 5$.

1.1.4 Partitions

In the above we did not pay attention to the organization of what was not chosen. We might reasonably have imposed the same conditions on the chosen and nonchosen halves. We call this a *partition* of X into 2 blocks. More generally we may partition X into m blocks.

Whereas we described an array as a function $a : L \to X$ *from* array locations, we may dually describe a partition of X into a collection B of blocks as a function $b : X \to B$ *to* partition blocks. Forbidding repetitions in subsets amounts to requiring that $a : L \to X$ be *injective*, meaning that if $i \neq j$ in L then $a[i] \neq a[j]$. The dual of this is to forbid empty blocks by requiring that $f : X \to B$ be *surjective*, meaning that for every block b in B there is an x in X such that $f(x) = b$.

Ignoring order of subsets amounts to treating L as "anonymous" (but still distinct) locations. The corresponding abstraction for partitions regards B as forming a set rather than an array; this distinction is sometimes made by saying the blocks are labeled (array of blocks) or unlabeled (set of blocks).

We leave most of the analysis of these situations to the reader. There are two cases worth special mention, both assuming unlabeled blocks. When the blocks are nonempty sets we have the standard notion of an *equivalence relation* as a partition of n things into m blocks. The number of such is written $\left\{ {n \atop m} \right\}$, called a *Stirling number of the second kind.* The corresponding Stirling number of the *first* kind, written $\left[{n \atop m} \right]$, gives the number of partitions when the blocks are organized as empty cycles.

Summing the second kind over all block counts yields $\sum_{m=0}^{m=n} \left\{ {n \atop m} \right\} = B_n$, the n-th *Bell number*, $B_n = 1, 1, 2, 5, 15, 52, 203, \ldots$ counting the number of *partitions* of n things, hence the number of equivalence relations on n things. And $\sum_{m=0}^{m=n} \left[{n \atop m} \right] = n!$, the n-th *factorial*, $n! = 1, 1, 2, 6, 24, 120, 720, \ldots\ldots$, counting the number of *permutations* of n things.

1.2 Maps

1.2.1 Binary Relations

An L-ary *relation* is a subset of $\prod_{i \in L} X_i$. In particular a binary relation $R : X \to Y$ *from* X *to* Y, also written $X \xrightarrow{R} Y$, is a subset of $X \times Y$. Thus if X is the set of all Irish and Y the set of all Scots, the father-of relation from X to Y is the set of all pairs $\langle x, y \rangle$ such that x is Irish, y a Scot, and x the father of y. We call X the *domain* of R, and Y its *codomain.*

We abbreviate $\langle x, y \rangle \in R$ to xRy, and the conjunction xRy and ySz to $xRySz$. For example if D is the set of all donkeys and M the set of all mules, then the father-of relation from D to M is the set of all pairs $\langle d, m \rangle$ such that d is a donkey, m is a mule, and d is the father of m. When $Y = X$ we call $R \subseteq X \times X$ a *square*[1] relation *on* the set X.

The *composition* of a relation $R : X \to Y$ with a relation $S : Y \to Z$ is the relation $R; S : X \to Z$ (also written $X \xrightarrow{R} Y \xrightarrow{R} Z$) defined as those pairs $\langle x, z \rangle$ satisfying $xRySz$ for some $y \in Y$. We shall call pairs of relations that match up in this way *consecutive.* For example the composition of the mother-of relation from X to Y with the father-of relation from Y to Z is the paternal-grandmother-of relation from X to Z. This is independent of whether X, Y, Z are distinct sets or all the same set.

Composition is *associative*, that is, $R; (S; T) = (R; S); T$ for all consecutive $W \xrightarrow{R} X \xrightarrow{R} Y \xrightarrow{R} Z$. This follows because both ways of associating the composition lead to the relation consisting of all pairs $\langle w, z \rangle$ such that there exist x and y satisfying $wRxSyTz$. (Check: What properties of these two quantifiers and the conjunctions implicit in $wRxSyTz$ are being used here?)

The *identity* or *equality* relation $I_X : X \to X$ is the diagonal subset $\langle x, x \rangle$ of X^2. We usually write xI_Xy as $x = y$, with the subscript understood. The composites $X \xrightarrow{I} X \xrightarrow{R} Y$ and $X \xrightarrow{R} Y \xrightarrow{I} Y$ both equal R, called the two *identity conditions.*

The *converse* $R^{\smile} : Y \to X$ of a relation $R : X \to Y$ is the set of pairs $\langle y, x \rangle$ such that $\langle x, y \rangle$ is in R. For example the converse of the parent-of relation is the child-of relation. The *complement* $R^{-} : X \to Y$ of $R : X \to Y$ is $X \times Y \; - \; R$, the set of pairs $\langle x, y \rangle$ in $X \times Y$ that are not in R. For example the complement of the $\leq$ relation on the reals is the $>$ relation.

The *right residual* $R{\to}T : Y \to Z$ (also $R \backslash T$) of $R : X \to Y$ *under* $T : X \to Z$ is the unique relation $S : Y \to Z$ satisfying $R; S \subseteq T$ if and only if $S \subseteq R{\to}T$. Equivalently $R{\to}T$ is the greatest $S : Y \to Z$ satisfying $R; S \subseteq T$. For example let `loves:PEOPLE→OFFICERS` specify which people love which officers, and let `pays:PEOPLE→OBLIGATIONS` specify which people pay which obligations. Then `loves→pays` denotes the relation from officers to obligations consisting of those pairs $\langle x, y \rangle$ such that x is an officer and y is an obligation and every person that loves officer x pays obligation y, e.g. x might be "the king" and y might be "taxes". The dual notion $T{\leftarrow}S$ of *left* residuation (with R, S, T connecting X, Y, Z as before) is the greatest R such that $R; S \subseteq T$. Residuation is as close as one can come to having an inverse of composition analogous to division as an inverse of multiplication. Unlike multiplication composition is not associative, whence the need for two "divisions."

The following attributes of a square binary relation R in the left column satisfy the corresponding conditions on the right for all x, y, and z. We abbreviate "xRy and yRz" to "$xRyRz$".

[1] Unlike matrices, for which the term "square" is standard, there is no standard term for this property of binary relations, and one might equally well call them endorelations. More often than not, binary relations are tacitly assumed to be square.

empty	$\neg(xRy)$
reflexive	xRx
irreflexive	$\neg(xRx)$
identity	$xRy \leftrightarrow x = y$
transitive	$xRyRz \rightarrow xRz$
symmetric	$xRy \rightarrow yRx$
antisymmetric	$xRyRx \rightarrow x = y$
clique	xRy

For any given X, three of these attributes are each satisfied by exactly one binary relation on X, namely empty, identity, and clique, written respectively 0_X, $1'_X$, and 1_X. As sets of pairs these are respectively the empty set, the set of all pairs (x, x), and the set of all pairs (x, y), for $x, y \in X$. As square $X \times X$ matrices these are respectively the matrix of all 0's, the matrix with 1's on the leading diagonal and 0's off the diagonal, and the matrix of all 1's.

Each of these attributes holds of R if and only if it holds of its converse $\breve{R}$, defined by $xRy \leftrightarrow y\breve{R}x$. This extends to Boolean combinations of these attributes, those formed using "and," "or," and "not," such as "reflexive and either not transitive or antisymmetric". For example we will be defining certain concepts such as partial order and equivalence relation, via such Boolean combinations, mainly using "and." Thus if R is a partial order so is $\breve{R}$, and similarly for the other concepts we define. This is the principle of duality. Its importance will emerge gradually as we make use of it.

When X is empty so is X^2, whence there is only one binary relation on X, namely the empty relation. Now all the attributes listed above are *universal* in the sense that the variables in their definitions are universally quantified: "for all x, y, z;" none are existentially quantified, as in "there exists w." By convention, when X is empty a universal statement about X is true; we say it holds *vacuously*. Hence the single binary relation on the empty set enjoys all of these attributes.

When X has n elements, X^2 has n^2, whence there are 2^{n^2} binary relations on X.

Exercises. 1. For each of the above attributes except transitivity, give an expression in n for the number of binary relations on X having that attribute.

2. For each attribute φ listed above, determine the size of the smallest X such that (i) for every attribute ψ above there exists a relation on X satisfying φ but not ψ; (ii) there exists a relation R on X such that X satisfies φ but no other attribute on the list.

1.2.2 Functions

A *function* $f : X \rightarrow Y$ is a relation which to each x relates exactly one y. The reader may verify that this can be defined equivalently in a more "algebraic" style as $I_X \subseteq f; f\breve{}$ and $f\breve{}; f \subseteq I_Y$, bearing in mind that $f\breve{}$ will not in general be a function.

Composition of functions is defined as for relations, and may be verified to yield a function. The identity relation can be seen to be a function. Hence the class of all sets and their functions form a category differing from the previous example in the maps, some of which have been deleted. This category is called **Set**, and is understood without further qualification to have functions for its maps.

Composition $f; g$ is also written $g \circ f$ or just gf; we shall use whichever best fits the situation. This "backwards" notation is almost always used when maps are functional in nature, where it fits in well with the equivalent definition for functions of composition as $(g \circ f)(x) = g(f(x))$.

A *partial* function is a relation which to each x relates at most one y. Again one may verify that composition of partial functions is a partial function and that the identity is a partial function, yielding the category **Pfn** of sets and partial functions.

An *injection* is a function such that for any two distinct elements $x \neq y$ in its domain, $f(x) \neq y$. The *range* or *image* $f(X)$ of a function $f : X \rightarrow Y$ is the set $\{f(x) \mid x \in X\}$. A *surjection* is one whose image is the whole of its codomain.

Chapter 2

Order

2.1 Ordered Sets

A *preorder* or *ordered set* is a structure $(X, \leq)$ consisting of an underlying set or carrier X, and a reflexive transitive binary relation $\leq$ on X. A *partial order* or *poset* is an antisymmetric preorder. An *equivalence relation* is a symmetric preorder. We denote the respective classes $\mathbf{Ord}, \mathbf{Pos}, \mathbf{Eqv}$.

A familiar example of a preorder is given by the set of points in the line together with the binary relation *left-of.* This relation is certainly not symmetric. However it is antisymmetric, and hence a partial order, since two distinct points cannot both be left of each other.

Now consider the real plane $\mathbf{R}^2$, ordered by *left-of*, with vertically aligned points considered left of each other. This relation is still reflexive, transitive, and not symmetric, but now it is not antisymmetric since vertical lines form cliques in this ordering of the plane. We thus have an example of a preordered set that is not a poset.

The relation of being vertically aligned is an equivalence relation on the real plane.

The relation of being *strictly* to the left is an irreflexive transitive relation.

The identity relation 1_X or I_X on X and the clique K_X on X are both preorders. (also I_X or $=_X$) Other names for 1_X are equality, $=$, and the *discrete* order, while K_X is variously called the *complete*, *codiscrete*, or *chaotic* relation. The empty relation 0_X on X is a preorder just when X is empty, since although it is always transitive it is not reflexive when X is nonempty.

Duality The only properties of relations arising in our treatment of ordered sets are reflexivity, transitivity, symmetry, antisymmetry, and completeness (being a clique). All of these properties are invariant under taking the converse of a binary relation. It follows that theorems about binary relations that can be reduced to first order statements quantifying over binary relations and involving just these properties are invariant under replacing any relational variable R by the expression $R^{\smile}$. This is the *duality principle* for ordered sets.

Hasse Diagrams. The reflexive transitive *closure* of a binary relation R on X, denoted R^*, is the least preordering of X *containing* R.

The reflexive transitive *hull* or interior of a preorder P on X is the least binary relation on X whose transitive closure contains P.

We may express these two conditions with the equivalences

$$\forall R \in \mathbf{Rel} \forall P \in \mathbf{Ord}[R^* \subseteq P \;\equiv\; R \subseteq \mathcal{R}(P)]$$

$$\forall R \in \mathbf{Rel} \forall P \in \mathbf{Ord}[P^\circ \subseteq R \;\equiv\; P \subseteq R^*]$$

. (These are both adjunctions, which we will treat formally when we come to categories.)

Finite partial orders are conveniently depicted as *Hasse diagrams.* This is a two-dimensional representation of a directed acyclic graph all of whose edges are drawn without arrowheads but which are assumed to be directed upwards.

More examples. There are four reflexive binary relations on a doubleton, corresponding to the four subsets of the two pairs $(0,1)$ and $(1,0)$, since reflexivity requires the other two pairs. All four of these reflexive relations are transitive, giving four ways to preorder a doubleton.

In order to find a reflexive relation that is not a preorder we require 3 elements. The binary relation R on $\{0,1,2\}$ satisfying only $0R0R1R1R2R2$ (6 pairs) is reflexive but not transitive.

Each of the set of integers, the set of rationals, and the set of reals forms a chain under their usual ordering.

Any set of subsets of a set, ordered by set inclusion, forms a partial order. For if $X \subseteq Y \subseteq X$ then for every element x, $x \in X$ iff $x \in Y$, whence $X = Y$.

The natural numbers ordered by the relation $x|y$, defined as $y = zx$ for some natural number z, is a partial order with least element 1 and greatest element 0.

Exercise. For any set X, the partial orders on X are in 1-1 correspondence with the irreflexive transitive binary relations on X.

2.1.1 Cliques

A clique *of* a preorder P is a suborder of that preorder that is a clique. A *maximal* clique P is one that is not a proper subset of any clique of P.

Exercise. Formalize and prove the statement that a preorder partially orders its maximal cliques.

The *skeleton* of a preorder P is the partial order whose elements are the maximal cliques of P. (This term is borrowed from category theory.)

We shall call an *endoskeleton* of a preorder P a suborder of P consisting of one element from each maximal clique of P.

Exercise. Every endoskeleton of P is isomorphic to the skeleton of P.

Cliques carry no more information than their underlying sets, and every preorder can be represented as a partially ordered set of cliques. Up to isomorphism therefore, a preorder can be represented as a partially ordered set labeled with numbers, each label giving the cardinality of its associated clique. To be more precise the numbers should be cardinals, since they may be infinite.

2.2 Chains

Elements x and y are *comparable* in a preorder when either $x \leq y$ or $y \leq x$, and otherwise are *incomparable.*

The binary relation of comparability may be seen to be reflexive and symmetric but not in general transitive.

A *chain* or *linear order* or *total order* is a partial order in which all pairs of elements are comparable.

Exercise. A partial order is a chain just when its complement is transitive.

Exercise. Any two finite chains of the same cardinality are isomorphic. (In contrast it can be shown that there are uncountably many nonisomorphic linear orderings of a countably infinite set. Of these, only four are dense (between every two elements lies another), distinguished by whether they have a least and/or a greatest element.)

2.3 Suborders

$(X, \leq)$ is a *suborder* of $(Y, \leq')$ when $X \subseteq Y$ and $\leq$ is the restriction of $\leq'$ to X. That is, for all $x, x' \in X$, $x \leq x'$ if and only if $x \leq' x'$.

The set $(\mathbf{Z}, \leq)$ of integers with their usual order is a suborder of the set $(\mathbf{R}, \leq)$ of reals with their usual order. Any set of subsets of X ordered by inclusion is a suborder of the power set of X ordered by inclusion.

The properties of reflexivity, irreflexivity, transitivity, symmetry, and antisymmetry are all *preserved by* suborders. That is, if a preorder is reflexive so are its suborders, and so on. Hence any suborder of a preorder is a preorder, any suborder of a partial order is a partial order, and similarly for equivalence relations. On the other hand the property of being a clique is not so preserved.

2.4 Direct Products

The direct product $(X, \leq)$ of a pair $((Y_1, \leq_1), (Y_2, \leq_2))$ consists of the set X of all pairs (y_1, y_2) such that $y_1 \in Y_1$ and $y_2 \in Y_2$, and ordered by $\leq$ such that $(y_1, y_1') \leq (y_2, y_2')$ just when $y_1 \leq_1 y_1'$ and $y_2 \leq_2 y_2'$.

For example the direct product $(\mathbf{R}, \leq)^2$ of the ordered set of reals with itself is the real plane ordered so that any two points p and q satisfy $p \leq q$ just when p is below and to the left of q. The direct product of the unit interval $[0, 1]$ with the set of all reals is a strip in the plane of unit width or height (depending on which axis we associate each of the two sets with), ordered as for $(\mathbf{R}, \leq)^2$.

For an arbitrary index set I, $(X, \leq)$ is the *direct product* of the family $(Y_i, \leq_i)_{i \in I}$ when X is the cartesian product of $(Y_i)_{i \in I}$ and $x \leq x'$ just when $x_i \leq_i x_i'$ for all $i \in I$.

An equivalent and insightful way of describing this direct product is in terms of the cartesian product $\prod_i \leq_i$ of the binary relations themselves. This product is a set of I-tuples of pairs. This is not quite right: what we wanted was a set of pairs of I-tuples. However there is a natural isomorphism between these which takes for example the triple of pairs $((x_1, x_2), (y_1, y_2), (z_1, z_2))$ to the pair of triples $((x_1, y_1, z_1), (x_2, y_2, z_2))$, and similarly for any I-tuple of pairs.

As a binary operation, direct product is not associative, since the pairs $((x, y), z)$ and $(x, (y, z))$ are not equal. However there is an isomorphism f from $(X \times Y) \times Z$ to $X \times (Y \times Z)$ defined as $f((x, y), z) = (x, (y, z))$. Hence $(X \times Y) \times Z$ is isomorphic to $X \times (Y \times Z)$. Thus direct product is associative *up to isomorphism.*

The definition of the direct product of the empty family is problematic. The easy way out is to take it to be undefined. However it would be nice if the result turned out to be the identity for direct product. We have already seen that direct product is associative only up to isomorphism, so we should expect nothing better for identities: the direct product of a preorder P with the, or an, identity should be isomorphic with P, albeit not necessarily equal to it.

If we are willing to name a suitable element such as 0, we may take it to be the unique preorder on $\{0\}$. Or we may regard every preordered singleton as a direct product of the empty family, making direct product no longer a single valued operation. The definition we shall adopt is to postulate a particular singleton $\{\bullet\}$ expressly for the purpose of defining the empty product, and say nothing further about the element $\bullet$; it may well be 0 for all we know.

Given an attribute ψ listed above and a family of structures (X_i, R_i) each satisfying ψ, the direct product of that family also satisfies ψ. Exception: the direct product of the empty family $(I = \emptyset)$, if we take it to be defined at all, is a reflexive singleton, ruling the properties empty and irreflexive.

Exercise. For which listed attributes does it suffice for only one member of the family to have that attribute in order that the direct product of the whole family have it?

2.5 Monotone functions

Given two preorders $(X, \leq_X)$ and $(Y, \leq_Y)$, a function $f : X \to Y$ is called *monotone* when $x \leq_X x'$ implies $f(x) \leq_Y f(x')$.

The squaring function on the natural numbers is monotone with respect to their usual order, as can be inferred from the nonnegative slope of its graph. However the squaring function on the integers is not monotone, since for example we have $-2 \leq -1$ but not $(-2)^2 \leq (-1)^2$. This is clear from the negative slope of x^2 for $x < 0$.

The identity function on any preorder, defined as $f(x) = x$, is monotone. So is any constant function.

Given three sets X, Y, and Z and functions $f : X \to Y$, $g : Y \to Z$, the *composition* $g \circ f : X \to Z$ of g with f satisfies $(g \circ f)(x) = g(f(x))$. Sometimes $g \circ f$ is written as $f; g$.

It can be seen that the composition of monotone functions is a monotone function.

Two functions $f : X \to Y$ and $g : Y \to X$ form an *isomorphism pair* when $g \circ f = 1_X$ and $f \circ g = 1_Y$. When there exists such an isomorphism pair between X and Y we say that X and Y are *isomorphic*. Isomorphism of preorders $(X, \leq)$ and $(Y, \leq)$ formalizes the intuitive idea that the two orders are identical except for the choice of underlying sets.

For example the ordered sets $(\mathbf{Z}, \leq)$ and $(\mathbf{Z}, \geq)$ consisting of the integers ordered in each direction are isomorphic, with the isomorphism given by negation, and more generally by any function $f_n : \mathbf{Z} \to \mathbf{Z}$ defined as $f_n(x) = n - x$.

2.6 Bounds

Given a subset Y of a preorder $(X, \leq)$, we say that an element x of X is *covered by* Y when there exists $y \in Y$ such that $x \leq y$, and a *lower bound* of Y when for all $y \in Y$, $x \leq y$. We write $\downarrow Y$ for the set of elements covered by Y, and $\mathcal{L}Y$ for the set of lower bounds of Y.

By the duality principle we may replace $\leq$ by its converse $\geq$ to obtain dual constructs with the corresponding properties, namely the set $\uparrow Y$ of elements *supported by* Y, and the set $\mathcal{U}Y$ of *upper bounds* of Y.

An *order ideal* or *lower set* of a preorder $(X, \leq)$ is a subset Z of X such that if $z \in Z$ and $x \in X$ then $x \leq z$ implies $x \in Z$. Equivalently, Z is an order ideal just when $\downarrow Z = Z$. Both $\downarrow Y$ and $\mathcal{L}Y$ can be seen to be order ideals, whence $\downarrow\downarrow Y = \downarrow Y$ (so $\downarrow$ is idempotent) and $\downarrow(\mathcal{L}Y) = \mathcal{L}Y$. The dual notion is *order filter* or *upper set*.

Proposition 1 *A subset Z of a preorder $(X, \leq)$ is an order ideal if and only if it is the inverse image $f^{-1}(0)$ of the least element 0 of a partial order $(Y, \leq)$ under a monotone function $f : X \to Y$.*

Proof: (If) Given $z \in Z$ and $x \in X$, if $x \leq z$ then $f(x) \leq f(z)$. Hence $f(x) = f(z)$, so $x \in Z$. (Only if) Let the elements of Y consist of the set Z and the cliques of X not appearing in Z, augmented so as to make Z the least element of Y. The function $f : X \to Y$, defined so that $f(x)$ is Z when $x \in Z$ and otherwise is the clique containing x, can be seen to be monotone, with Z the inverse image of the identity. ∎

By reflexivity, $Y \subseteq \downarrow Y$; we then say that $\downarrow$ is *increasing*. In fact $\downarrow Y$ is the least order ideal containing Y, and is called it the order ideal *generated by* Y.

However not only do we not have $Y \subseteq \mathcal{L}Y$, they have only a clique in common. For if y and y' both belong to both Y and $\mathcal{L}Y$ then $y \leq y'$ and $y' \leq y$. We call the elements of this clique the *least* or *minimum* elements of Y. In a partial order Y can have at most one least element, there being no nontrivial cliques.

The dual notion to order ideal is *order filter*, or *upper set*. We write $\uparrow Y$ for the order filter generated by Y, and $\mathcal{U}Y$ for the set of upper bounds of Y.

Let $Y \subseteq Z$ be two subsets of $(X, \leq)$. Then $\downarrow Y \subseteq \downarrow Z$, whereas $\mathcal{L}Z \subseteq \mathcal{L}Y$. That is, $\downarrow$ is monotone with respect to inclusion between subsets of a preorder, whereas $\mathcal{L}$ is antimonotone.

A *closure operation* on a preorder $(W, \leq)$ is a function $f : W \to W$ that is monotone, idempotent, and increasing. Here $W = 2^X$ with $\leq$ as $\subseteq$. Hence $\downarrow: 2^X \to 2^X$ is a closure operation.

Certainly $\mathcal{L}$ is not a closure operation on $(2^X, \subseteq)$; in fact it fails all three conditions. On the other hand we do have $Y \subseteq \mathcal{U}\mathcal{L}Y$ and $Y \subseteq \mathcal{L}\mathcal{U}Y$ (why?). That is, $\mathcal{U}\mathcal{L}$ and $\mathcal{L}\mathcal{U}$ are increasing.

Given two preorders P and Q, a *Galois connection* between P and Q is a pair of antimonotone functions $f : P \to Q$, $g : Q \to P$, the *polarities*, such that gf and fg are increasing. Setting $P = Q = (2^X, \subseteq)$ (a poset), $f = \mathcal{L}$, and $g = \mathcal{U}$ (or vice versa) in this definition shows that $\mathcal{U}$ and $\mathcal{L}$ are the polarities of a Galois connection.

By antimonotonicity, $Y \subseteq \mathcal{U}\mathcal{L}Y$ implies $\mathcal{L}\mathcal{U}\mathcal{L}Y \subseteq \mathcal{L}Y$, while substituting $\mathcal{L}Y$ for Y in $Y \subseteq \mathcal{U}\mathcal{L}Y$ yields $\mathcal{L}Y \subseteq \mathcal{L}\mathcal{U}\mathcal{L}Y$. Hence $\mathcal{L}\mathcal{U}\mathcal{L}Y = \mathcal{L}Y$, that is, $\mathcal{L}\mathcal{U}\mathcal{L} = \mathcal{L}$. By the duality principle, $\mathcal{U}\mathcal{L}\mathcal{U} = \mathcal{U}$. Hence $\mathcal{U}\mathcal{L}\mathcal{U}\mathcal{L} = \mathcal{U}\mathcal{L}$ and $\mathcal{L}\mathcal{U}\mathcal{L}\mathcal{U} = \mathcal{L}\mathcal{U}$, that is, $\mathcal{U}\mathcal{L}$ and $\mathcal{L}\mathcal{U}$ are idempotent.

The composition of two antimonotone functions is monotone, whence $\mathcal{U}\mathcal{L}$ and $\mathcal{L}\mathcal{U}$ are monotone.

Hence $\mathcal{U}\mathcal{L}$ and $\mathcal{L}\mathcal{U}$ are closure operations. This is true of any Galois connection.

Exercise. Show that $\downarrow X \subseteq \mathcal{L}\mathcal{U}X$ but not always conversely, even in a partial order.

Certain subsets Y are of particular interest. When Y is empty, so is $\downarrow Y$, whereas $\mathcal{L}Y = X$. When Y is a clique, $\downarrow Y = \mathcal{L}Y$; since $\downarrow$ is monotone and $\mathcal{L}$ antimonotone, we may think of them as "crossing over" at cliques. When $Y = X$, $\downarrow X = X$, whereas $\mathcal{L}X$ is the set of least elements of X, at most a singleton when X is partially ordered.

Exercise. (i) $\mathcal{L}Y \cap Y = Y$ (i.e. $Y \subseteq \mathcal{L}Y$) just when Y is a clique. (ii) Hence $\mathcal{L}Y \cap Y = \mathcal{U}Y \cap Y$ just when Y is a clique. (iii) However $\mathcal{L}Y \cap \mathcal{U}Y = Y$ just when Y is a maximal clique. (In a partial order these equations hold only for Y the empty set and singletons, there being no other cliques.)

Exercise. Show that $\downarrow$ is injective if and only if the preorder in which it is defined is a partial order.

Theorem 2 *(Dedekind) Every partial order $P = (X, \leq)$ is representable as (that is, isomorphic to) some set M of subsets of some set Y, ordered by inclusion.*

Proof. Take $Y = X$ and represent each $x \in X$ as $\downarrow \{x\}$. Now the function $\downarrow: X \to 2^X$, the restriction of the function $\downarrow$ to the singletons of 2^X, is monotone, and the preceding exercise showed that it is injective. Take M to be the range of $\downarrow$. Then the function $\downarrow: X \to M$ is bijective, and so has an inverse $\downarrow^-: M \to X$. For two arbitrary elements $\downarrow x$ and $\downarrow y$ of M, if $\downarrow x \subseteq \downarrow y$ then since x is in $\downarrow x$ it must also be in $\downarrow y$, that is, $x \leq y$, making $\downarrow^-$ monotone. Hence $\downarrow$ and $\downarrow^-$ constitute an isomorphism pair, making P isomorphic to M. ∎

An element x of a preorder $(X, \leq)$ is called *minimal* when for every $y \in X$, other than x itself, it is not the case that $y \leq x$; equivalently, when $\downarrow x = \{x\}$. It is called *minimum*, or *least*, or (the categorical term) *initial*, when for every $y \in X$, $x \leq y$; equivalently, when $\uparrow x = X$.

A partial order may contain many minimal elements—consider the discrete order on any set. However it may contain at most one least element. For if there were two least elements x, x' we would have $x \leq x' \leq x$, whence $x = x'$.

A preorder however may contain more than one least element. In the plane preordered by left-of, the least elements of the unit square are all those along its left edge. By the argument of the previous paragraph however the least elements of a preorder form a clique. Furthermore all elements of a clique containing a least element must be least, whence if the set of least elements is nonempty it forms a maximal clique. (In a nonempty preorder the empty set is always a nonmaximal clique.)

The least element of an ordered set is a minimal element of that set. In a chain the converse holds: a minimal element must be least and hence unique. In partial orders in general however the converse need not hold;

indeed a partial order may contain minimal elements without containing any least element—consider the discrete order on a two-element set. Or it may contain no minimal elements at all, as with the integers, or the reals.

The notions of least and minimal element apply to any preorder, including suborders of a preorder. Thus if Y is a suborder of X then a minimal element of Y means an element of Y with no other element of Y below it. The least elements of Y if any mean those lower bounds of Y that are in Y, that is, $Y \cap \mathcal{L}Y$. In a partial order this set is either empty or a singleton.

When $Y = \mathcal{U}Z$, a least element of Y is an upper bound of Z that is least among all such upper bounds; these constitute the set $\mathcal{U}Z \cap \mathcal{L}\mathcal{U}Z$. We call such an element a *supremum* or *sup* of Z, and write the set of them either $\sup Z$ or $\bigvee Z$. In a partial order $\bigvee Z$ is either a singleton or empty, in which case it is customary to take $\bigvee Z$ to be the element of the singleton if present, calling it undefined or nonexistent if not. The dual notion is the *infimum* or *inf* of Z, written $\inf Z$ or $\bigwedge Z$.

For example in $(2^X, \subseteq)$, if Y is any set of subsets of X then $\bigvee Y = \bigcup Y$ and $\bigwedge Y = \bigcap Y$, both of which are always defined. However $\bigcup Y$ need not belong to Y.

When talking about $\bigvee Y$ it is important to specify which partial order Y is a subset of. If Y is a subset of two different partial orders, $\bigvee Y$ may be different in each. For example if Y is the set of all rationals less than π then Y is a subset of both the reals and the rationals, with the same order in each case. However $\bigvee Y$ is π in the reals but does not exist in the rationals. In the power set of $\{0, 1, 2\}$, the sup of $\{0\}$ and $\{1\}$ is $\{0, 1\}$, but in the suborder of this obtained by omitting $\{0, 1\}$, $\{0\}$ and $\{1\}$ still have a sup but it is now $\{0, 1, 2\}$.

If in a partial order $Z \cap \mathcal{U}Z$ is nonempty then its unique element will be $\bigvee Z$. However the case $\bigvee Z \notin Z$ is possible: consider the sup of the negative reals, which exists, namely 0, but is not a negative real.

A *principal* order ideal is one generated by a singleton, i.e. of the form $\downarrow \{x\}$. The set of negative integers is a principal order ideal of the integers, being of the form $\downarrow -1$, but the set of negative reals is not a principal order ideal of the reals, though the set of nonpositive reals is. Similar observations hold dually for order filters.

Exercise. For each of the following subsets of the ordered set $(2^X, \subseteq)$ indicate whether it is an order ideal and whether it is an order filter. In each case if it is such then indicate whether it is principal. If the answer depends on properties of X, say how. (i) All subsets. (ii) The nonempty subsets. (iii) The proper subsets. (iv) Those subsets containing a given element x. (v) Those subsets not containing a given element x.

Chapter 3

Monoids

3.1 Semigroups, Monoids, Groups

A *semigroup* is a pair $(X, \cdot)$ where X is a set and $\cdot$ is a binary operation satisfying $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ (associativity). The operation is generically called *multiplication* and its result the *product* of its arguments.

A *monoid* is a triple $(X, \cdot, 1)$ where $(X, \cdot)$ is a semigroup and 1 is an element of X satisfying $1 \cdot x = x = x \cdot 1$. We call 1 the *identity* of the monoid.

A *group* G is a monoid every element x of which has an *inverse* x^{-1} in G, defined as $x \cdot x^{-1} = 1$.

A *commutative* monoid satisfies $x \cdot y = y \cdot x$; commutative groups are called *Abelian*.

Any singleton $\{x\}$ with a binary operation and a constant trivially forms a one-element monoid, that constant automatically being the identity of that monoid. This monoid is trivially commutative. If it is expanded with a unary operation it automatically becomes a group.

The triple $(\mathbf{N}, +, 0)$ of natural numbers with the addition operation $+$ and the additive identity 0 forms a commutative monoid. The triple $(\mathbf{Z}, +, 0)$ of integers forms a commutative monoid. Likewise $(\mathbf{N}, \times, 1)$ and $(Z, \times, 1)$ are monoids. Of these four monoids, only $(\mathbf{Z}, +, 0)$ has an expansion to a group, obtained by adding the negation operator to yield $(\mathbf{Z}, +, -, 0)$. This is the only possible expansion of $(\mathbf{Z}, +, 0)$ to a group: the inverse operator is uniquely determined.

$(\mathbf{Z}_n, +, 0)$ is the commutative monoid of integers modulo n ($n \geq 1$). $(\mathbf{Z}_n, +, -, 0)$ is the Abelian group of integers modulo n.

Exercise. $(\mathbf{Z}_n, \times, 1)$ for $n \geq 2$ is a commutative monoid but not a group.

The set X^* of all finite strings over a set X (the *alphabet* of X^*) forms a monoid under concatenation of strings with identity element the empty string. It is commutative just when X has at most one element, in which case it is (isomorphic to) $(\mathbf{N}, +, 0)$.

The set F of all functions $f : A \to A$ forms a monoid under function composition, with the identity function providing the identity.

For $n \geq 1$, the set of all $n \times n$ integer-valued matrices forms a monoid under matrix multiplication, with the $n \times n$ identity matrix I_{nn} as the monoid identity.

The set of all functions on the reals, under addition of functions defined by $(f + g)(x) = f(x) + g(x)$, with the constantly zero function as identity, forms a commutative monoid.

Monoids are not the exclusive domain of the mathematician. Here is a basic example from electrical engineering.

Suppose we have inexhaustible supplies of each of several types of electronic black boxes each having a male connector at one end and a matching female connector at the other end, with all male connectors being identical and likewise for the female. Associated with each type of box is its *behavior*, defined in some way. We may plug these boxes together (in a straight line - no loops allowed) to form assemblies which also have certain behaviors. This determines an operation xy on behaviors, such that xy is the behavior of the assembly obtained by plugging the output of an assembly with behavior x into the input of one with behavior y.

The order in which the two connections are made when building a three-box assembly does not matter — to build xyz you can start with either xy or yz. This shows that the operation is associative and hence that the set $\mathcal{M}$ of all behaviors obtainable in this way form a semigroup. Treating the absence of boxes as an assembly, $\mathcal{M}$ is a monoid.

Exercises. 1. Suppose the order of boxes is immaterial, two assemblies with the same behavior always connect to yield the identity behavior, and there are only three types of boxes. (i) How many distinct behaviors can be constructed? (ii) Reformulate this electrical engineering problem as a problem in pure mathematics.

2. A monoid has exactly one identity.

3.2 Submonoids

Given a monoid $\mathcal{M} = (X, \cdot, 1)$, we say that a subset Y of X is *closed* when it contains 1, and for all y, y' in Y, $y \cdot y'$ is also in Y. Let $\cdot_Y : Y^2 \to Y$ be the binary operation on Y satisfying $y \cdot_Y y' = y \cdot y'$ for all y, y' in Y; call $\cdot_Y$ the *restriction* of $\cdot$ to Y. Then $(Y, \cdot_Y, 1)$ must be a monoid. For if not it must violate one of the three monoid equations, but then $\mathcal{M}$ would violate that equation with the same values for the variables of that equation.

We call such a $(Y, \cdot_Y, 1)$ a *submonoid* of $(X, \cdot, 1)$.

For example the monoid $(\mathbf{N}, +, 0)$ of natural numbers under addition is a submonoid of the monoid $(\mathbf{Z}, +, 0)$ of integers under addition.

The monoid of all functions $f : A \to A$, denoted A^A, yields many examples of useful submonoids: its injections, its surjections, its bijections (constituting the largest submonoid of A^A that is a group), its continuous functions (defined for an appropriate topology on A), its monotone functions (when A is partially ordered), and so on. In turn A^A is itself a submonoid, namely of the monoid of all binary relations on A under composition of binary relations.

The monoid $(\mathbf{Z}_n, \times, 1)$ of integers modulo n under multiplication has a submonoid forming an Abelian group, obtained simply by omitting 0.

Related notions are subsemigroup and subgroup. A subsemigroup of a semigroup is a subset of the semigroup closed under the operation of the semigroup. A subgroup of a group is a nonempty subset of the group closed under both the unary and binary operations of the group (and hence containing the identity).

The union of two submonoids of $\mathcal{M}$ need not be a monoid. For example consider the submonoid of $(\mathbf{N}, +, 0)$ consisting of all even numbers, and that consisting of all multiples of 3. Their union contains 2 and 3 but not 5 and hence is not a submonoid of $(\mathbf{N}, +, 0)$.

Intersection however is better behaved.

Lemma 3 *The intersection $\bigcap S$ of any set S of submonoids of $\mathcal{M}$ is a submonoid of $\mathcal{M}$.*

Proof: For any k-ary operation f_j and elements $x_1, \ldots, x_k$ in the intersection, $f_j(x_1, \ldots, x_k)$ must be in every submonoid in S, whence it must be in the intersection. ∎

It follows that the submonoids of an algebra form a complete lattice, with $\bigwedge$ provided by intersection. However $\bigvee$ is not union. Let us now see just what $\bigvee$ is.

A subset Y of a monoid $\mathcal{M}$ is said to *generate* $\mathcal{M}$ when $\mathcal{M}$ has no proper submonoid which includes Y. For example $(\mathbf{N}, +, 0)$ is generated by $\{1\}$. Furthermore any set of generators of $(\mathbf{N}, +, 0)$ must contain 1, since omitting 1 yields a proper submonoid.

Proposition 4 *Let $\mathcal{M} = (X, \cdot, 1)$ be a monoid, and let Y be a subset of X. Then $\mathcal{M}' = (Z, \cdot', 1)$ is a submonoid of $\mathcal{M}$ generated by Y if and only if $\mathcal{M}'$ is the intersection of all submonoids of $\mathcal{M}$ containing Y.*

Proof: (If) Y is a subset of $\mathcal{M}'$ because the intersection of a set of sets each including Y must itself include Y. $\mathcal{M}'$ is an intersection of submonoids and hence a submonoid. So no proper submonoid of $\mathcal{M}'$ can include Y, whence $\mathcal{M}'$ is generated by Y.

(Only if) If $\mathcal{M}'$ is a submonoid of $\mathcal{M}$ generated by Y, and $\mathcal{M}''$ is a submonoid of $\mathcal{M}$ including Y, then the intersection of $\mathcal{M}'$ and $\mathcal{M}''$ is a submonoid of them both and including Y. But $\mathcal{M}'$ has no such proper submonoids, whence the intersection is M', making $\mathcal{M}'$ a submonoid of $\mathcal{M}''$. Hence $\mathcal{M}'$ is a submonoid of every submonoid of $\mathcal{M}$ including Y, and hence must be the intersection of the set of such. ∎

Another way to say this is that Y generates the least submonoid of $\mathcal{M}$ that includes Y. It can be seen that there is exactly one submonoid of $\mathcal{M}$ generated by any given subset Y of $\mathcal{M}$.

Instead of forming the submonoid generated by Y as an intersection, we may form it as a union as follows. Given a monoid $\mathcal{M} = (X, \cdot, 1)$, let $F_M : 2^X \to 2^X$ be defined by $F_M(Y) = Y \cup \{1\} \cup \{y \cdot y' \mid y, y' \in Y\}$. Thus $F_M(Y)$ consists of the elements of Y together with 1 and those elements that can be formed from elements of Y via one application of $\cdot$. Define $F_M^i(Y) = F(F(\ldots F(Y)\ldots))$ (F_M repeated i times), and define $F_M^*(Y) = \bigcup_{i \geq 1} F^i(Y)$.

Thus $F_M^*(Y)$ consists of those elements of the monoid representable as a finite product of elements of Y, including Y itself and the empty product 1.

Proposition 5 *$F_M^*(Y)$ is the submonoid of $\mathcal{M}$ generated by Y.*

Proof: Any two elements y, y' in $F_M^*(Y)$ must be in $F_M^i(Y)$ for some i, whence $y \cdot y'$ is in $F_M^{i+1}(Y)$ and hence in $F_M^*(Y)$. Moreover 1 is in $F(Y)$ and hence in $F_M^*(Y)$. Hence $F_M^*(Y)$ is closed. Moreover $F_M^*(Y)$ includes Y since $F_M(Y)$ does. Thus $F_M^*(Y)$ is a monoid including Y. Now for any submonoid $\mathcal{M}' \subseteq \mathcal{M}$ that includes Y we have $F_M(Y) \subseteq F_M(M') = M'$, and by induction $F_M^i(Y) \subseteq \mathcal{M}'$ for all $i \geq 1$. Hence $F_M^*(Y) \subseteq \mathcal{M}'$. Thus $F_M^*(Y)$ is the least monoid including Y. ∎

3.3 Direct Products

The *direct product* $(X_1, \cdot_1, 1_1) \times (X_2, \cdot_2, 1_2)$ of two monoids is $(X_1 \times X_2, \cdot, (1_1, 1_2))$ where $\cdot : (X_1 \times X_2)^2 \to X_1 \times X_2$ is a binary operation defined as $(x_1, x_2) \cdot (y_1, y_2) = (x_1 \cdot_1 y_1, x_2 \cdot_2 y_2)$.

It can be seen that $(1_1, 1_2)$ is the identity for $\cdot$. Furthermore $\cdot$ is associative. For if it were not then associativity would fail in in either coordinate 1 or 2, whence associativity would fail for the corresponding monoid, a contradiction. Hence the direct product of two monoids is itself a monoid.

For example the direct product of the monoid $(\mathbf{N}, +, 0)$ with itself yields the lattice points (points with integer coordinates) of the upper right quadrant of the plane, under the operation of vector addition $((x, y) + (x', y') = (x + x', y + y'))$, with identity the origin. With $\mathbf{R}$ in place of $\mathbf{N}$ we obtain the whole quadrant.

Direct product generalizes to an arbitrary family straightforwardly. The *direct product* of a family $(X_i, \cdot_i, 1_i)_{i \in I}$ is $(\prod_{i \in I} X_i, \cdot, \mathbf{1})$ where $(x \cdot y)_i = x_i \cdot_i y_i$ and $\mathbf{1}_i = 1_i$.

That is, the direct product of a family has for its underlying set the cartesian product of the family of underlying sets, and is equipped with a binary operation $\cdot$ which acts as $\cdot_i$ in the i-th coordinate and a constant $\mathbf{1}$ which acts as 1_i in the i-th coordinate.

The direct product of semigroups and groups is defined similarly. The direct product of a family of semigroups is as for monoids but with the identity $(1_i)_{i\in I}$ omitted. The direct product of a family of groups is as for monoids but with an inverse operation defined as $(x_i)_{i\in I}^{-1} = (x_i^{-1})_{i\in I}$.

3.4 Monoid Homomorphisms

A *monoid homomorphism* from $(X, \cdot_X, 1_X)$ to $(Y, \cdot_Y, 1_Y)$ is a function $f : X \to Y$ such that (i) $f(x \cdot_X x') = f(x) \cdot_Y f(x')$ for all $x, x' \in X$, and (ii) $f(1_X) = 1_Y$. The second condition is omitted for semigroup homomorphisms. A group homomorphism is just a monoid homomorphism, in which case it automatically satisfies the condition $h(x^{-1}) = h(x)^{-1}$, since $h(x)h(x^{-1}) = h(xx^{-1}) = h(1) = 1$ whence $h(x^{-1}) = h(x)^{-1}$.

Monoid homomorphisms, or just homomorphisms in this chapter, are traditionally referred to as *linear* functions.

The function $h : (N, +, 0) \to (Z_n, +, 0)$ defined as $h(m) = m \bmod n$ is a homomorphism, as is $h : (Z, +, 0) \to (Z_n, +, 0)$ defined similarly.

Consider $(\mathbf{R}^{\mathbf{R}}, +, 0)$, the monoid under addition of all functions on the reals. The derivative operator on the submonoid consisting of the differentiable such functions is a homomorphism, since $d(f + g)/dx = df/dx + dg/dx$ and $d0/dx = 0$.

The identity function on a monoid $\mathcal{M}$ is trivially a monoid homomorphism.

The restriction of the identity function to a submonoid $\mathcal{M}'$ of $\mathcal{M}$ is an injective homomorphism from $\mathcal{M}'$ into $\mathcal{M}$, called the *inclusion* of $\mathcal{M}'$ into $\mathcal{M}$.

The function $\pi_1 : \mathcal{M} \times \mathcal{N} \to \mathcal{M}$ defined as $\pi_1((x, y)) = x$ is a homomorphism, called the *first projection* of $\mathcal{M} \times \mathcal{N}$ onto $\mathcal{M}$. The second projection $\pi_2 : \mathcal{M} \times \mathcal{N} \to \mathcal{N}$ is defined similarly as $\pi_2((x, y)) = y$. More generally the i-th projection $\pi_i : \prod_{i\in I} \mathcal{M}_i \to \mathcal{M}_i$ is defined as $\pi_i(x) = x_i$.

Whereas inclusions are injective, projections are surjective.

An *isomorphism* of monoids is a bijective homomorphism. When there exists an isomorphism between two monoids we say that they are *isomorphic*. An isomorphism of a monoid onto itself is called an *automorphism*.

Exercises. 1. An Abelian group G may be defined to be a group whose inverse operation $^{-1} : G \to G$ is a homomorphism.

2. Up to isomorphism, there are just two two-element monoids.

3.5 Congruences and Quotients

A *semigroup congruence* is an equivalence relation $\cong$ on a monoid such that if $x \cong x'$ and $y \cong y'$ then $x \cdot y \cong x' \cdot y'$.

For example the identity relation ($x \cong y$ implies $x = y$) and the complete relation ($x \cong y$ always) are semigroup congruences on any semigroup. The relation "$x = 0 = y$ or $x \neq 0 \neq y$" (an equivalence relation with two classes, $\{0\}$ and the rest) is a congruence on $(\mathbf{N}, +)$ but not on $(\mathbf{Z}, +)$ since $-1 \cong 1$ but $1+1 \ncong -1+1$. The relation "same parity" (both even or both odd) is a congruence on both $(\mathbf{N}, +)$ and $(\mathbf{Z}, +)$.

An equivalence relation on a set X partitions X into blocks called *equivalence classes.* When the equivalence relation is a congruence the blocks are called *congruence classes.* We write $[x]_\cong$ for the congruence class containing a specified element x.

The congruence classes of a semigroup $\S = (X, \cdot)$ forms a semigroup under the operation $\cdot'$ defined by $[x] \cdot' [y] = [x \cdot y]$, called the *quotient* of $\S$ *by* $\cong$, written $\S/\cong$. The definition of a congruence ensures that $\cdot'$ is well-defined in that it does not depend on the choice of representative x of the class $[x]$: any other y in the same class would yield the same result.

A quotient of a monoid is a monoid, with identity $[1]$. A quotient of a group is a group, with identity $[1]$ and an inverse $[x]^{-1} = [x^{-1}]$ for each class $[x]$.

The *kernel* of a homomorphism $h : \mathcal{M} \to \mathcal{N}$, written $\ker h$, is the binary relation $\cong$ on X_M defined by $x \cong y$ just when $h(x) = h(y)$.

Proposition 6 *A binary relation on X_M is a congruence on $\mathcal{M}$ just when it is the kernel of a homomorphism $f : \mathcal{M} \to \mathcal{N}$ for some $\mathcal{N}$.*

Proof: It can be seen that the kernel of a homomorphism is a congruence. Conversely any congruence determines a quotient, whose kernel is that congruence. ∎

We have defined the quotient $\S/\cong$ to be a semigroup of equivalence classes. The function $h : \S \to \S/\cong$ defined by $h(x) = [x]$ can be seen to be a homomorphism, which we also refer to as a quotient when there is no ambiguity. Note that such a quotient is always surjective.

A congruence $x \cong y$ being the set of pairs (x, y) for which $x \cong y$, the intersection $\cong_1 \cap \cong_2$ of two congruences $\cong_1$ and $\cong_2$ on a semigroup $\S$ then has the property that $x(\cong_1 \cap \cong_2)y$ holds just when $x \cong_1 y$ and $x \cong_2 y$.

Proposition 7 *The intersection of two congruences is itself a congruence.*

Proof: If $x(\cong_1 \cap \cong_2)y$, then $x \cong_1 y$ and $x' \cong_1 y'$, whence $x \cdot x' \cong_1 y \cdot y'$ and similarly $x \cdot x' \cong_2 y \cdot y'$. Hence $x \cdot x'(\cong_1 \cap \cong_2)y \cdot y'$. ∎

This generalizes immediately to the intersection of any family $(\cong_i)_{i \in I}$ of congruences on a semigroup $\S$.

Exercises. 1. For which congruences on $(\mathbf{N}, +)$ is $0 \cong 1$? For which is $0 \cong 2$? (Remember $1 \cong 1$.) What is the least congruence containing $2 \cong 5$? What are the corresponding answers for $(\mathbf{Z}, +)$? (Remember $-1 \cong -1$).

2. For any natural number n, the binary relation "x is congruent to y mod n," defined as $\exists ab[x+an = y+bn]$, is a congruence on any submonoid of $(\mathbf{Z}, +, 0)$, in particular on $(\mathbf{N}, +, 0)$.

3. In X^* each of the following three relations between strings x and y is a congruence: of equal length; agree in their first n symbols; contain the same number of occurrences of each symbol.

4. Given homomorphisms $h : \mathcal{M} \to \mathcal{N}$, $h' : \mathcal{M} \to \mathcal{N}'$, define their *product* $h \times h' : M \to \mathcal{N} \times \mathcal{N}'$ as $(h \times h')(x) = (h(x), h'(x))$. Show $\ker(h \times h') = \ker(h) \cap \ker(h')$. Generalize to an arbitrary family $(h_i)_{i \in I}$ of homomorphisms.

3.5.1 Semigroup Ideals

We have already encountered ideals in the context of ordered sets. Essentially the same notion arises for semigroups and monoids, though the following elementary definition does not make this connection apparent.

An *ideal* of a semigroup $\S = (X, \cdot)$ is a nonempty subset $Z \subseteq X$ such that for all $z \in Z$ and $x \in X$, $x + z \in Z$ if and only if $x \in Z$, and $z + x \in Z$ if and only if $x \in Z$.

The "if" direction of the definition ensures that every ideal of $\S$ is a subsemigroup of $\S$. The "only if" direction however rules out some subsemigroups. For example the subsemigroup of $(\mathbf{N}, +)$ omitting just 1 is not an ideal because $2 \in \mathbf{Z}$ and $2 + 1 \in \mathbf{Z}$ but $1 \notin \mathbf{Z}$.

The semigroup X^* of strings on X has a least ideal, namely the singleton $\{\epsilon\}$, and a greatest ideal, namely X^* itself. In between there is for each n the ideal I_n consisting of all strings of length a multiple of n. There is also for each subset $Y \subseteq X$ the ideal Y^* of all strings on Y.

Theorem 8 *If Z and Z' are ideals of $\S$ so is $Z \cap Z'$.*

Proof: $x \in \bigcap_i Z_i$ iff $\forall i[x \in Z_i]$ iff $\forall i[(x+z \in Z$ and $x \in Z \cap Z'$ iff $[x \in Z$ and $x \in Z']$ iff $[x+z \in Z$ and $x+z \in Z']$ iff $x+z \in Z \cap Z'$. ∎

This generalizes immediately to the intersection $\bigcap_i Z_i$ of a family of ideals $(Z_i)_{i \in I}$ of a semigroup §.

The ideal *generated by* a subset X of a semigroup § is the least ideal of § containing X. The above theorem ensures that the least such ideal exists, namely the intersection of those ideals containing X.

Exercises. 1. A subset Z of a monoid $\mathcal{M}$ is a submonoid of $\mathcal{M}$ if it is an ideal of $\mathcal{M}$, but not necessarily conversely.

2. A subset Z of a group G is a subgroup of G if and only if it is an ideal of G.

The following observation relates semigroup ideals to congruences and to ideals in other structures besides semigroups.

Proposition 9 *Z is an ideal of $\S = (X, \cdot)$ if and only if it is the inverse image of the identity 1 of a monoid $\mathcal{M}$ under a semigroup homomorphism $h : \S \to \mathcal{M}$.*

Proof: (If) Let $Z = h^{-1}(1)$. Let $z \in Z$ and $x \in X$. Then $h(x \cdot z) = h(x) \cdot h(z) = h(x)$, so $x \in Z$ iff $x \cdot z \in Z$; and likewise for $z \cdot x$.

(Only if) Let Z be an ideal of §. Define $x \cong y$ iff there exist $z, z' \in Z$ such that $x \cdot z = y \cdot z'$, making $\cong$ a congruence. Now for any $x \in X$ and $z \in Z$, $[x]+[z] = [x+z] = [x]$, and $[z]+[x] = [z+x] = [x]$, whence the quotient $\S/\cong$ is a monoid with identity $[z] = Z$. Hence Z is the inverse image of $[z]$ under this quotient. ∎

We may now make the connection with order ideals. Recall that an order ideal of an ordered set $(X, \leq)$ is the inverse image of the least element of a partial order $(Y, \leq)$ under a monotone function $f : X \to Y$. The least element plays an analogous role to that of the monoid identity. Note that X need have neither a least element nor be antisymmetric; the notion of order ideal is defined for any ordered set. By the same token, the notion of monoid ideal does not depend on the identity of the monoid, and could have been defined more generally for any semigroup.

A *principal* ideal of a monoid $\mathcal{M}$ is an ideal generated by a single element of $\mathcal{M}$.

Theorem 10 *(Principal Ideal Theorem for Z.) Every ideal I in $(\mathbf{Z}, +, 0)$ is principal.*

Proof: If I contains only one number it must be 0, which then generates I. Otherwise, let $x < y$ be two consecutive numbers in I as close together as any pair in I. From $x \cong y$ and $y-x \cong y-x$ we infer $y \cong 2y-x$, whence $2y-x$ is also in I, and more generally I must contain $x + n(y-x)$ for all $n \in Z$, an arithmetic progression infinite in both directions. There is now no room in I for additional numbers without violating the hypothesis that x and y are as close as any pair in I. That I contains 0 determines the alignment of this progression, which can then be seen to consist of all multiples of $y-x$. Hence $y-x$ generates I. ∎

Chapter 4

Lattices

4.1 Semilattices

An *upper semilattice* is a partial order in which every nonempty finite subset has a least upper bound. The dual of a semilattice is a *lower semilattice*, in which every nonempty finite subset has a greatest lower bound.

For example the integers under their usual order form a semilattice, since any finite nonempty set of integers contains a greatest element.

Exercise 11 *"Nonempty finite" may be replaced by "two-element".*

If "nonempty" is dropped then the semilattice must have a least element, in which case it is called unital.

We may equip a semilattice with a binary operation $x \vee y$ yielding $\sup\{x, y\}$. The dual of $\vee$ (the operation for lower semilattices) is written $x \wedge y$.

We may define $\vee$ and $\wedge$ in terms of $\leq$ via the following equivalences.

$$\begin{aligned} x \vee y \leq z \quad &\text{iff} \quad x \leq z \text{ and } y \leq z \\ x \leq y \text{ and } x \leq z \quad &\text{iff} \quad x \leq y \wedge z \end{aligned}$$

Exercise 12 *Show that $\vee$ and $\wedge$ so defined are commutative, associative, and idempotent.*

Thus the equational theory of a semilattice must include the equations $x \vee y = y \vee x$, $x \vee (y \vee z) = (x \vee y) \vee z$, and $x \vee x = x$, and dually for a lower semilattice. This makes (lower) semilattices a special kind of commutative semigroup that we shall call *idempotent*, meaning satisfying $x \vee x = x$ (or $x \wedge x = x$). Unital semilattices form a commutative idempotent monoid, with identity the least element.

Conversely we may obtain $\leq$ from $\vee$ by defining $x \leq y$ as $x \vee y = y$.

Exercise 13 *Show that $\leq$ so defined in any commutative idempotent semigroup is a partial order. Infer that semilattices are completely axiomatized by the commutative idempotent semigroup axioms.*

4.2 Lattices

A *lattice* is a semilattice that is also a lower semilattice.

For example every chain is a lattice. The power set $(2^X, \subseteq)$ is a lattice, with union and intersection as join and meet respectively. The natural numbers ordered by $x|y$ is a lattice with lcm and gcd as join and meet.

To axiomatize the theory of lattices requires more than just the axioms of the theory of semilattices together with those of lower semilattices. For one thing there is nothing in the axioms thus far to prevent $\vee$ and $\wedge$ being the same operation!

We add to the axioms for semilattices and lower semilattices the *absorption laws*, $x = x \vee (x \wedge y)$ and its dual $x = x \wedge (x \vee y)$.

Exercise 14 *Show that the absorption laws ensure that $\vee$ and $\wedge$ define the same partial order, and are duals with respect to that order. Infer that the axioms for two commutative idempotent monoids together with the absorption laws completely axiomatize the theory of lattices.*

Exercise 15 *Both absorption laws are required. (Hint: consider MIN and bitwise OR of natural numbers.)*

Thus a lattice is an algebra $(X, \vee, \wedge)$ satisfying equations expressing associativity, commutativity, and idempotence of $\vee$ and $\wedge$, and satisfying the two absorption equations. The class of lattices is thus a finitely axiomatized equational class.

4.3 Distributive Lattices

A lattice of fundamental importance is the two-element chain. It is the only two-element lattice. This lattice features prominently in logic as the lattice of truth values.

The equational theory of the two-element lattice goes beyond the theory of lattices, for it includes the distributive law $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ as well as its dual. This is not a law of lattices in general, where only one direction of this law holds, namely $(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z)$. A lattice satisfying the distributive laws is itself called *distributive*.

Exercise 16 *Derive either distributive law from its dual.*

As usual we ask whether there are yet more laws to be discovered for $(2, \vee, \wedge)$. The answer is that we now have a complete axiomatization: the theory of $(2, \vee, \wedge)$ is just the theory of distributive lattices.

Theorem 17 *(Birkhoff's representation theorem for distributive lattices) Every free distributive lattice X^* is isomorphic to a subalgebra of a direct power of $(2, \vee, \wedge)$.*

We call a subalgebra of a direct power of $(2, \vee, \wedge)$ a *ring of sets*, that is, a set of sets closed under union and intersection.

Proof: The elements of the free distributive lattice on X have a long history as the monotone Boolean formulas with variables drawn from X. A typical such formula is $xy \vee yz \vee wz$, a join of 3 terms, and a term such as xy is a meet of two variables. First construct the algebra $T_\wedge$ of terms as the free lower semilattice generated by X, which we may take to consist of the finite nonempty subsets of X under union. Then form the free semilattice $E'_\vee$ generated by T, which we may take to consist of the finite nonempty subsets of T. Finally define $\wedge : E'^2 \to E'$ as the (necessarily unique) extension of $\wedge : T^2 \to T$ to a bimorphism, possible since $E'_\vee$ is symmetric (why?). With $\wedge$ so defined we have

$$x \wedge y = \{s \cup t | s \in x, t \in y\}.$$

This construction ensures that $E'_{\vee\wedge}$ is Γ $\models$-free where Γ consists of the axioms for semilattices and lower semilattices together with $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (one of the two distributivity axioms). This may be proved in just the same way as our earlier proof that the ring of polynomials on X is free for commutative rings with identity.

To satisfy the absorption laws we define a binary relation $\cong$ on these expressions such that $x \cong y$ just when every set in exactly one of x or y is a subset of some set in $x \cap y$.

Exercise 18 *Show that $\cong$ is a congruence on $E'_{\vee\wedge}$.*

Exercise 19 *Each congruence class under $\cong$ is closed under arbitrary intersection.*

Let $E''_{\vee\wedge}$ be the quotient of $E'_{\vee\wedge}$ by $\cong$, possible by the first exercise; E'' consists of congruence classes of elements of E'. By the second exercise the intersection of the class is a member of the class, providing it with a canonical representative. Therefore let us take $E_{\vee\wedge}$ to consist of these representatives, each of which can be seen to be finite and nonempty and to have members which are finite and nonempty. The correspondence between the classes and the their representatives provides an isomorphism between E'' and E.

Exercise 20 *Show that $E_{\vee\wedge}$ is a distributive lattice, and that it is free.*

Now consider the function $\uparrow: E \to 2^{2^X}$ mapping each finite set x of finite subsets of X to the least upper set that includes x.

Exercise 21 $\uparrow: E_{\cup\cap} \to 2^{2^X}_{\vee\wedge}$ *is a lattice homomorphism.*

Exercise 22 *For each x in E, the minimal elements of $\uparrow x$ are the elements of x.*

It follows that $\uparrow$ is an injective homomorphism, and the rest follows routinely. ∎

Although this theorem seems to make it obvious what the elements of free distributive lattices, also known as monotone Boolean functions, look like, the problem of enumerating them, known as Dedekind's problem, is by no means straightforward. The sizes of the free distributive lattices on 1 to 7 generators are 1, 4, 18, 166, 7579, 7828352, and 2414682040996. For unital distributive lattices add 1 for each of top and bottom; thus if both are present the sequence becomes 3, 6, 20, 168, etc. For comparison the sizes of the corresponding free Boolean algebras, whose definition we will encounter shortly, are 4, 16, 256, 65536, 4294967296, 18446744073709551616, and 340282366920938463463374607431768211456.

4.4 Ideals and Filters

A semilattice ideal of a semilattice L is a nonempty subsemigroup of L that is also an order ideal.

Exercise 23 *A semilattice ideal of a semilattice L is a semigroup ideal of L.*

A semigroup ideal of an upper semilattice is called a *filter.*

Exercise 24 *An ideal is a directed lower set, and a filter a filtered upper set.*

Ideals and filters exist in lattices by virtue of any lattice being simultaneously a semilattice and a lower semilattice.

Example. Define an abstract proposition to be a congruence class of proposition under the congruence "logically equivalent." Abstract propositions are partially ordered by implication and form a lattice L under conjunction and disjunction. A *theory* T is a subset of L which (among other things) is closed under conjunction (if p and q are in T so is $p \wedge q$) and modus ponens (if p and $p \to q$ are in T then so is q). Provided T includes all implications holding in L it is a filter. (Alternatively we could define modus ponens so that if p is in T and $p \to q$ is an implication of L then q is in T.)

The dual to a theory is a set of counterexamples closed under disjunction and under the dual of modus ponens (if p is a counterexample and $q \to p$ is an implication in L, then q is a counterexample). Such a set forms an ideal of L.

Exercise 25 *Noting that every ideal of a lattice is a semilattice congruence class, need it also be a lattice congruence class (the inverse image of 0 under a lattice homomorphism)?*

Example. Any chain is closed under both $\vee$ and $\wedge$. Hence an ideal in a chain is simply a lower set, and a filter an upper set. In particular the set of ideals in the chain of rationals is what we earlier referred to as $C \cup O$, the closed and open lower sets of rationals, from whence sprang the reals. An ideal in the chain R of reals is also a lower set, and it too may be either open or closed, this time for every ideal, not just those whose sup is a rational.

What do ideals in product lattices look like? Let $L \times M$ be the direct product of lattices $L_{\vee\wedge}$ and $M_{\vee\wedge}$. If I and J are ideals of L and M respectively then it is easy to check that the Cartesian product $I \times J$ is an ideal of $L \times M$. The following gives a very strong converse to this that is analogous to the unique factorization theorem for positive integers.

Theorem 26 *(Unique factorization of ideals) Every ideal K of a product lattice $L \times M$ factors uniquely as $I \times J$ for ideals $I \subseteq L$ and $J \subseteq M$.*

Proof: Let (ℓ, m) be any element of K. Define $I = \{i | (i, m) \in K\}$ and $J = \{j | (\ell, j) \in K\}$. If i is in I and $i' \leq i$ is in L then $(i', m) \leq (i, m)$ is in $L \times M$, but since (i, m) is in the ideal K so is (i', m), so i' is in I. Similar reasoning shows that I is closed under $\vee$, whence I is an ideal, and similarly for J.

We now claim that $K = I \times J$. To show $I \times J \subseteq K$, suppose (i, j) are in $I \times J$. Then (i, m) and (ℓ, j) must have been in K, so $(i, m) \vee (\ell, j) = (i \vee \ell, m \vee j)$ is in K, so (i, j) is in K. To show $K \subseteq I \times J$, suppose (i, j) is in K. Then $(i, j) \vee (\ell, m) = (i \vee \ell, j \vee m)$ is in K, whence $(\ell, j \vee m)$ and $(i \vee \ell, m)$ are in K, so $(i \vee \ell, j \vee m)$ is in $I \times J$, so (i, j) is in $I \times J$.

To see that this factorization is unique, suppose there were a distinct factorization $K = I' \times J'$. Then there is some element either of L in exactly one of I and I', or of M in exactly one of J and J'. If it were i in I but not I' then K contains (i', m) but $I' \times J'$ does not, so this is not a factorization of K. The other cases are argued similarly. ∎

We infer from the last two theorems that an ideal in a product of chains is a product of lower sets. In particular an ideal in the plane $R^(2, \vee, \wedge)$ is the half-box $\{(x, y) | x \leq x_0, y \leq y_0\}$ defined by its upper right corner (x_0, y_0), with either of x_0 or y_0 permitted to be $+\infty$ (what do we get with $-\infty$?). Dually a filter in the plane is a half-box defined by its lower left corner. Half-boxes are good examples to keep in mind while following the proof below of Stone's theorem.

The geometric intuition in this example can be strengthened by rotating the plane 45 degrees to make the line $y = x$ vertical. Then the relationship $p \leq q$ between distinct points p, q in $R^(2, \vee, \wedge)$ holds just when the line from p to q points up and has a slope of at least 45 degrees. We can then think of $p \leq q$ as one definition of "p is below q". With this orientation of R^2 filters are filter-shaped and ideals are mountain-shaped, and as the song goes your ideal should be to climb them all.

A *prime* ideal is an ideal whose complement is a filter. Since the complement of a lower set is always an upper set, an ideal is prime just when its complement is nonempty and closed under conjunction. Equivalently, if $x \wedge y$ belongs to a prime ideal then so does at least one of x or y. The dual is a prime filter, a filter not the whole lattice such that if $x \vee y$ belongs to it then so does one of x or y.

Example. A complete theory (one containing every sentence or its negation, i.e. a know-it-all theory) forms a prime filter: if $x \vee y$ is in the theory then one of x or y must be. This need not be the case for an incomplete theory, e.g. what biologists know. Just because they that one of x or y has to be true does not mean that they have to know which one is true. This is particularly obvious when y is the negation of x. A filter that is the whole lattice corresponds to an inconsistent theory; prime filters are consistent in this sense.

A *bipartition* of a set is a partition of that set having two blocks. A *bicongruence* is a congruence having exactly two equivalence classes.

Theorem 27 *The following conditions on a bipartition of a lattice L are equivalent.*

(i) The bipartition is a lattice bicongruence.

(ii) The bipartition is the kernel of an onto lattice homomorphism $h : L \to 2$.

(iii) One block of the bipartition is a prime ideal. (A prime filter would serve the same purpose.)

(iv) The blocks of the bipartition are respectively an ideal and a filter.

Proof: (i)→(ii). The bicongruence is the kernel of some lattice homomorphism, whose image must have two elements and must be a lattice. But there is only one two-element lattice, namely $(2, \vee, \wedge)$.

(ii)→(iii). If x and y are in the block mapped by h to 0 then for any $z \le x$, $h(z) \le h(x) = 0$ so $h(z) = 0$, and $h(x \vee y) = h(x) \vee h(y) = 0$, making $h^{-1}(0)$ an ideal. If $x \wedge y$ is in this ideal then $0 = h(x \wedge y) = h(x) \wedge h(y)$ so one of $h(x)$ or $h(y)$ must be 0, i.e. one of x or y must be in $h^{-1}(0)$, so $h^{-1}(0)$ is prime.

(iii)→(iv). The complement of an ideal must be upward closed. If x and y both belong to the complement then so must $x \wedge y$, since the ideal is prime.

(iv)→(i). Suppose $x \equiv x'$ and $y \equiv y'$. If x or y is in the ideal then so is $x \wedge y$; also so is one of x' or y', and thus so is $x' \wedge y'$. If both x and y are in the filter then so is $x \wedge y$; also so are both of x' and y', and thus so is $x' \wedge y'$. Hence $x \wedge y \equiv x' \wedge y'$. Dually $x \vee y \equiv x' \vee y'$. ∎

The next theorem, while not particularly difficult, is nevertheless among the more influential theorems of this century, in that it did more to stimulate interest in lattice theory, Boolean algebra, and universal algebra than any other single theorem.

Theorem 28 *(Stone, 1935) Two disjoint subsets of a distributive lattice, respectively an ideal and a filter, are separable by a lattice bicongruence.*

Proof: Let I and F be the given ideal and filter respectively on the lattice L. Consider the ideals in the interval $[I, L - F]$ of 2^L, that is, those ideals J for which $I \subseteq J \subseteq L - F$. These ideals form a partial order under inclusion, from which we select any maximal chain. (That partial orders have maximal chains is Kuratowski's Lemma, one of the more self-evident forms of the Axiom of Choice.) Take I^+ to be the union of the chain, itself an ideal (why?) disjoint from F. It remains to show that I^+ is prime, which we will do in a moment.

Exercise 29 *(A geometric instance of this construction.) Show that the lattice $R^(2, \vee, \wedge)$ is distributive. Taking L to be this lattice in our proof, show that there are at most two choices for I^+. When is there only one choice? How many choices can there be for I^+ if L is $R^3_{\vee\wedge}$?*

Exercise 30 *Show that the least ideal containing I^+ and an element x is $I^+ \cup \downarrow (x \vee I^+)$. Infer that if x is not in I^+ then there exists an element u of I^+ such that $x \vee u$ is in F.*

We now complete the proof by showing that the complement of I^+ is closed under conjunction. Let x and y be any two elements of the complement. Let u and v be the elements given by the preceding exercise for x and y respectively, both in I^+. Let $w = u \vee v$, also in I^+ since I^+ is closed under $\vee$. Since $x \vee u$ and $y \vee v$ belong to F, so must $x \vee w$ and $y \vee w$, F being upper. But then $(x \vee w) \wedge (y \vee w)$ must also be in F, F being closed under $\wedge$. But by distributivity this element of F is $(x \wedge y) \vee w$. Since w is in I^+ and I^+ is closed under disjunction, $x \wedge y$ must be in the complement of I^+. ∎

A set of partitions of a set X *separates* X when its intersection is the identity relation. That is, every pair of elements is separated by some partition.

Theorem 31 *Every distributive lattice is separated by its bicongruences.*

Proof: Without loss of generality we may assume $x \not\leq y$. Apply the above theorem to the filter $\uparrow x$ and ideal $\downarrow y$, which can be seen to be disjoint. ∎

Theorem 32 *Every distributive lattice L is in ISP$((2, \vee, \wedge))$. Equivalently, every lattice is isomorphic to a so-called ring of sets, a set of sets closed under union and disjunction, that is, a sublattice of a direct power of $(2, \vee, \wedge)$.*

Proof: Let M be any set of bicongruences separating L. Then the product of all lattice homomorphisms $h : L \to (2, \vee, \wedge)$ corresponding to some bicongruence in M must be an injective lattice homomorphism $h^M : L \to 2^M_{\vee\wedge}$, whence L is isomorphic to a subalgebra of $2^M_{\vee\wedge}$. ∎

(What we have done in effect is to make enough cuts, in the form of bicongruences, across L to carve it up into its singletons. Each cut can then be regarded as corresponding to some object that is present as an element of every element of L on the filter side of the cut and absent from elements on the ideal side.)

Exercise 33 *Any set of partitions separating X has a subset separating X of cardinality at most $\binom{|X|}{2}$. (Hence M need not be particularly large.)*

Also note that the h constructed by this proof is in effect a function mapping each x to a set of proper prime filters all containing x. In particular if M consists of all bicongruences on L then $h(x)$ consists of all prime filters of L containing x.

Earlier we saw, via a normal form argument, that every *free* distributive lattice is in ISP$((2, \vee, \wedge))$, that is, every free distributive lattice is isomorphic to a ring of sets. An equivalent statement is that every distributive lattice is in HSP$((2, \vee, \wedge)$, since a variety consists of the homomorphic images of its free algebras. The corollary just proved is a strengthening of this; we can either drop "free" in the first statement or replace H by I in the second.

Corollary 34 *Every distributive lattice can be completed, that is, extended to a complete lattice.*

Proof: Extend the ring of subsets of M representing L to the power set of M. L may be verified to be a sublattice of this complete lattice. More refined completions are possible, but we will settle for this brute force one. ∎

Chapter 5

Birkhoff-Stone Duality and Chu Spaces

5.1 Birkhoff-Stone Duality

In previous chapters we have represented various objects—sets, posets, semilattices, distributive lattices, Boolean algebras—as sets of subsets of some set. Birkhoff-Stone duality refines this by breaking the representation into two stages and studying the structures obtained at each stage. At the start of the first stage we have the object to be represented, and the end of the second we have its representation, an isomorphic object. In between we obtain the Birkhoff-Stone dual of the object.

Birkhoff duality is the finite case of Stone duality. The infinite case is traditionally a difficult business, which our use of Chu spaces will greatly simplify. Much of the general picture is conveyed by the finite case, with which we therefore begin.

Although we have not previously considered so trivial a matter as representing sets, it is a good place to begin the study of Stone duality. The Stone dual of a finite set X is its power set 2^X, a Boolean algebra.

Now up to isomorphism there is a 1-1 correspondence between sets and Boolean algebras. Each finite set X determines the finite Boolean algebra 2^X, while any finite Boolean algebra, which is necessarily atomic, determines the set of its atoms. The correspondence is established by observing that the atoms of 2^X form a set isomorphic to (i.e. equinumerous with) X itself, while the power set of the atoms of any finite Boolean algebra B is a Boolean algebra isomorphic to B.

This correspond extends on the set side to finite posets, and on the Boolean algebra side to finite bounded distributive lattices. The Stone dual of a finite poset is the set of its order filters, which form a finite bounded distributive lattice.

As a special case we start with finite chains. The Stone dual of a finite chain C is another chain which we shall denote $C\perp$ (though 2^C is just as reasonable). C has a bottom if and only if $C\perp$ does not have a top, and dually; in particular the Stone dual of an unbounded chain has both bounds, while that of a chain with one bound (e.g. top) has the same bound (top in this example).

To extend the set-Boolean algebra correspondence to chains-chains and thence to posets-distributive lattices, we need a replacement for the notion of atom. This is apparent since a chain, regardless of its length, can have at most one atom. The appropriate notion is that of a *prime* element, namely one that is not the join of the set of elements strictly below it. The bottom element is not prime because the set of elements strictly below it is the empty set, whose join is bottom. Any other element x of a finite chain however is prime, since the join of the elements strictly below x is the predecessor of x in that chain.

Theorem.

5.2 Chu Spaces

Definition 35 A *Chu space* $A = (P, S, v)$ *over* a set K consists of a set P of *points*, a set S of *states*, and a function $v : P \times S \to K$ assigning a *value* $v(p, s)$ to each point p in each state s. ∎

We associate with $v : P \times S \to K$ the functions $v_- : P \to (S \to K)$ and $v^- : S \to (P \to K)$ satisfying $v_-(p)(s) = v^-(s)(p) = v(p, s)$. We abbreviate $v_-(p)$ to v_p, called the *extension* of p, and $v^-(s)$ to v^s, the *extension* of s. We think of v^s as one of the permitted *paintings* of the underlying set P, with K for our palette. Dually v_p is a painting of S, understood as the varying values of one point encountered as one traverses the "possibility space" of alternative paintings of P. We write $V^* = \{v^s \mid s \in S\}$ for the set of extensions of states, through which $S \xrightarrow{v^-} K^P$ factors as $S \to V^* \to K^P$. Dually $P \xrightarrow{v_-} K^S$ factors through the set $V_* = \{v_p \mid p \in P\}$ of extensions of points as $P \to V_* \to K^S$. When all states have distinct extensions, i.e. v^- is injective ($S \cong V^*$), we call v *extensional* and say it has *enough points* (to distinguish states). (This situation is very important, allowing us to interpret (P, S) for $S \subseteq K^P$ as a Chu space.) When all points have distinct extensions ($P \cong V_*$) we call A T_0 (by analogy with the topological property of that name), and say it has *enough states*. Locales are the prototypical example of a nonextensional but T_0 space.

Logically speaking, points are *necessary* in the sense that they are necessarily all present in the space at the one time. Dually, states are *possible*, in that the space is in one state *or* another, like the possible worlds of a Kripke structure (our conventional understanding of spaces does not permit us to imagine that the whole space is in all states simultaneously).

More states mean more degrees of freedom, corresponding to less structure. At one extreme the extensional space (P, K^P) contains all possible states and hence has the vacuous structure of a set or discrete space, which we shall identify with the set P itself. We view the omitted states, those in $K^P - V^*$, as the *atomic* properties of the space, collectively constituting the theory or *structure* of the space. At the other extreme the space $(P, \emptyset)$ omits all states, which we view as the inconsistent structure on P. A one-state space, S a singleton, is "rigid," every point having a uniquely determined or constant value. The canonical rigid space is $(K, \{0\})$, which we denote $\perp$ (not K, which a moment ago we associated, in its capacity as a set, with the discrete space (K, K^K)).

The dual of $A = (P, S, v)$ is the *state space* $A \perp = (S, P, v^{\smile})$ where $v^{\smile}(s, p) = v(p, s)$. Duality interchanges points and states, and hence necessity and possibility. This confers on duality one of the qualities of logical negation, another being double negation, $A^{\perp\perp} = A$.

Theorem 36 *Let $X \subseteq B$ generate[1] the CABA B, and let $L \subseteq B$ be any subcomplete lattice[2] of B such that $X \subseteq L$. Then B is the unique extension of L to a CABA generated by X.*

That is, given just the Hasse diagram of such a lattice L, with $X \subseteq L$ indicated, we can deduce not only the full n-cube from whence it came but where L must embed in it, i.e. what bit-strings to label both the real (X) and imaginary ($L - X$) states of L with.

Proof: The idea is to identify certain elements of L, which we then put in 1-1 correspondence with the atoms of B. We take $\mathcal{E}$ to be the set of sup-irreducible elements of L, definable as those elements not expressible as the sup (the arbitrary, i.e. possibly infinite or empty, join) of the set of elements strictly below them. We embed L in the CABA $2^{\mathcal{E}}$ via $f : L \to 2^{\mathcal{E}}$ defined by $f(y) = \mathcal{E} \cap \downarrow y$ where $\downarrow y = \{x \in L \mid x \leq y\}$. This function can be seen to preserve all joins and meets of L and to be injective.

In order to show that $\mathcal{E}$ is the smallest set such that L embeds in $2^{\mathcal{E}}$, suppose L embeds in $2^{\mathcal{E}'}$ by the embedding $g : L \to 2^{\mathcal{E}'}$. Then for each sup-irreducible $x \in L$, let $a_x \in g(x) - \bigcup_{y<x} g(y)$. Such an a_x exists as x is sup-irreducible. Now for any two sup-irreducibles, if $a_x = a_y$ then $a_x \in (x \wedge y) < x$, which would contradict the definition of a_x. Thus there are as many elements in $\mathcal{E}'$ as in $\mathcal{E}$. So $2^{\mathcal{E}}$ is the smallest CABA

[1] That is, B has no proper sub-CABA containing X

[2] That is, a "sub-(complete lattice)" of B, meaning a subset of B closed under arbitrary (including empty and infinite) meets and joins of B.

containing L. However X generates B, and $X \subseteq L$, so B is also the smallest. Thus $2^{\mathcal{E}}$ is isomorphic to B. We can now construct an isomorphism between $2^{\mathcal{E}}$ and B such that L is embedded similarly in both, by mapping each $x \in \mathcal{E}$ to the a_x constructed earlier. ∎

For example the Chu space $a \to b$ can be redrawn as a chain (a distributive lattice) with a reduction from $2^2 = 4$ elements to $2 + 1 = 3$ elements, more generally from 2^n to $n + 1$. The theorem applies only to T_0 gates. It is evident that X generates $2^{\mathcal{E}}$ iff $(\mathcal{E}, X)$ is T_0.

5.3 Chu Transforms

Definition 37 A *Chu transform* $(f, g) : (P, S, v) \to (Q, T, w)$ consists of functions $f : P \to Q$, $g : T \to S$ satisfying $v(p, gt) = w(fp, t)$ for all $p \in P$ and $t \in T$. *Composition* of $(f', g') : (Q, T, w) \to (R, U, X)$ with $(f, g) : (P, S, v) \to (Q, T, w)$ is defined by $(f', g')(f, g) = (f'f, gg')$, satisfying $v(p, gg'u) = w(fp, g'u) = x(f'fp, u)$ and hence a Chu transform. Associativity is inherited from that of function composition. The *identity* transform on (P, S, v) is the pair $(1_P, 1_S)$ of identity functions on P, S respectively. We abbreviate $(f, g) : (P, S, v) \to (Q, T, w)$ to $f : A \to B$ when unambiguous. ∎

Taking the duals $A \perp = (S, P, v\breve{})$ and $B \perp = (T, Q, w\breve{})$ of the domain and codomain of $(f, g) : (P, S, v) \to (Q, T, w)$ necessarily entails replacing (f, g) by (g, f). To construe this as a Chu transform we must then treat it as $(g, f) : (T, Q, w\breve{}) \to (S, P, v\breve{})$. That is, duality reverses the direction of transforms, much as transposing a matrix reverses the direction of the linear transformation it defines.

5.4 Universality of Chu Spaces

We show here that every n-ary relational structure is concretely representable as an object of $\mathbf{Chu}(2^n)$, and that the Chu transforms between the representing Chu spaces are (as functions) exactly the homomorphisms between the represented structures. Here "concretely" means that the representation has the same underlying set as the represented structure.

Here by "A represented as B" we shall mean throughout that the category C_A of all A's *fully* embeds[3] in the category C_B of all B's.

Definition 38 For any ordinal n, an n-ary *relational structure* (X, ρ) consists of a set X, the *carrier*, and an n-ary relation $\rho \subseteq X^n$ on X. A *homomorphism* $f : (X, \rho) \to (Y, \sigma)$ between two such structures is a function $f : X \to Y$ between their underlying sets for which $f\rho \subseteq \sigma$. Here $f\rho$ denotes $\{f\mathbf{a} \mid \mathbf{a} \in \rho\}$, where $\mathbf{a}$ denotes $(a_0, \ldots, a_{n-1})$ and $f\mathbf{a}$ denotes $(fa_0, \ldots, fa_{n-1})$. We denote by $\mathbf{Str}_n$ the category formed by the n-ary relational structures and their homomorphisms. ∎

It suffices to treat structures with a single carrier and relation, since k carriers can be combined as their disjoint union, kept track of with k unary relations ($\lceil \log_2 k \rceil$ is enough information-theoretically, but not enough to ensure that homomorphisms respect type). Multiple nonempty relations on a set can be joined to form a single relation on the same set, of arity at most the sum of the arities of its constituent relations. For algebras, structures all of whose $(n+1)$-ary relations are n-ary operations, the join may share the input coordinates of the operations, reducing the total arity to the maximum of the input arities plus the number of operations (including constants).

This notion of homomorphism is standard in the strong sense that *any* class of n-ary relational structures and their homomorphisms constitutes a full subcategory of $\mathbf{Str}_n$. Familiar examples of such categories and their arities include those of semigroups (3), monoids (4), groups (3), rings (4), rings with a multiplicative

[3] An embedding is a faithful functor $F : C_A \to C_B$, i.e. for distinct morphisms $f \neq g$ of C_A, $F(f) \neq F(g)$, and is *full* when for all pairs a, b of objects of C_A and all morphisms $g : F(a) \to F(b)$ of C_B, there exists $f : a \to b$ in C_A such that $g = F(f)$.

unit (5), fields (4), lattices (3), lattices with top and bottom (5), Boolean algebras (3), vector spaces (4),[4] directed graphs or binary relations (2), multigraphs (4), posets (2), and categories (4).

Many of these numbers benefit from group structure, for which homomorphisms preserve inverses and identities even when these operations are not given explicitly as part of the relation. Units of monoids, including tops and bottoms of lattices, are not so fortunate and each requires its own unary relation in order to be recognized and preserved by homomorphisms.

The universality achieved here is of a different kind from that achieved by say ZF set theory. Externally a model of ZF is a single object of $\mathbf{Str}_2$ of some cardinality, with membership as its only relation, "internally" coding objects larger than any fixed cardinal including its own. Our universality has no separate notion of an internal world; instead we code our objects purely externally.

We now define the promised functor $F : \mathbf{Str}_n \to \mathbf{Chu}(\mathbf{2}^n)$, namely in definitions 5.4 and 5.4, and prove that it is full, faithful, *and concrete.*

The complementarity of constraints and states indicates ρ and $\overline{\rho}$ as the appropriate respective sources of each. We shall define a state to be essentially a subset of $\overline{\rho}$, with however a small but essential refinement. The following lemma obtains from the standard constraint-based definition of homomorphism an equivalent state-based characterization.

Lemma 39 $f\rho \subseteq \sigma \;\Leftrightarrow\; f^{-1}\overline{\sigma} \subseteq \overline{\rho}$. Here $\overline{\rho} = A^n - \rho$ and $\overline{\sigma} = B^n - \sigma$.

Proof:

$$\begin{array}{rcll} f\rho \subseteq \sigma & \Leftrightarrow & \rho \subseteq f^{-1}\sigma & (\text{Definition of } f^{-1}) \\ & \Leftrightarrow & \overline{f^{-1}\sigma} \subseteq \overline{\rho} & (\text{Complement}) \\ & \Leftrightarrow & f^{-1}\overline{\sigma} \subseteq \overline{\rho} & (f^{-1} \text{ preserves Boolean operations}) \end{array}$$

∎

Definition 40 (F on objects). Let $\mathbf{2}^n$ denote the set of n-bit bit vectors, that is, n-tuples over $\mathbf{2}$. We define the object part of the functor $F : \mathbf{Str}_n \to \mathbf{Chu}(\mathbf{2}^n)$ as taking the n-ary relational structure (A, ρ) to the Chu space (A, R, v) defined as follows. Take R to consist of those n-tuples $r \in (2^A)^n$ of subsets of A for which $\prod_i r_i \subseteq \overline{\rho}$. Let $v : A \times R \to 2^n$ satisfy $v(a,r)_i = 1$ if $a \in r_i$, and 0 otherwise. ∎

It might seem that R could be represented more naturally and conveniently as just the power set of $\overline{\rho}$. But observe that a state r as defined here can be recovered from the set $\prod_i r_i$ of its n-tuples just when no component r_i is empty. The definition of $f^{-1} : S \to R$ in Definition 5.4 below requires each r_i to be available independently even when some are empty.

The crucial test of whether (A, R, v) faithfully represents (A, ρ) is whether ρ can be recovered from (A, R). We show this constructively as follows.

Lemma 41 *For all* $\mathbf{a} \in A^n$, $\;\mathbf{a} \in \rho \;\Leftrightarrow\; \forall r{\in}R\; \exists i{<}n : v(a_i, r)_i = 0$.

Proof:

$$\begin{array}{rcll} \mathbf{a} \in \rho & \Leftrightarrow & \forall r{\in}R : \mathbf{a} \notin \prod_i r_i & (\text{Construction of } R) \\ & \Leftrightarrow & \forall r{\in}R\; \exists i{<}n : a_i \notin r_i & (\text{Definition of product}) \\ & \Leftrightarrow & \forall r{\in}R\; \exists i{<}n : v(a_i, r)_i = 0 & (\text{Construction of } v) \end{array}$$

∎

Corollary 42 *F is injective on objects.*

[4] Treat as partial rings, with uv defined just when u is on a specified axis. This works equally well for homogeneous vector spaces (all over the one field) and heterogeneous, the only nontrivial field endomorphisms being automorphisms.

Lemma 43 *(A, R, v) is extensional.*

Proof: If $v^r = v^{r'}$ then $\forall i[a \in r_i \Leftrightarrow a \in r'_i]$, so $\forall i : r_i = r'_i$, whence $r = r'$. ∎

If we regard R as a subset not of $(2^A)^n$ but of the isomorphic $(2^n)^A$, this makes Lemma 43 clear by qualifying (A, R) as an extensional object of $\mathbf{Chu}(2^n)$. We may view (A, R) for arbitrary $R \subseteq (2^n)^A$ as a generalization of $(A, \overline{\rho})$, which in the case $n = 1$ reduces to ordinary binary relations, which as previously noted capture topological spaces along with other similar structures such as complete lattices etc.

Definition 44 (F on maps). Let $f : (A, \rho) \to (B, \sigma)$ be a homomorphism, with $F(A, \rho) = (A, R, v)$ and $F(B, \sigma) = (B, S, w)$ as per Definition 5.4. Define $f^{-1} : (2^n)^B \to (2^n)^A$ to take $g : B \to 2^n$ to $gf : A \to 2^n$. Now for all $s \in S$, $\prod_i s_i \subseteq \overline{\sigma}$ by construction of S. Hence $\prod_i f^{-1}s_i \subseteq \overline{\rho}$, by Lemma 39. Thus $f^{-1}s \in R$ by construction of R. We may therefore define $F(f)$ as (f, f^{-1}) where $f^{-1} : S \to R$. ∎

Theorem 45 *The functor F of Definitions 5.4 and 5.4 is concrete, faithful, and full.*

Proof: F is concrete by construction, and *a fortiori* faithful.

For fullness consider any Chu transform $(f, g) : F(A, \rho) \to F(B, \sigma)$ where $F(A, \rho) = (A, R)$ and $F(B, \sigma) = (B, S)$. If $\mathbf{a} \in \rho$, then for every $s \in S$ there exists $i < n$ such that

$$\begin{array}{rcll} v(a_i, gs)_i & = & 0 & \text{(Lemma 41 with } r = gs\text{)}, \\ \text{whence } w(fa_i, s)_i & = & 0 & \text{((}f, g\text{) is a Chu transform).} \end{array}$$

Hence by Lemma 41, $f\mathbf{a} \in \sigma$, establishing that f is a homomorphism. And since (A, R, v) is extensional, by Lemma 43, g is determined by f. Hence $F(f) = (f, g)$. ∎

Remarks. (i) Where size matters, R need contain only those states representable as the inverse image of a tuple of singletons. These can be characterized explicitly as those states r with the property that either $r_i = r_j$ or $r_i \cap r_j = \emptyset$ for all $i, j < n$, observing that f^{-1} preserves this property. (ii) Lemma 43 is an inessential bonus. Had Definition 5.4 produced a nonextensional (A, R, v), we would simply have enforced extensionality, needed for fullness, by identifying those states having the same extension.

5.5 Profinite Chains

Chains are a special case of distributive lattices where profiniteness is easily defined and understood while providing an ideal stepping stone to the general notion for arbitrary distributive lattices.

A *gap* in a chain is a pair $x < z$ of elements with no other element between them, i.e. no y such that $x < y < z$. A chain is *profinite* when its gaps are dense, that is, when for every every pair $x < z$ there exists a gap $x' < z'$ satisfying $x \leq x' < z' \leq z$.

Neither the rationals nor the reals standardly ordered form a profinite chain, since they contain no gaps at all. Now the reals are constructed as Dedekind cuts in the rationals, i.e. order ideals (we could equally well have taken order filters, since the complement of an order ideal in a chain is an order filter). But not *all* order ideals, or every rational would occur twice as a real, once as the sup of an ideal and again as the inf of a filter; we prevent this by omitting one of these occurrences. What harm is there in leaving both copies in? Well, they create gaps, since no other ideal can come between these two copies of a given rational.

Moreover these gaps are clearly dense. Now one might think to prove density of gaps from density of the rationals, but this is not necessary.

Theorem 46 *In any chain formed as the ideals of a chain, whether dense like the rationals or discrete like the integers, gaps are dense.*

Proof: Take any two distinct ideals $Y \subset Y'$ in the chain of all ideals of the chain C, and let $x \in Y' - Y$. Then $C - \uparrow x \subset \downarrow x$ is a gap lying between Y and Y'. ∎

Theorem 47 *Every profinite chain is isomorphic to the order ideals of some chain.*

Proof: TBD ∎

5.6 Profinite Distributive Lattices

1. A distributive lattice L is *profinite* when there exists a poset P whose order ideals ordered by inclusion form a poset isomorphic to L.

2. An intrinsic definition of profinite distributive lattice (i.e. one that refers only to properties of objects internal to the lattice, as opposed to postulating new objects as in the preceding definition of "profinite") is as follows.

A distributive lattice is *profinite* when it is complete and completely distributive (infs distribute over sups), and its maximal chains are totally disconnected.

A distributive lattice is *complete* when it has all sups and infs, and *completely distributive* when infs distribute over sups, of whatever cardinality. An element z of a chain is a *successor* when there exists $x < z$ in the chain such that there is no y in the chain for which $x < y < z$. A chain is *profinite* when every element is the sup of successors. (Any dense chain contains no successors and hence is not profinite.)

A proof that definitions 1 and 2 are equivalent amounts to making the first definition intrinsic by taking P to consist of those elements x of L such that x is a successor in some maximal chain of L.

Now the crux of the theorem to be proved is that the notion of free CABA $F(L)$ on (i.e. generated by) a profinite distributive lattice L is well-defined, which is isomorphic to the power set 2^X on the set X of sup-irreducibles or complete primes of L, with L embedded in 2^X (and hence in $F(L)$ by the isomorphism) by sending each element x of L to $\downarrow x \cap X$.

This embedding of L in $F(L)$ may be efficiently understood as the dual of the notion of the underlying set X of a poset $(X, \leq)$, which is isomorphic to the underlying set of the discrete poset on X.

Denote by CABA the category of CABA's and CABA homomorphisms (for which the condition that they be complete lattice homomorphisms suffices), and by **StoneDLat**the category of profinite distributive lattices and profinite distributive lattice homomorphisms (again it suffices that they be complete lattice homomorphisms). We have the following equivalences:

$$CABA\ \mathbf{Set}^{\mathrm{op}} \tag{5.1}$$

$$\mathbf{StoneDLat}\ \mathbf{Pos}^{\mathrm{op}} \tag{5.2}$$

that is, CABA's are the dual of sets and profinite distributive lattices are the dual of posets.

There is an evident adjunction

```
            F
          ---->
    Set          Pos                          (3)
          <----
            U
```

such that $U(P)$ is the underlying set of P, with U having as left adjoint the functor F for which $F(X)$ is the discrete poset on X. Furthermore the unit $\eta_X : X \to UFX$ embedding X in the underlying set of the discrete poset FX is an isomorphism, equivalent in any adjunction to saying that F is full and faithful; these equivalent conditions are lumped together by saying that **Set** is a full coreflective subcategory of **Pos**.

The dual of this adjunction is then

```
                  op
                 U
          op  ------>    op
      Pos              Set                        (4)
              <------
                  op
                 F
```

with its orientation indicating that U^{op} is left adjoint to F^{op}. This is obtained simply by reversing the arrows of **Set** and **Pos**; as such it is meaningful only category theorically.

We make this duality meaningful set theoretically as follows. To each set X (an object of **Set**) we associate the set power set 2^X thought of as the set of characteristic functions $c : X \to 2$, and to each function $f : X \to Y$ (a morphism of **Set**) we associate the function $2^f : 2^Y \to 2^X$ defined as composition on the right by f, i.e. $(2^f)(c) = cf$, i.e. extending each c backwards with f as in

```
     f        c
X------>Y----->2                                  (5)
```

This defines a functor $2^- : \mathbf{Set}^{\mathrm{op}} \to \mathbf{Set}$ (definable as $\mathrm{Hom}(-,2)$ in **Set**, but that's another story) that is evidently faithful and injective on objects, hence an embedding of $\mathbf{Set}^{\mathrm{op}}$ in **Set**. This embedding constitutes a *representation* of $\mathbf{Set}^{\mathrm{op}}$. Note that any set with at least 2 objects may be substituted for 2 in this representation of $\mathbf{Set}^{\mathrm{op}}$, with an equivalent effect: 3^f is as good as 2^f. However 0^f and 1^f will not do.

Regarding **Set** as a full subcategory of **Pos** (posets), we may extend this functor to $2^- : \mathbf{Pos}^{\mathrm{op}} \to \mathbf{Set}$ (definable as $\mathrm{Hom}(-,2)$ in **Pos**) where 2^P is the set of *monotone* functions from P to the 2-element chain, and for each monotone $f : P \to Q$, 2^f is the function (not necessarily monotone) mapping each c in 2^Q (monotone) to cf as before, this time a function of 2^P because composition of monotone functions is monotone. This functor too is faithful and injective on objects, and hence an embedding of $\mathbf{Pos}^{\mathrm{op}}$ in **Set**, i.e. a representation of posets. As for $\mathbf{Set}^{\mathrm{op}}$, any sufficiently large poset will do in place of 2, namely one that contains 2 as a subposet.

We name the 2-based representations of $\mathbf{Set}^{\mathrm{op}}$ and $\mathbf{Pos}^{\mathrm{op}}$ respectively CABA and **StoneDLat**, each forming a subcategory of **Set** via the embedding 2^-.

(4) now renames naturally to

```
                  F
              ------>
   \StoneDLat           CABA                           (6)
              <------
                  U
```

This follows the convention of calling the left adjoint F and the right U; we rely on context to distinguish these from other uses of F and U. Given a CABA B, $U(B)$ is the underlying profinite distributive lattice of B, obtained by simply "forgetting" the complement operation. Since $F : \mathbf{Set} \to \mathbf{Pos}$ is full, faithful, and injective on objects, so is its dual $U : CABA \to \mathbf{StoneDLat}$ (giving a one-line argument that it suffices to take CABA homomorphisms to be **StoneDLat** homomorphisms). Furthermore the counit $\epsilon_B : B \to FUB$ of (6) is an isomorphism, corresponding to the fact that although UB "forgets" that the complements are there it does not actually delete them, and "unwittingly" even preserves them (i.e. the **StoneDLat** morphisms turn out to preserve complement even though they are not required to), so that FUB does not require any new elements in order to restore the complements to the status of being officially recognized.

Thus far all we have done is to list certain adjunctions, and observe some obvious facts about them. However we can now read off the desired theorem as the triviality that the notion of the free CABA FL on a profinite distributive lattice L is well-defined, as is the unit $\eta_L : L \to UFL$ embedding L in FL, which is the image under the contravariant power set functor of the monotone function $\epsilon_P : FUP \to P$, viz. $2^{\epsilon_P} : 2^P \to 2^{FUP}$, where 2^{FUP} is evidently a CABA.

Chapter 6

Equational Logic

6.1 Language

The notion of expression, or term as we shall call it for reasons that will emerge below, plays a central role in logic. One satisfactory model of a term is an ordered rooted vertex-labelled tree. The label on interior vertex v is an operation symbol, intended to denote an operation whose arguments are denoted by v's immediate descendants. Each exterior node (leaf) is labelled with either a constant or a variable. Constants amount to zeroary operation symbols. Variables have one of several roles, depending on context: they may behave indistinguishably from constants (valuation); they may be explicitly bound by some quantifier (first order or other logic); they may be implicitly universally quantified (identities in equational logic); or they may serve as tokens for subtrees (instantiation in a deductive system).

Another model of a term is a string of symbols from an alphabet containing a symbol for each operation, a symbol for each variable, and the symbols "," "(", and ")" (comma, left parenthesis, and right parenthesis). Parentheses and comma are used as in "$F(x, F(y, x))$" where F is an operation symbol and x,y are variables. A variation of this model permits certain binary operation symbols to be written between their two arguments in place of the comma (infix notation).

Whether a term such as $(x + y) + z$ is modelled as a tree with two internal vertices and three leaves, or as a string of 7 symbols, is clearly a matter of convention. The substance of the subject of logical algorithms should be independent of the choice of representation of terms. It is possible and even convenient to formalize this independence with the help of algebra. The necessary concept is that of a term algebra.

6.1.1 Term Algebras

Consider the set of all expressions that can be built from a set X of variables, say $\{\mathtt{x,y}\}$, using say the binary function symbol `f` and the ternary function symbol `g`. Those expressions are `x`, `y`, `f(x,x)`, `f(x,y)`, `f(y,x)`, `f(y,y)`, `g(x,x,x)`, ..., `f(g(f(x,x),y,g(y,x, y)),x)`, ... For definiteness let us consider each expression to be a string over the 7-symbol alphabet `x y f g , ( )`. Any other choice of representation of expression, while yielding a different set, would serve our purpose equally well.

Now define on this set a binary operation f which given expressions x and y yields the expression $\mathtt{f(}x\mathtt{,}y\mathtt{)}$, and a ternary operation g which given expressions x, y, and z yields the expression $\mathtt{g(}x\mathtt{,}y\mathtt{,}z\mathtt{)}$. Thus $f(\mathtt{f(x,g(y,y,x))},\mathtt{y}) = \mathtt{f(f(x,g(y,y,x)),y)}$. To make this less cryptic let us put quotes on the expressions we are talking about, to distinguish them from the expressions we are talking with. The above equation then reads $f($"`f(x,g(y,y,x))`","`y`"$)$ ="`f(f(x,g(y,y,x)),y)`''.

(This potential for confusing the domain of discourse with the discourse itself is characteristic of logic, where one discourses on discourse. Because of this self-referential aspect, definitions may sometimes appear circular.)

We now have in hand a set (of expressions) together with two operations on that set. This constitutes an algebra of similarity type 2,3. This particular algebra is what we call a term algebra of that similarity type.

Note that this term algebra has a specific set. If we changed our conventions for representing expressions, or if we changed the set of variables to some other equinumerous set, we would change the set of expressions, and hence the algebra. However the two algebras would be isomorphic. (We assume that whatever representation is used is unambiguous; ambiguous languages are inadequate for the purpose of constructing term algebras.)

Almost inseparable from term algebras is the notion of valuation. A valuation is a homomorphism $h : T \to A$ where T is a term algebra. What makes term algebras special among all algebras of a similarity type is that for each assignment of values in A to variables of T there exists a unique valuation whose restriction to the variables is that assignment. Its uniqueness follows from the fact that the variables of T together generate T. Its existence follows from the fact that every element of T can be generated by the variables in only one way, so that we do not run into a conflict in having to assign distinct elements of A to the same element of T. Thus to specify a valuation it suffices to specify its action on the variables, which is just how evaluation should behave.

To illustrate this, let T be a term algebra generated by $X = \{x, y\}$ and having one binary operation f, and let A be the natural numbers under addition. Assign 3 to x and 5 to y. This specifies a valuation h, whose value at various values may be calculated as follows:

$$\begin{aligned} h(x) &= 3 \\ h(y) &= 5 \\ h(f(x,y)) &= h(x) + h(y) = 8 \\ h(f(f(x,y),y) &= h(f(x,y)) + h(y) = 8 + 5 = 13 \end{aligned}$$

Thus far we have viewed term algebras as an abstraction of the intuitively obvious notion of expressions. While good for intuition this is scarcely a formal definition. A straightforward and precise definition is to take the existence of valuations as the defining characteristic of a term algebra, rather than as merely a derived characteristic.

Given a similarity type, a class C of algebras of that type, and a set X, a *C-free algebra $F(X)$ freely generated by X* is an algebra of that similarity type such that X is a subset of the underlying set $U(F(X))$ of $F(X)$, and for any algebra B in C and function $h_X : X \to U(B)$, h_X extends to a homomorphism $h : F(X) \to B$ in exactly one way.

We shorten "freely generated by" to "on" where context makes our intent clear.

When a C-free algebra A is itself in C we call A a *free* C-algebra on X. If in addition X is empty we call A an *initial* algebra of C; initial algebras are nonempty just when the similarity type includes constants (zeroary operations). When C is the class of all algebras of a given similarity type we call A a *term* algebra on X. (In this case A must be in C, whence term algebras are automatically free C-algebras.)

The next theorem gives an upper bound on how many "different" free algebras there can be.

Theorem 48 *Free C-algebras with isomorphic (i.e. equal cardinality) sets of free generators are isomorphic.*

Proof: Given two such algebras A and B, select an isomorphism ϕ mapping the generator set of A to that of B. This extends uniquely to a homomorphism $h : A \to B$ since A is C-free and B is in C. By symmetry we may find $g : B \to A$ extending the inverse of ϕ. But then $g \cdot h : A \to A$ is a homomorphism extending the identity map on the generator set of A. There can only be one such homomorphism since A is C-free and A is in C. But the identity homomorphism on A is such, whence $g \cdot h$ must be that identity. By symmetry $h \cdot g$ must be the identity on B. Hence g and h are isomorphisms and A and B are isomorphic. ∎

Of course if free algebras did not even exist this theorem would hold vacuously. For suitable choices of C this may in fact be the case; C empty suffices but is not necessary. In the next section we shall show that when C is closed under subalgebras and direct products, for any choice of generator set X there exists a free C-algebra on X.

6.2 Equational Theories

In this section we settle at the outset on a fixed similarity type. Given any set X of variables, we let T_X denote a term algebra of that similarity type generated by X. As observed earlier, T_X is unique up to isomorphism.

Given a term algebra T, we define an *equation* $s = t$ *in* T to be a pair (s,t) of terms of T.

Algebra A *satisfies* equation (s,t) just when for every homomorphism $h : T_X \to A$, $h(s) = h(t)$, that is, just when (s,t) is in the kernel of every homomorphism from T_X to A. This says that no matter what values we assign to the variables of T_X (defining the homomorphism h), the corresponding values of s and t (namely $h(s)$ and $h(t)$) always turn out the same.

We say that a *class* of algebras satisfies an equation just when every algebra of the class satisfies the equation.

The set of all equations on a term algebra T_X satisfied by some algebra or class is the *equational theory on* X *of* that algebra or class.

For example the equational theory of $(\mathbf{Z}, +, -, \times)$ on $\{x, y, z\}$ includes $x+y = y+x$, $x+(y+z) = (x+y)+z$, $-(x+y) = (-x)+(-y)$, and $x \times (y+z) = x \times y + x \times z$.

Define an A-*congruence on* T_X to be the kernel of some $h : T_X \to A$, and define a C-*congruence* on T_X to be an A-congruence on T_X for some algebra A in class C. Write $\equiv_C^X$ for the intersection of all C-congruences on T_X, itself a congruence on T_X.

Exercise 49 *$\equiv_C^X$ is the equational theory of C on X.*

2. $T_X/\equiv_C^X$ is C-free freely generated by $X/\equiv_C^X$ (where $X/\equiv_C^X$ denotes those congruence classes containing elements of X).

An *ISP-class* is a class closed under isomorphisms, subalgebras, and direct products.

Theorem 50 *When C is an ISP-class, $T_X/\equiv_C^X$ is a free C-algebra.*

Proof: It suffices to show that $T_X/\equiv_C^X$ is in C. Each C-congruence is the kernel of some $h : T_X \to A$ for some A in C. Now $h(T_X)$ is a subalgebra of A and hence also in C since C is closed under subalgebras. Also $h(T_X)$ is isomorphic to $T_X/\ker(h)$, so the latter is also in C since C is closed under isomorphisms. And the direct product of $T_X/ker(h)$ over all such $ker(h)$'s must also be in C since C is closed under direct products. But since $\equiv_C^X$ is the intersection of all such $ker(h)$'s, $T_X/\equiv_C^X$ must be isomorphic to a subalgebra of this direct product, and hence also in C. ∎

A *variety* (also, *equational* class) is an ISP-class closed under homomorphisms.

A *model* of a set of equations is an algebra that satisfies all the equations in the set.

Theorem 51 *(Birkhoff). A class is a variety if and only if it consists of all models of some equational theory.*

Proof: The "if" direction is an easy exercise. Now suppose C is a variety. We must show that C consists of all models of $\equiv_C^X$. Certainly every A in C is such a model, so it suffices to show that every such model A is in C. Let A be a model of $\equiv_C^X$ with underlying set $U(A)$. Now the identity function $1_{U(A)} : U(A) \to U(A)$ extends to a homomorphism $h_A : T(U(A)) \to A$. Since $1_{U(A)}$ is onto so is h_A. But $\equiv_C^X \subseteq \ker(h)$ whence h_A factors as $h_A : T(U(A)) \to T(U(A))/\equiv_C^X \to A$ (earlier exercise) showing that A is the homomorphic image of $T(U(A))/\equiv_C^X$, h_A being onto. But the latter is in C because C is an ISP-class. But C is closed under homomorphisms, whence A is in C. ∎

6.3 Substitution

There is one further useful property of equational theories: they are substitutive. In order to define this term we need the notion of a substitution.

A *substitution* is an endomorphism on a term algebra.

Informally, a substitution replaces the variables of each term by terms. All occurrences of a variable are replaced by the same term throughout. A variable may be replaced by itself, so a substitution need only change a single variable, or even none at all (the identity substitution).

A binary relation R on a term algebra T_X is *substitutive* when for any substitution h on T_X, if (s,t) is in R then so is $(h(s), h(t))$.

Lemma 52 *Equational theories are substitutive.*

Proof: Let C be a class having equational theory $\equiv_C^X$ on T_X and let $h : T_X \to T_X$ be an endomorphism of T_X. If for some pair (s,t) of terms $(h(s), h(t))$ is not in $\equiv_C^X$, there must exist A in C and a homomorphism $g : T_X \to A$ distinguishing $h(s)$ and $h(t)$. But then the homomorphism $gh : T_X \to A$ distinguishes s and t, whence (s,t) is not in $\equiv_C^X$ either. ∎

The *substitutive closure* of R, written $S(R)$, is the set $\{h(s), h(t) \mid (s,t) \in R \text{ and } h \text{ is a substitution}\}$. $S(R)$ can be seen to be the least substitutive relation containing R.

Exercise 53 *The intersection of any set of substitutive relations on a term algebra is a substitutive relation on that algebra.*

Exercise 54 *The congruence closure of a substitutive relation is substitutive.*

6.4 Equational Logic

In equational logic we begin with certain equations called axioms and from them infer further equations called theorems. We are interested in the appropriate rules of inference for equational logic.

The *semantic consequences* of a set Γ of equations, written $\Gamma \models$, is the equational theory of the class of all models of Γ. The *syntactic consequences* of Γ, written $\Gamma \vdash$, is $C(S(\Gamma))$, the congruence closure of the substitutive closure of Γ.

Now $C(S(\Gamma))$ is just another way of conveying the same information as is found in the following conventional-looking inference system.

R1. Infer $x = x$ for any term x.
R2. From $x = y$ infer $y = x$.
R3. From $x = y$ and $y = z$ infer $x = z$.
R4. From $x_1 = y_1, \ldots, x_n = y_n$ infer $f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
for any n-ary operation f of T_X.
R5. From axiom $x = y$ infer $h(x) = h(y)$ where h is a substitution.

Rules R1–R4 give the conditions for $\Gamma \vdash$ to be a congruence. R5 forces $\Gamma \vdash$ to contain $S(\Gamma)$. Hence the theory $\Gamma \vdash$ defined by this system is $C(S(\Gamma))$, the congruence closure of $S(R)$.

Lemma 55 *(Soundness)* $\Gamma \vdash \subseteq \Gamma \models$.

Proof: Now $\Gamma \subseteq \Gamma \models$, so $S(\Gamma) \subseteq \Gamma \models$ since $\Gamma \models$ is substitutive, so $C(S(\Gamma)) \subseteq \Gamma \models$ since $\Gamma \models$ is a congruence. ∎

(Note. The system would remain sound if the word "axiom" were omitted from R5, which can only strengthen the system, i.e. no theorems can be lost. This would be equivalent to defining $\Gamma \vdash$ to be the least substitutive congruence containing Γ, but we have already seen in an exercise that $C(S(\Gamma))$ is a substitutive congruence.)

In order to prove the converse of this lemma we introduce the Lindenbaum algebra of Γ, $T_X/\Gamma \vdash$, which exists because $\Gamma \vdash$ is a congruence on T_X.

Lemma 56 *$T_X/\Gamma \vdash$ is a model of Γ.*

Proof: Let $h : T_X \to T_X/\Gamma \vdash$ be an arbitrary homomorphism from T_X to $T_X/\Gamma \vdash$. Take $g : T_X \to T_X$ to be the substitution mapping each generator x_i of T_X to some element of the equivalence class $h(x_i)$, and take $q : T_X \to T_X/\Gamma \vdash$ to be the quotient of T_X by $\Gamma \vdash$. Since the homomorphism qg agrees with h on the generator set of T_X, by construction, it must equal h (since T_X is a term algebra). Now if (s,t) is in Γ, $(g(s), g(t))$ is in $\Gamma \vdash$ since $\Gamma \vdash$ is substitutive. Hence $q(g(s)) = q(g(t))$ by the definition of q, i.e. $h(s) = h(t)$. ∎

Lemma 57 *(Completeness)* $\Gamma \models \subseteq \Gamma \vdash$.

Proof: Let (s,t) be in $\Gamma \models$. Since $T_X/\Gamma \vdash$ is a model of Γ, $q(s) = q(t)$ where $q : T_X \to T_X/\Gamma \vdash$ is the quotient of T_X by $\Gamma \vdash$. But this just expresses membership of (s,t) in $\Gamma \vdash$. ∎

Theorem 58 $\Gamma \vdash = \Gamma \models$.

We conclude that the syntactic and semantic versions of "consequences of Γ" need no longer be distinguished.

Chapter 7

Categories

The set M of all functions $f, g, \ldots : X \to X$ on a set X forms a monoid $(M, \circ, 1_X)$. The operation is that of function composition, $g \circ f$ or gf, with domain M^2. The identity is the identity function 1_X on X.

The obvious generalization of this to multiple sets $X, Y, Z, \ldots$ takes M to consist of all functions between these sets. However M then does not form a monoid under composition for two reasons. First, not all pairs of functions compose; for example we cannot compose two functions both going from X to Y. Instead we restrict the domain of composition to that subset of M^2 consisting of pairs g, f of functions configured as $X \xrightarrow{f} Y \xrightarrow{g} Z$, that is, those pairs for which the codomain or target of f is equal to the domain or source of g. Second, we no longer have a single identity function. Instead there is a separate identity function $1_X : X \to X$ for each set X.

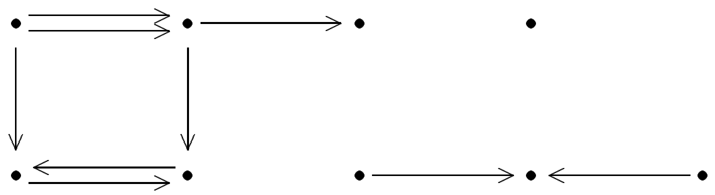
Such a "partial" monoid with multiple identities is called a category. The elements of M are called morphisms and the points connected by the morphisms form a set O of objects.

The objects and morphisms of a category may be viewed as respectively the vertices and edges of a graph. Just as a monoid M has an underlying set $U(M)$, so does a category C have an underlying graph $U(C)$. Like monoids and Boolean algebras, graphs form a class of (two-sorted) algebras of interest in their own right, which we now treat preparatory to our study of categories.

7.1 Graphs

A *graph* $G = (V, E, s, t)$ consists of a set V of vertices, a set E of edges, and a pair of functions $s, t : E \to V$ assigning to each edge $e \in E$ its source vertex $s(e)$ and target vertex $t(e)$.

Here is a representative graph with nine vertices and nine edges.



Category theory differs from graph theory in that it permits more than one edge from one vertex to another, e.g. the two horizontal edges at the upper left in the above example. Except for this difference the situation is as for graph theory. Thus we permit loops, that is, an edge whose source is also its target (none in the example). Graphs need not be connected (the example has three connected components), and not every vertex need be adjacent to an edge (e.g. the vertex at top right), although every edge must have a vertex at each end, namely its source and its target. Graphs can be finite, as in the example, infinite, or even empty (no vertices or edges at all).

The following examples of graphs arise frequently.

• The *n-discrete* graph $n = (\{1, 2, \ldots, n\}, \emptyset, \emptyset, \emptyset)$ consists of n vertices and no edges. A synonym for 0 is the *empty* graph, having no vertices and hence no edges.

• The *n-path* $P_n = (\{0, 1, \ldots, n\}, \{1, 2, \ldots, n\}, s, t)$, with $s(i) = i-1$ and $t(i) = i$, is $0 \to 1 \to 2 \to \ldots \to n$. We call P_0 the *empty path* and P_1 the *unit path.*

• The *n-cone* is as for the n-path but with $s(i) = 0$, e.g. the 2-cone is $1 \leftarrow 0 \to 2$. The n-cone coincides with the n-path for $n \leq 1$. The *n-cocone* is the n-cone with s and t interchanged, e.g. the 2-cocone is $1 \to 0 \leftarrow 2$.

• The *square* is

$$\begin{tikzcd} 0 \arrow[r] \arrow[d] & 2 \arrow[d] \\ 1 \arrow[r] & 3. \end{tikzcd}$$

This generalizes to the n-cube in the obvious way, with the 0-cube being the empty path, followed by the unit path, the square, the cube, etc. The vertices are numbered 0 to $2^n - 1$, with an edge from i to j when $i < j$ and i and j differ in exactly one bit in their binary representation, the so-called Hamming-distance-one criterion.

• The *loop* $(\{1\}, \{1\}, s, t)$ has $s(1) = t(1) = 1$.

• The *parallel pair* $(\{0, 1\}, \{1, 2\}, s, t)$, with $s(i) = 0$, $t(i) = 1$, consists of two edges with a common source 0 and a common target 1.

The numbers constituting the vertices of these graphs are significant, and the graphs must therefore not be considered as defined merely up to isomorphism. Many constructs of category theory are defined in terms of these graphs, and we need to keep track of the identities of the vertices, for example in order to distinguish the first and second projections of a product, of a pullback, etc.

As an occasional notation we shall let n denote the n-discrete graph, $\to$ the 1-path, $\to\to$ the 2-path, $<$ the 2-cone, $>$ the 2-cocone, and $=$ the parallel pair. This notation will come in handy later on with functor categories, as in C^n, $C^{\to}$, $C^{\to\to}$, $C^{<}$, $C^{>}$, and $C^{=}$.

We shall call a graph having at most one edge from one vertex to another a *binary relation* on V. All of the preceding examples are binary relations except for the parallel pair.

The set of edges in G from u to v is called the *homset* $\mathrm{Hom}(u, v)$, also $\mathrm{Hom}_G(u, v)$ or $G(u, v)$. Edges from the same homset are called *parallel.*

7.2 Universal Algebra for Graphs

Like other classes of algebraic structures, graphs have subgraphs, direct products, and homomorphisms. These notions all obey the usual rules of universal algebra, and nothing about the following definitions of those notions is peculiar to graphs per se, besides the signature of course.

A graph (V', E', s', t') is a *subgraph* of (V, E, s, t) when $V' \subseteq V$, $E' \subseteq E$, and s', t' are the respective restrictions of s, t to E'. Hence for any $e \in E'$, $s(e)$ and $t(e)$ must belong to V'. Such a subgraph is called *full* when if u and v are in V' and $e : u \to v$ is in E, then e is in E'.

The *direct product* $G_1 \times G_2$ of graphs (V_1, E_1, s_1, t_1) and (V_2, E_2, s_2, t_2) is the graph $(V_1 \times V_2, E_1 \times E_2, s, t)$ where $s(e_1, e_2) = (s_1(e_1), s_2(e_2))$ and $t(e_1, e_2) = (t_1(e_1), t_2(e_2))$. This generalizes as usual to the direct product $\prod_i G_i$ of a family of graphs.

Homomorphisms are defined for graphs just as for any other class of algebras. Given two graphs $G = (V, E, s, t)$ and $G' = (V', E', s', t')$, a graph homomorphism or *diagram* $D : G \to G'$ is a pair of functions $D_V : V \to V'$ and $D_E : E \to E'$ satisfying $D_V(s(e)) = s'(D_E(e))$ and $D_V(t(e)) = t'(D_E(e))$ for each edge $e \in E$. The composition $D'D : G \to G''$ of homomorphisms $D' : G' \to G''$ and $D : G \to G'$, and the identity homomorphism 1_G for each graph G, are defined as usual for homomorphisms.

For the following examples fix G to be

$$\begin{tikzcd} w \arrow[r, "j"] & x \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & y \arrow[r, shift left=2, "h"] \arrow[r, "k"] & z \arrow[l, shift left=2, "i"] \end{tikzcd}$$

Taking J to be the 9-vertex graph we saw earlier on, a typical diagram $D : J \to G$ is

$$\begin{tikzcd} x \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[d, "g"'] & y \arrow[r, "k"] \arrow[d, "h"] & z & x & \\ y & z \arrow[l, shift right, "i"'] \arrow[l, phantom] & x \arrow[r, "f"] & y & z \arrow[l, "i"'] \end{tikzcd}$$

where in the lower left the pair between y and z is $y \xleftarrow{i} z$ and $y \xrightarrow{h} z$.

We see that some elements of G may appear more than once in a diagram in G, and others not at all. Note that any two edges with the same label must have the same label on their respective sources, and likewise on their targets. However just because two edges have the same labels on their respective sources and targets does not imply that they themselves have the same label.

Certain diagrams are named after their shapes; for example a diagram in G with shape $J = P_n$ is called an n-path in G, or path in G of length n. Similarly we have n-cones in G, parallel pairs in G, etc. Taking G as before, there are four empty paths in G, six unit paths, ten 2-paths, etc.

Exercises. 1. Give a closed form formula in n for the number of n-cones in the above G.

2. Give a closed form formula in m and n for the number of m-paths in the n-cube.

Graphs G and G' are *isomorphic* when there exist diagrams $D : G \to G'$ and $D' : G' \to G$ such that $D'D = 1_G$ and $DD' = 1_{G'}$.

The inclusion of a subgraph into its parent, and the projections of a direct product onto its constituent graphs, are examples of diagrams, just as with monotone maps and monoid homomorphisms, etc.

The *image* $D(J)$ of a graph homomorphism is the graph $(D_V(V), D_E(E), s'', t'')$ where s'', t'' are the restrictions of s', t' in the target of D to $D_E(E)$.

7.3 Constructs Specific to Graphs

Besides the above standard constructs of algebra, applicable to any class of algebras, there are also some constructs specific to graphs.

The *opposite* of a graph $G = (V, E, s, t)$, denoted G^{op}, is the graph (V, E, t, s). It has the same vertices and edges as G, but the direction of the edges has been reversed, as indicated by the interchange of s and t.

The *direct sum* $G_1 + G_2$ of graphs (V_1, E_1, s_1, t_1) and (V_2, E_2, s_2, t_2) is the graph $(V_1 + V_2, E_1 + E_2, s, t)$ where the sums denote disjoint union of sets and $s(e)$ is $s_1(e)$ or $s_2(e)$ depending on whether e came from E_1 or E_2, and similarly for t. Like product this generalizes to the sum $\sum_i G_i$ of a family $(G_i)_{i \in I}$ of graphs.

The *concatenation* $G_1; G_2$ of graphs (V_1, E_1, s_1, t_1) and (V_2, E_2, s_2, t_2) is the graph $(V_1 + V_2, E_1 + E_2 + V_1 \times V_2, s, t)$ where s and t are as for $G_1 + G_2$ but extended so that on $V_1 \times V_2$ they satisfy $s(u, v) = u$ and $t(u, v) = v$. The n-cone is thus isomorphic to $1; n$ and the n-cocone to $n; 1$. (They are not equal however.) On the other hand the graph $P_m; P_n$ is not isomorphic to P_{m+n} in general since $P_m; P_n$ has $(m + 1)(n + 1)$ edges from P_m to P_n. P_{m+n} is constructed from $P_m + P_n$ by adding just one edge from the last element of P_m to the first of P_n, or vice versa.

An essential step in the general construction of a categorical limit is that of the *coning* $\hat{G}$ of a graph G. This is $1; G$, but with the edge set $E_1 + E_2 + V_1 \times V_2$ simplified to $E_2 + V_2$ since E_1 is empty and $|V_1| = 1$. Hence the vertices of $\hat{G}$ are a new vertex $(1, 0)$ and the old vertices $(2, u)$, the edges of $\hat{G}$ are the old edges $(1, e)$ and the new edges $(2, u)$, and s', t' satisfy $s'(1, e) = (2, s(e))$, $t'(1, e) = (2, t(e))$ on the old edges and $s'(2, u) = (1, 0)$, $t'(2, u) = (2, u)$ on the new.

A vertex u is *initial* in a graph when for every vertex u' in the graph there is exactly one edge $e : u \to u'$ in E, i.e. for which $s(e) = u$ and $t(e) = u'$ (so $|\mathrm{Hom}(u, u')| \leq 1$). Dually we call u *final* when there exists exactly one edge *from* each vertex u' to u.

We have said that the vertices and edges of a graph form a set. However we would like the entity **Set** consisting of all sets, viewed as vertices, and all functions between them, viewed as edges, to be a graph. Russell's paradox prevents this. For every set X there exists a set $Y \notin X$, namely $Y = \{x \in X \mid x \notin x\}$.[1] Hence the vertices of **Set** cannot form a set X, since some set Y would then fail to be an object of **Set**. There is however no harm in saying that the objects of **Set** form a "large set" or *class*, an entity not itself a member of the class of (small) sets. We therefore extend our definition of graph to allow V and E to be classes. We call a graph *small* when V and E are both sets. An intermediate notion is *locally small*, namely when $\mathrm{Hom}(u, v)$ is a set for all $u, v \in V$.

7.4 Definition of Category

We pass from graphs to categories by adjoining operations of composition and identity. We call the vertices of categories *objects* and their edges *morphisms*. Domain and codomain are commonly used synonyms for source and target respectively, though we shall not use them here.

A *category* $C = (O, M, s, t, c, i)$ consists of a graph (O, M, s, t) called the *underlying graph* $U(C)$, a partial binary operation $c : M^2 \to M$ called *composition*, and a function $i : O \to M$ associating to each object $x \in O$ the *identity* morphism $i(x) \in M$ on x. Composition and the identities behave as follows.

The composition $c(g, f)$ is notated variously gf, $g \circ f$, or $f; g$. It is defined just when $s(g) = t(f)$, that is, its domain corresponds to the set of 2-paths in $U(C)$. When defined, composition satisfies $s(gf) = s(f)$ and $t(gf) = t(g)$. Furthermore it is associative, that is, it satisfies $(hg)f = h(gf)$ where defined.

[1] This simplifies to $Y = X$ if we assume the Foundation Axiom FA of set theory stating that set membership is well-founded, which prevents $x \in x$ from ever holding. There is some sentiment nowadays in favor of dropping FA for a weaker "Anti-Foundation" Axiom AFA, which permits $x \in x$. There we do need the above more complicated construction of a nonmember Y of X.

The identity morphism $i(x)$ at object x is notated 1_x. It satisfies $s(1_x) = x = t(1_x)$, that is, it is a loop at x. And for each $f : x \to y$, $f1_x = f = 1_y f$, that is, each identity is both a left and a right identity with respect to composition when defined.

This completes the definition of category.

A category is small, that is, O and M are sets, just when its underlying graph is small.

This passage from graphs to categories generalizes that from binary relations to ordered sets. Just as an ordered set is a reflexive transitive binary relation, so is a category a graph with identities and composites. Moreover the generalization is *constructive* in the following sense. Existence of identities is not just a matter of $\mathrm{Hom}(x, x)$ being nonempty for every x, but a particular member of $\mathrm{Hom}(x, x)$ must be specified as the identity. Likewise having composites is not just a matter of $\mathrm{Hom}(x, z)$ being nonempty whenever there exist morphisms $x \xrightarrow{f} y \xrightarrow{g} z$, but a particular member must be specified as gf.

This constructive aspect is vacuous for ordered sets. It is redundant to give c and i explicitly in an ordered set since $c(g, f)$ and $i(x)$ must each be the unique morphism in their respective homsets.

Exercise 1. G is the underlying graph of some category if and only if (i) $\mathrm{Hom}_G(x, x)$ is nonempty and (ii) if $\mathrm{Hom}_G(x, y)$ and $\mathrm{Hom}_G(y, z)$ are nonempty then $\mathrm{Hom}_G(x, z)$ is nonempty,

7.5 Examples of Categories

Here is a representative sample of the many categories that arise naturally.

• Any class of algebraic structures and the homomorphisms between them (whence the term "morphism") form a category. Besides the trivial case **Set** of all sets and functions, there is **Ord**, all ordered sets and monotone maps, **Mon**, all monoids and monoid homomorphisms, **AbMon**, **Mon** restricted to Abelian monoids, and **Grph**, all graphs and diagrams. We will shortly encounter category homomorphisms or functors, turning the class of all small categories into the category **Cat** of all small categories. None of these examples is a small category.

• Omitting "all" in the preceding examples leads to more categories, called subcategories. Thus any class O of sets and class M of functions between those sets containing an identity function 1_X for each X in O, and with M closed under function composition, is a category of sets. The subcategories of **Set** are precisely those categories formed in this way. Similarly we may form subcategories of **Ord**, **Mon**, etc. Such categories, whose morphisms are functions between the underlying sets of their objects, are called *concrete*. They may or may not be small, depending on whether or not O and M are sets.

• Every ordered set $(X, \leq)$ corresponds to a category $(X, \{(x, y) \mid x \leq y\}, s, t, c, i)$ where $s(x, y) = x$, $t(x, y) = y$, $c((y, z), (x, y)) = (x, z)$, and $i(x) = (x, x)$. Transitivity and reflexivity respectively ensure that such a c and i exist. The *discrete category* on X is (the category corresponding to) the ordered set $(X, =)$. The *clique* on X is the category (X, X^2), having no empty homsets. The discrete category and the clique are the two extremal elements among ordered sets on a given set X.

Conversely every category with all homsets of cardinality at most one determines an ordered set. We therefore identify such categories with ordered sets.

• Every monoid $(M, \circ, 1)$ corresponds to a one-object category $(\{\bullet\}, M, K\bullet, K\bullet, \circ, K1)$ where Kx denotes the constant function with value x. Conversely every one-object category must be of the form $(\{x\}, M, Kx, Kx, \circ, K1)$ with $\cdot$ total on M^2, hence determining the monoid $(M, \circ, 1)$. We therefore identify one-object categories with monoids.

• Tuples of terms in a given language L form a category $\mathbf{Trm}_L$ whose objects are the natural numbers and whose morphisms from m to n are n-tuples of terms in m variables, with composition defined as substitution. Thus if L is the language of Boolean algebra then $x_1 \vee x_2$ is a morphism from 2 to 1, $(x_1 \wedge x_2, x_1 \wedge x_3)$ is a morphism from 3 to 2, their composition is the morphism $(x_1 \vee x_2) \wedge (x_1 \vee x_2)$ from 3 to 1, and $(x_1, x_2, \ldots, x_n)$ is 1_n. In such a category certain terms may be identified, e.g. $(x_1 \vee x_2) \wedge (x_1 \vee x_2)$ and $x_1 \wedge (x_2 \vee x_3)$. A category of this kind is called an *algebraic theory*. In such a theory equations take the form of commuting diagrams.

• The $m \times n$ matrices over the reals constitute the morphisms, from m to n, of a category whose objects are the natural numbers, with composition provided by matrix multiplication, and with identities provided by the identity matrices. (For each n there is a unique $0 \times n$ matrix and a unique $n \times 0$ matrix.) Matrices of complex numbers yields another such category. In general any ring R determines a category $\mathbf{Matr}_R$ whose morphisms are matrices over R under the usual matrix multiplication.

• Every graph $G = (V, E, s, t)$ corresponds to a category (V, E', s', t', c, i) where E' is the set of paths in G; s' and t' are such that for each n-path $D : P_n \to G$ in E', $s'(D) = D_V(0)$ and $t'(D) = D_V(n)$; $c(D', D)$ is path concatenation, defined where $s'(D') = t'(D)$; and $i(x)$ is the empty path in G at x. This category is called the *free category on*, or *generated by*, G, written $F(G)$.

7.6 Homogeneity: Objects as Morphisms

Objects are to morphisms as integers are to reals. We may regard the integers as disjoint from the reals but being embeddable in the reals via float: $\mathbf{Z} \to \mathbf{R}$. Or we can simply treat an integer as a kind of real. By the same token we may regard the objects of a category as disjoint from the morphisms but embedded via $i : O \to M$. Or we can treat objects as a special kind of morphism. The former is a heterogeneous view of categories, the latter homogeneous.

In the case of categories of sets, structures, or algebras, the homogeneous view identifies the set X with the identity function 1_X. More generally the homogeneous view identifies the object x with the identity morphism 1_x.

The homogeneous view has the advantage of simplicity: there is only one type to deal with. A category then simplifies to the structure (M, s, t, c), with O and i omitted, and with $s, t : M \to M$ and $c : M^2 \to M$ a partial binary operation satisfying

(i) $ss = s = ts$ and $tt = t = st$;

(ii) gf is defined if and only if $s(g) = t(f)$;

(iii) $s(g) = t(f)$ implies $s(gf) = s(f)$ and $t(gf) = t(g)$;

(iv) $s(g) = t(f) = f$ implies $gf = g$;

(v) $s(g) = t(f) = g$ implies $gf = f$;

(vi) $s(g) = t(f)$ and $s(h) = t(g)$ implies $h(gf) = (hg)f$.

This homogeneous view entails no loss of generality in the definition of the notion of category. For if $f \in s(M)$ then $f = s(g)$ so $t(f) = t(s(g)) = s(g) \in s(M)$. Hence $t(M) \subseteq s(M)$. Similarly $s(M) \subseteq t(M)$), whence $s(M) = t(M)$. Thus we can recover the set O of objects as either $s(M)$ or $t(M)$. Since $O \subseteq M$ we can recover $i : O \to M$ as the inclusion of O into M. And by cutting back the targets of $s, t : M \to M$ to their common range O we recover $s, t : M \to O$.

Whether one takes the heterogeneous or homogeneous view depends on the circumstances. When working with abstract categories it is easier to use the homogeneous formulation in some though not all situations. On the other hand particular categories from our everyday experience such as **Set**, **Ord** and **Grp** tend to be easier to think about when the distinction is drawn between objects and morphisms. Although the correspondence between such structures and the identity functions on them makes the structures themselves redundant, we are not used to thinking about structures in this way and it takes some practice to work with the homogeneous view. The homogeneous view is easier to get used to for abstract categories.

Note that the underlying graph $U(C)$ of a category C assumes the heterogeneous viewpoint: the objects of C become vertices of $U(C)$ whereas the identity morphisms of C become loops.

Exercise 1. Discuss the prospects for viewing graphs homogeneously. What if we defined an intermediate notion of a *reflexive graph*, where we introduced identities but not a composition law? (Without composition identities have no special properties until we come to graph homomorphisms, which should preserve identities.)

7.7 Universal Algebra for Categories

As for other classes of algebraic structures, categories have direct products, subalgebras, and homomorphisms. Direct product is conveniently defined from the homogeneous viewpoint. The *direct product* of categories (M_1, s_1, t_1, c_1) and (M_2, s_2, t_2, c_2) is $(M_1 \times M_2, s, t, c)$ where $s(f_1, f_2) = (s_1(f_1), s_2(f_2))$, $t(f_1, f_2) = (t_1(f_1), t_2(f_2))$, and $c((g_1, g_2), (f_1, f_2)) = (c_1(g_1, f_1), c_2(g_2, f_2))$.

More generally the direct product of a family of categories $(M_i, s_i, t_i, c_i)_{i \in I}$ is $(\prod_i M_i, s, t, c)$ where $s(f)(i) = s_i(f(i))$, $t(f)(i) = t_i(f(i))$, and $c(g, f)(i) = c_i(g(i), f(i))$.

A subcategory of (M, s, t, c) is a set (M', s', t', c') such that $M' \subseteq M$, s', t' are the respective restrictions of s, t to M', and c' is the restriction of c to ${M'}^2$ (though still only a partial operation).

C is always a subcategory of itself, called the *improper* subcategory; all other subcategories of C are *proper*. The empty category, containing neither objects nor morphisms, is the least subcategory of every category. C' is a *full* subcategory of C when $U(C')$ is a full subgraph of $U(C)$, i.e. for any two objects x, y of C', any morphism $f : x \to y$ of C is a morphism of C'.

Exercises. 1. Define "subcategory" from the heterogeneous viewpoint.

2. Enumerate the subcategories of **Set** having as objects $\{0\}$ and $\{0, 1\}$. Show the subcategory relationships between them.

3. The category **Grp** of all groups and group homomorphisms is a full subcategory of **Mon**.

A homomorphism of categories is called a functor. A *functor* $F : (M, s, t, c) \to (M', s', t', c')$ is a function $F : M \to M'$ satisfying $F(s(f)) = s'(F(f))$, $F(t(f)) = t'(F(f))$, and $F(gf) = F(g)F(f)$ when gf is defined.

The identity functor $C \stackrel{1_C}{\to} C$ takes each object and morphism of C to itself. The composition $C \stackrel{F}{\to} D \stackrel{G}{\to} E$, namely $C \stackrel{GF}{\to} E$, is just function composition, which is associative. It follows that the class of all small categories and their functors forms a category, called **Cat**, not itself small.

An *isomorphism* of categories C and D is a pair of functors $C \stackrel{F}{\to} D \stackrel{G}{\to} C$ such that $GF = 1_C$ and $FG = 1_D$. F and G are then each called isomorphisms, and C and D are called *isomorphic*. Isomorphism is an equivalence relation between categories.

The behavior of a functor $F : C \to D$ on any one homset $\mathrm{Hom}_C(x, y)$ in C can be described as a function $F_{xy} : \mathrm{Hom}_C(x, y) \to \mathrm{Hom}_D(F(x), F(y))$. If for all x, y F_{xy} is injective then F is said to be *faithful*, and if surjective then F is said to be *full*. That is, a functor is *faithful* when distinct morphisms with the same source and target are mapped to distinct morphisms, and *full* when for any composite hgf in the target, if h and f are in the image of F so is g, a sort of convexity property. Thus to say that $C' \subset C$ is a *full* subcategory of C is to say that the inclusion functor from C' to C, necessarily faithful, is also full.

7.8 Category-specific Constructs

In addition to the general constructs of universal algebra there are certain constructs peculiar to categories. The *opposite* of a category $C = (M, s, t, c)$ is the category $C^{\mathrm{op}} = (M, t, s, \breve{c})$. Here $\breve{c}$ denotes the converse of c, satisfying $\breve{c}(g, f) = c(f, g)$. This corresponds to reversing the directions of the morphisms, and hence of composition. Evidently $(C^{\mathrm{op}})^{\mathrm{op}} = C$ and $U(C^{\mathrm{op}}) = U(C)^{\mathrm{op}}$.

For any category C of sets and functions, C^{op} has the same objects and morphisms as C. However these morphisms do not act as in C. In $\mathbf{Set}^{\mathrm{op}}$ for example there are as many distinct morphisms from a given set

X to the singleton $\{0\}$ as there are elements in X. Thus although the morphisms of $\mathbf{Set}^{\mathrm{op}}$ are functions, the source of a function is its codomain and the target its domain.

If the functions of C were all bijections then it is easy to find a category of sets of functions isomorphic to C^{op}: just take the morphisms of C^{op} to be the inverses of those bijections.

Here is a more general method of constructing, for any category C of sets and functions, a category D of sets and functions isomorphic to C^{op}. The trick is to allow the sets of D to be considerably larger than the corresponding sets of C.

Pick any set Z having at least two elements. For each object X of C take Z^X to be an object of D, where Z^X denotes the set of all functions from Z to X. For each function $f : X \to Y$ in C equip D with a corresponding function $Z^f : Z^Y \to Z^X$ defined on Z^Y as $Z^f(g) = gf$.

Exercises. 1. Show that D so constructed is isomorphic to C^{op}.

2. Show that $\mathbf{Set}^{\mathrm{op}}$ is not isomorphic to $\mathbf{Set}$ but that it is isomorphic to a subcategory of $\mathbf{Set}$.

A *monic* is a morphism f such that if $fg = fh$ then $g = h$ (cancellation on the left). An *epi* is a morphism f such that if $gf = hf$ then $g = h$ (cancellation on the right). A *bimorphism* is a morphism which is both a monic and an epi.

An *isomorphism pair* is a pair of morphisms $f : x \to y$, $g : y \to x$ such that $gf = 1_x$ and $fg = 1_y$. Two objects with an isomorphism pair between them are called *isomorphic*. Isomorphism is an equivalence relation. An *isomorphism class* of C is the class of all objects of C isomorphic to a given object X of C, and may be denoted $[X]$. (It follows that isomorphism classes are nonempty.)

An *endomorphism* is a loop morphism, one whose source is its target. An *automorphism* is a loop isomorphism. Identities are a special case of automorphisms.

Exercises. 3. Show that every isomorphism is a bimorphism.

4. Show that in $\mathbf{Set}$ every bimorphism is an isomorphism.

5. List all two-element monoids that, as categories, contain a bimorphism that is not an isomorphism?

A *groupoid* is a category all of whose morphisms are isomorphisms. A *skeletal* category C is one all of whose isomorphisms are automorphisms. A *group* is a skeletal groupoid, or equivalently a groupoid that is a monoid. A partial order is a skeletal preorder. But $\mathbf{Set}$ is not skeletal because any two sets of the same finite cardinality n have $n!$ isomorphisms from one to the other.

A *skeleton* of a category C is the skeletal category C' obtained by taking for the objects of C' one representative of each isomorphism class of objects of C, and for the homsets of C' the corresponding homsets of C. Two categories are called *equivalent* when they have isomorphic skeletons.

7.9 Graphs, Free Categories, and Diagrams

To each category C corresponds its underlying graph $U(C)$. This defines the object part of a functor $U : \mathbf{Cat} \to \mathbf{Grph}$, the *forgetful functor* from $\mathbf{Cat}$ to $\mathbf{Grph}$. The morphism part of U takes each functor $G : C \to C'$ to the diagram $U(G) : U(C) \to U(C')$ defined just as for G but with no mention of composition and identities.

The section on examples of categories included the free category $F(G)$ on a graph G. This defines the object part of a functor $F : \mathbf{Grph} \to \mathbf{Cat}$ taking graphs to categories. For the morphism part, given a diagram (morphism of graphs) $D : G \to G'$, define $F(D) : F(G) \to F(G')$ as $F(D)(D') = DD'$ where $D' : P_n \to G$ is a path in G and hence a morphism of $F(G)$. That is, $F(D)$ is the functor (morphism of categories) from $F(G)$ to $F(G')$ taking each n-path $D' : P_n \to G$ in G (i.e. each morphism of $F(G)$) to the n-path in G' (morphism of $F(G')$) consisting of the images under F of the morphisms along the path D', in order. This defines a functor $F : \mathbf{Grph} \to \mathbf{Cat}$.

An equivalent way of thinking about the specification of composition and identities is to regard them collectively as the specification of a map from paths in C to morphisms. Composition specifies the map from paths of length 2. The map is extended to longer paths by applying composition repeatedly, with associativity ensuring that the result will be unambiguous. The identities determine the map at paths of length 0. And paths of length 1 are automatically mapped to their one morphism.

Exercises. 4. Define "functor" from the heterogeneous viewpoint. Show that your definition is consistent with the homogeneous one.

5. For each of the basic examples of graphs given at the beginning, give the number of morphisms in the free category it generates.

Recall that a diagram $D : J \to G$ is said to be a diagram in G. Of special importance is the case $G = U(C)$ of a diagram in the underlying graph of a category C. In this case, by "diagram in C" we shall understand "diagram in $U(C)$."

There is a bijection between diagrams $D : G \to U(C)$ and functors $D' : F(G) \to C$. Any assignment of vertices and edges of G to the vertices and edges of $U(C)$ extends in the obvious way to a functor mapping paths of $F(G)$ to morphisms of C. Conversely any such functor restricts to a diagram on G in $U(C)$. The situation is essentially as for free monoids on sets, where we have a bijective correspondence between functions $f : X \to U(M)$ and monoid homomorphisms $h : F(X) \to M$.

For example let G be a square graph, one with four vertices and four edges, and let C include among its objects w, x, y, z, and among its morphisms $f : w \to x$, $g : w \to y$, $h : x \to z$, and $i : y \to z$. Then the diagram

$$\begin{tikzcd} w \arrow[r, "f"] \arrow[d, "g"'] & x \arrow[d, "h"] \\ y \arrow[r, "i"] & z \end{tikzcd}$$

is a square in C depicting $D : G \to U(C)$ explicitly. Implicitly it also depicts $D' : F(G) \to C$, by mapping the four empty paths of $F(G)$ to the corresponding identities of C, and the two paths from top left to bottom right to the respective composites ig and hf.

Such a diagram is said to *commute* when, viewed as a functor $D' : F(G) \to C$, it maps parallel paths in $F(G)$ (paths with a common source and a common target) to the same morphism of C. The example has only one case of parallel paths, namely the two sides of the square from top left to bottom right, and hence commutes just when $hf = ig$.

We remark that any diagram in an ordered set automatically commutes.

7.10 Products and Coproducts

(For this topic the heterogeneous viewpoint proves more convenient.)

Ordinarily we specify a Boolean algebra by giving say the operations $\vee$, $\wedge$, and $\neg$. But we could actually specify it just as an ordered set, since $\vee$ and $\wedge$ can be recovered from the order as supremum and infimum, which are uniquely determined in any partial order when they exist, while $\neg$ is complement, uniquely determined in any distributive lattice when it exists.

A similar situation arises with categories. Once we have the basic category, corresponding to an ordered set, we can recover final and initial elements, corresponding respectively to top and bottom of an ordered set, and products and coproducts of a set of objects, corresponding respectively to infimum or greatest lower bound and supremum or least upper bound of a set of elements.

One difference is that categories in general are like preorders rather than partial orders. In a preorder a set may have more than one infimum, though the infima of a set are related in that they form a (maximal) clique. Another difference is that there are also other relationships we can recover from the category structure, such as that of being an equalizer, a pullback, and more generally a limit, and their respective duals. With ordered sets the only limits and colimits are infima and suprema respectively.

The notions of greatest and least element of an ordered set generalize to categories as the notions of final and initial object respectively. A *final object* z of a category C is a final vertex of the underlying graph. That is, it is an object such that for every object y of C (including z), there is exactly one morphism $h : y \to z$. Dually an *initial* object z is one such that for every object y there is exactly one morphism $h : z \to y$.

For example **Set** has just one initial object, namely the empty set $\emptyset$, since there is exactly one function from the empty set to any set. However it has many final objects, namely all singletons.

The free category on the n-path, containing $\binom{n+2}{2}$ morphisms, has initial object 0 and final object n. The free category on the n-cone has initial object 0, but no final object unless $n \leq 1$. Dually 0 is the final object of the free category on the n-cocone.

An initial object can have no nontrivial automorphisms, otherwise it would have more than one morphism from itself back to itself.

Now for any category C consider the full subcategory C' whose objects are the initial objects of C. It is easily seen that C' is a clique. It follows that any two initial objects of C are isomorphic in C. We sometimes say that a category has at most one initial object *up to isomorphism*. The same holds for final objects. In **Set** the final objects are singletons, all isomorphic (which in **Set** just means having the same cardinality).

We turn now to products. First let us give a familiar example. In the category **Set**, the cartesian product $X_1 \times X_2$ of sets X_1 and X_2, itself a set, has two associated projection functions p_1, p_2 that we can diagram as the 2-cone $(p_i : X_1 \times X_2 \to X_i)_{i=1,2}$ (that is, $X_1 \overset{p_1}{\leftarrow} X_1 \times X_2 \overset{p_2}{\to} X_2$) in **Set**. We call this diagram a *cone to* the pair (X_1, X_2).

As we saw when studying sets, this cone is *universal* among cones to (X_1, X_2). This means that for any cone $(g_i : Y \to X_i)_{i=1,2}$ there exists exactly one function $g : Y \to X_1 \times X_2$ such that $p_1 g = g_1$ and $p_2 g = g_2$, namely the function $g(y) = (g_1(y), g_2(y))$. Conversely any $g : Y \to X_1 \times X_2$ uniquely determines a cone to (X_1, X_2), namely $(p_i g)_{i=1,2}$. We call g the *representative* of the cone $(g_i : Y \to X_i)_{i=1,2}$, and say that the cone g_i *factors through* the cone p_i as $p_i g$.

Universality is *abstract* in the sense that it is phrased purely in terms of sets, functions, composition, and identities. No mention is made of elements of sets or of application of functions. This abstractness permits the notion of product to be formulated for an arbitrary category.

A cone $p_i : z \to x_i)_{i=1,2}$ in a category C is called a *product* of x_1 and x_2, with *projections* p_1, p_2, when every cone $(g_i : y \to x_i)_{i=1,2}$ factors through p_i in exactly one way, i.e. when there exists exactly one morphism $g : y \to z$ for which $p_1 g = g_1$ and $p_2 g = g_2$.

When every pair x_1, x_2 of objects in a category C has a product in C we say that C *has binary products.*

Note the similarity with final objects, whose definition has this same universal character. Later we shall show that the natural organization into a category of 2-cones to a pair has for its final objects the products of that pair.

For any given x_1 and x_2, the property of being a product of x_1 and x_2 is relational rather than functional: there may be multiple products. An example of this in **Set** is provided by $X_1 \times X_2$ and $X_2 \times X_1$, which are products of X_1 and X_2, the latter having respective projections $p_2 : X_2 \times X_1 \to X_1$ and $p_1 : X_2 \times X_1 \to X_2$. $X_2 \times X_1$ is in general a different set from $X_1 \times X_2$.

This nonfunctional aspect of product notwithstanding, we shall often refer to a particular product of x_1 and x_2 as $x_1 \times x_2$. In some situations context will provide a specific functor $\times : C^2 \to C$, such as cartesian product in **Set**, picking out a particular product of each pair. Other times there will be no particular product functor, but rather $x_1 \times x_2$ will merely be serving as a more mnemonic identifier than z for some arbitrarily chosen product of x_1 and x_2.

The following provides some additional justification for this convention.

Proposition 59 *Any two products of two objects x_1 and x_2 are isomorphic.*

Proof: The proof runs along similar lines to the proof that any two final objects are isomorphic. Let $x_1 \xleftarrow{p_1} z \xrightarrow{p_2} x_2$ and $x_1 \xleftarrow{p'_1} z' \xrightarrow{p'_2} x_2$ be two such products. These factor through each other and through themselves in four combinations. Through each other we obtain $p' : z' \to z$ representing (p'_1, p'_2) with respect to (p_1, p_2), and $p : z \to z'$ representing (p_1, p_2) with respect to (p'_1, p'_2). Through themselves we must obtain 1_z and $1_{z'}$. We then have the following diagram.

$$\begin{tikzcd} z \arrow[r, "p_2"] \arrow[d, "p_1"'] \arrow[dr, shift left, "p"] & x_2 \\ x_1 & z' \arrow[l, "p'_1"] \arrow[u, "p'_2"'] \arrow[ul, shift left, "p'"] \end{tikzcd}$$

Now $p_1 p' p = p'_1 p = p_1$ and $p_2 p' p = p'_2 p' = p_2$. But we also have $p_1 1_z = p_1$ and $p_2 1_z = p_2$. Since there is only one such morphism, $p' p = 1_z$. Similarly $pp' = 1_{z'}$. Hence z and z' are isomorphic. (This argument also shows that the above diagram commutes.) ∎

As an application we infer that the two cartesian products $X_1 \times X_2$ and $X_2 \times X_1$ are isomorphic, both being abstract products of the same pair. This is an argument by *general nonsense*. An *elementary proof* of the same fact follows from the 1-1 correspondence between pairs (x_1, x_2) and (x_2, x_1). The latter proof depends on sets having elements and does not generalize to arbitrary categories, where arguments by general nonsense come into their own.

In a skeletal category products are unique.

The notion of a ternary product generalizes that of binary product in the obvious way. A ternary product of objects x_1, x_2, x_3 is a diagram consisting of an object z and morphisms $p_i : z \to x_i$ for $i = 1, 2, 3$, universal among such diagrams.

Exercise 1. Extend the preceding proposition to show that ternary products are unique up to isomorphism.

Proposition 60 *A category with binary products has ternary products.*

Proof: Let $x_1 \times x_2$ be a product with projections q_1, q_2 and let $z = (x_1 \times x_2) \times x_3$ be a product with projections p_1, p_2, yielding the 3-cone $(q_1 p_1, q_2 p_1, p_2)$ from z to (x_1, x_2, x_3). We claim this 3-cone is the desired ternary product, for which it suffices to show its universality.

Let $(g_i : y \to x_i)_{i=1,2,3}$ be any 3-cone to (x_1, x_2, x_3). Let $g_{12} : y \to x_1 \times x_2$ represent (g_1, g_2), namely as $(q_1 g_{12}, q_2 g_{12})$, and let $g : y \to z$ represent (g_{12}, g_3), as $(p_1 g, p_2 g)$. Hence g represents $(g_i : y \to x_i)_{i=1,2,3}$, as $(q_1 p_1 g, q_2 p_1 g, p_2 g)$. Now for any other such representative $g' : y \to z$, $p_1 g'$ represents (g_1, g_2) and hence equals g_{12}. But g' then represents (g_{12}, g_3) and hence equals g. Hence (g_1, g_2, g_3) has a unique representative, whence $(q_1 p_1, q_2 p_1, p_2)$ is universal. ∎

Corollary 61 *In any category, binary product is associative up to isomorphism.*

Proof: The previous proof showed that $(x_1 \times x_2) \times x_3$ is a ternary product of (x_1, x_2, x_3). A similar proof shows this for $x_1 \times (x_2 \times x_3)$. Hence the two are isomorphic. ∎

A familiar example of this is cartesian product in **Set**, which is not associative because $X\times(Y\times Z)$ is not equal to $(X\times Y)\times Z$ in general. These two sets are however isomorphic, e.g. via the bijection α_{XYZ} : $X\times(Y\times Z) \to (X\times Y)\times Z$ defined as $\alpha_{XYZ}(x,(y,z)) = ((x,y),z)$.

The dual of product is coproduct. In any category C, a diagram $x_1 \xrightarrow{p_1} z \xleftarrow{p_2} x_2$ is called a *coproduct* of x_1 and x_2 when for every diagram $x_1 \xrightarrow{p'_1} z' \xleftarrow{p'_2} x_2$ there exists exactly one morphism $h : z \to z'$ for which $hp_1 = p'_1$ and $hp_2 = p'_2$. This is exactly as for the definition of product with all the morphisms and compositions reversed. Everything we have said about binary and ternary products applies to coproducts *mutatis mutandis.*

Exercises.

2. Define product $\prod_{i\in I} x_i$ and coproduct $\coprod_{i\in I} x_i$ of a family $(x_i)_{i\in I}$ of objects of a category C. Show their uniqueness up to isomorphism.

3. Show that a category with binary products has n-ary products for all finite $n \geq 1$.

4. Show that zeroary product is the same notion as final object. Exhibit a category having all finite nonempty products but no final object.

5. For small categories C and D, show that $C\times D$ is a product of C and D in **Cat**.

6. In **Set** the disjoint union $X+Y$ of sets X, Y is defined as $\{1\}\times X \cup \{2\}\times Y$, consisting of elements of the forms $(1,x)$ for $x \in X$ and $(2,y)$ for $y \in Y$. Show that $X+Y$ is a coproduct of X and Y in **Set**.

7. What are binary product and coproduct in the subcategory of **Set** whose morphisms are just the inclusions? (An inclusion is a function $f : X \to Y$ satisfying $f(x) = x$ for all $x \in X$.)

7.11 Equalizers and Pullbacks

Products are a special case of limits, and coproducts of colimits. Before treating the general case we shall investigate further commonly encountered instances of these notions.

In **Set** the *concrete equalizer* of two parallel functions $f_1, f_2 : X_1 \to X_2$ is the subset Z of X_1 defined as $Z = \{x \in X_1 \mid f_1(x) = f_2(x)\}$. The subset aspect can be expressed more functionally as an inclusion $p_1 : Z \to X_1$ such that $f_1p_1 = f_2p_1$, that is, f_1 and f_2 are "equalized" by composition on the right with the inclusion. We let $p_2 : Z \to X_2$ denote the function they are equalized to.

This set Z and function p_1 can be seen to have the property that for any set Y and function $g_1 : Z' \to X_1$ satisfying $f_1p'_1 = f_2p'_1$, there exists exactly one function $g : Y \to Z$ such that $p_1g = g_1$, namely $g(y) = p_1^{-1}(g_1(y))$, well-defined since the range of g_1 must be a subset of the range of p_1.

We call any set Z and function $p_1 : Z \to X_1$ with this property an *equalizer* of f_1 and f_2. Thus the concrete equalizer is one of (possibly many) equalizers.

An *equalizer* of a parallel pair $f_1, f_2 : x_1 \to x_2$ in C is an object z of C and a universal morphism $p_1 : z \to x_1$ such that $f_1p_1 = f_2p_1$ ($= p_2$ say), that is, f_1 and f_2 are "equalized." As with product, "universal" means that for any $g_1 : y \to x_1$ satisfying $f_1g_1 = f_2g_1$ ($= g_2$ say), there exists exactly one morphism $g : y \to z$ such that $p_1g = g_1$ (and hence $p_2g = g_2$).

Equalizers closely resemble products. Whereas a product is a 2-cone to a pair of objects, an equalizer is a 1-cone $p_1 : z \to x_1$ to a parallel pair of morphisms $f_1, f_2 : x - 1 \to x_2$, or a 2-cone, strengthening the resemblance, if we include the redundant $p_2 : z \to x_2$ defined as f_1p_1. Products and equalizers are both universal among their respective classes of cones. The essential difference is that both equalizers and the cones they compete with for universality must satisfy an additional condition $f_1p_1 = f_2p_1$.

The dual of equalizer is coequalizer.

Exercises. 1. Show that equalizers are unique up to isomorphism.

2. Define the concrete coequalizer of two functions $f, g : X \to Y$ to be the set $Y/\equiv$ of equivalence classes in Y where $\equiv$ is the least equivalence relation such that $f(x) \equiv g(x)$ for all $x \in X$. Show that concrete coequalizers are (abstract) coequalizers in **Set**.

We generalize the notion of equalizer to that of pullback by dropping the requirement that the given morphisms f_1 and f_2 of an equalizer have a common source. A *pullback* of morphisms $f_1 : x_1 \to x_3$ and $f_2 : x_2 \to x_3$ is a commuting square

$$\begin{tikzcd} z \arrow[r, "p_2"] \arrow[d, "p_1"'] & x_2 \arrow[d, "f_2"] \\ x_1 \arrow[r, "f_1"] & x_3 \end{tikzcd}$$

such that for any commuting square (the competition)

$$\begin{tikzcd} y \arrow[r, "g_2"] \arrow[d, "g_1"'] & x_2 \arrow[d, "f_2"] \\ x_1 \arrow[r, "f_1"] & x_3 \end{tikzcd}$$

there exists exactly one morphism $g : y \to z$ such that $p_1 g = g_1$ and $p_2 g = g_2$.

Whereas with equalizers we made a 1-cone into a 2-cone by adding a redundant projection, in this case we turn a 2-cone into a 3-cone with the redundant projection $p_3 = f_1 p_1 = f_2 p_2$.

The dual of pullback is not "copullback" but *pushout.*

A pullback for which $f_1 = f_2$ is called a *kernel pair*, and its dual a *cokernel pair.*

Exercises. 3. Show that pullbacks and pushouts are unique up to isomorphism.

4. Show that a category with all finite nonempty products and equalizers also has all pullbacks.

5. Apply your solution to the previous exercise to define concrete pullbacks in **Set** in terms of cartesian (concrete) products and concrete equalizers. Express this concept in elementary terms.

6. Dualize the previous exercise in terms of disjoint unions and concrete coequalizers.

7. Derive an elementary description of concrete kernel pairs in **Set**. Using the graph representation for relations, define equivalence relations in terms of 2-cones in **Set**. Show that a kernel pair in **Set** is isomorphic to an equivalence relation, and that every equivalence relation arises in this way.

8. Derive an elementary description of concrete cokernel pairs in **Set**.

7.12 Limits

All of these notions—final object, product, equalizer, pullback, and their respective duals—are subsumed under one notion of limit, and its dual notion, colimit, defined as follows. Recall from the section on constructs specific to graphs the coning $\hat{G} = 1; G$ augmenting a graph G with a cone to G.

Let $D : G \to U(C)$ be a diagram in C, and write its vertex labels $D_V(v)$ as x_v and its edge labels $D_E(e)$ as f_e. A *cone to D* is a diagram $D' : \hat{G} \to U(C)$ augmenting D with a vertex labeled y and for each vertex v of G an edge labeled $g_v : y \to x_v$, such that $f_e g_{se} = g_{te}$ for each edge e of G. Equivalently, D' maps all paths from the apex to v in $F(G)$ to $g_v : y \to x_v$ in C.

A *limit* of D is a *universal* cone to D. That is, it is a cone $(p_v : z \to x_v)_{v \in V}$ to D such that every cone $(g_v : y \to x_v)_{v \in V}$ to D has exactly one morphism $g : y \to z$ for which $p_v g = g_v$ (the v-th projection of g is g_v) for all $v \in V$.

Thus a limit combines the notions of product and equalizer in the one construct. We use the notation $\lim D$ for z, by analogy with the notation $x \times y$ and with corresponding caveats about nonuniqueness.

Exercises. 1. Define colimit.

2. Show that limits and colimits are unique up to isomorphism.

3. Explain why equalizers and pullbacks are limits.

4. When $p_u = 1_{x_u}$ in some limit of D we say that u itself is a limit of D. When G is (i) an n-discrete graph; (ii) an n-path; (iii) an n-cone; (iv) an n-cocone; (v) an n-cube, which vertices of G are limits in every diagram on G? Give a general condition for a vertex of G to be a limit in every diagram on G.

The kind of a limit to a diagram D, whether product, pullback, etc., is determined by the shape of D. Discrete graphs give rise to products, parallel pairs to equalizers, cocones to pullbacks. From exercise 9 we infer that certain other kinds of limits are trivial inasmuch as such limits can already be found within the diagram.

A finite (countable, small, etc.) limit means a limit of a finite (countable, small, etc.) diagram.

Proposition 62 *A category with finite products and equalizers has finite limits.*

Proof: Given a diagram D in C on finite $G = (V, E, s, t)$, let $\prod_{v \in V} x_v$ be a product of the objects of D with projections $(p_v : \prod_{v \in V} x_v \to x_v)_{v \in V}$. Let the two cones $(f_e p_{se} : \prod_{v \in V} x_v \to x_{te})_{e \in E}$ and $(p_{te} : \prod_{v \in V} x_v \to x_{te})_{e \in E}$ have respective representatives $g, h : \prod_{v \in V} x_v \to \prod_{e \in E} x_{te}$ with respect to $\prod_{e \in E} x_{te}$.

Exercise 5. (i) Show that for all z and all $q : z \to \prod_{v \in V} x_v$, q is a cone to the parallel pair (g, h) if and only if $(p_v q : z \to x_v)_{v \in V}$ is a cone to D. (ii) Infer that q is an equalizer of (g, h) if and only if $(p_v q)_{v \in V}$ is a limit of D.

Hence any such equalizer yields a limit of D. ∎

"Finite" may be replaced throughout this proposition and its proof by any infinite cardinal, or by "small".

Exercise 6. Any category with a final object and all pullbacks of a given cardinality has all products of that cardinality, and all equalizers.

7.13 Limits and Colimits in Various Categories

Thus far we have been content to draw all our examples of limits and colimits from **Set**. In this section we explore further afield.

With the concluding two propositions of the previous section in mind, we will only treat products, equalizers, coproducts, and coequalizers. Pullbacks, pushouts, and other limits and colimits can then be easily inferred. Since such limits arise frequently in practice the reader will find it worthwhile to carry out these inferences as exercises.

Limits in C^{op} are just colimits in C and vice versa, doubling at one stroke the number of categories both whose limits and colimits we understand.

Proposition 63 *The limits and colimits of a diagram D in an ordered set are the infs and sups respectively of the objects of D.*

Proof: The only influence exerted by the morphisms of a diagram D on its limits and colimits is via commutativity requirements. But diagrams commute automatically in an ordered set, so their morphisms make no difference. Hence limits are products, i.e. infs, while colimits are coproducts, i.e. sups. ∎

The subcategory of **Set** whose morphisms are the inclusions of **Set** is a partial order, ordered by inclusion. Hence limits and colimits are products and coproducts, namely intersection and union. All such products and coproducts exist in this category, with one exception: the empty product or final object does not exist, there being no set of all sets. Equalizers and coequalizers are trivial: parallel pairs are equal to begin with.

Now broaden the class of morphisms of **Set** to include all partial functions $f : X \to Y$, those subsets of $X \times Y$ for which (x, y) and (x, y') both in the subset implies $y = y'$. The first thing to notice is that no new isomorphisms are created (and of course none are lost).

There are 2^n partial function from a set of n elements to a singleton, so singletons are not final. There is just one partial function from X to the empty set $\emptyset$, namely the nowhere-undefined function. And there is just one function from $\emptyset$ to X, which is both the everywhere-defined and the nowhere-defined function on $\emptyset$. Thus the category of partial functions has the empty set as both its initial and final object. When the initial objects coincide with the final objects they are called *null* objects.

If we further broaden the morphisms of **Set** to all binary relations from X to Y, i.e. subsets of $X \times Y$, $\emptyset$ remains the one null object.

Regarding **Set** as the trivial category of algebras, the simplest nontrivial category of algebras is $\mathbf{Set}_*$, pointed sets, i.e. algebras whose signature consists of just one zeroary operation or constant, and their homomorphisms. Such an algebra has the form $(X, *)$ where $* \in X$ is the constant or "pointed element." A homomorphism $f : (X, *_X) \to (Y, *_Y)$ is a function $f : X \to Y$ satisfying $f(*_X) = *_Y$.

Exercise. 1. Show that products, equalizers, coproducts, and coequalizers in $\mathbf{Set}_*$ all exist, and have concrete definitions identical to those in **Set**, with the exception of a detail in coproduct. What is that detail?

The category **Ord** of ordered sets and their monotone maps behaves as for **Set**. Products, equalizers, coproducts, and coequalizers are obtained in terms of those operations on the underlying sets. The coproduct of $(X, \leq_X)$ and $(Y, \leq_Y)$ is $(X + Y, \leq)$ where $\leq$ is the least relation whose respective restrictions to X and Y are $\leq_X$ and $\leq_Y$.

The category **Pos** of partially ordered sets and their monotone maps differs in one detail from **Ord**. Coequalizers no longer behave as in **Set**.

Proposition 64 *Any equational class has all limits.*

Proposition 65 *Every product in* **AbMon** *is a coproduct and conversely.*

Chapter 8

Adjunctions

8.1 Natural Transformations

The need for natural transformations is felt both in categorical algebra and categorical logic.

Categorical algebra equips categories with functors just as ordinary algebra equips sets with operations. One such operation is the exponentiation functor Y^X from $\mathbf{Set}^2$ to $\mathbf{Set}$. This functor maps the pair X, Y of sets to the set of all functions from X to Y. A characteristic property of exponentiation is that to each function $f : X \times Y \to Z$ there corresponds an equivalent function $f' : X \to Z^Y$. We may think of this correspondence as the "currying" law for $\mathbf{Set}$.

The analogous notion for $\mathbf{Cat}$ would be the exponentiation functor B^A which from two *categories* A and B forms the *category* of all functors from A to B. As for $\mathbf{Set}$ we want the corresponding characteristic correspondence between functors $F : A \times B \to C$ and $F' : A \to C^B$. Now the objects of the category C^B are functors from B to C, but what are its morphisms? In order for this correspondence to work we need a specific notion of morphism of functors. This notion is supplied by natural transformations, as we shall see after we have defined them.

Categorical logic extends logic based on ordered sets to logic based on categories. The essential change is that whereas an ordered set has at most one morphism from any object to another, a category permits many. This change is reflected in the form that laws take.

Typical laws are $x \wedge y \leq x$ and $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. For any given values of x, y, z the truth of these laws in an ordered set is a simple binary matter: they either hold or not.

If we regard an ordered set as a category then to say that $x \wedge y \leq x$ holds is to say that for each x and y there is a morphism from the object $x \wedge y$ to the object x. The question of which morphism does not come up because there is at most one morphism from one object, here $x \wedge y$, to another, here x, in an ordered set.

The appropriate generalization of such laws to other categories besides ordered sets requires that the law not only hold for all values of its variables but that in doing so it specify a particular morphism for each choice of values. This morphism can be thought of as an abstract proof of that law. This is the essential difference between ordinary logic and categorical logic. The result is a logic with a more constructive flavor: laws include their proofs.

For example in $\mathbf{Set}$, the associativity law that held for $x \wedge y$ in partial orders does not hold in the usual sense for the cartesian product $X \times Y$. The sets $X \times (Y \times Z)$ and $(X \times Y) \times Z$ are equal only when at least one of X, Y, or Z is empty. Nevertheless there is a weaker sense in which cartesian product is associative, namely that $X \times (Y \times Z)$ and $(X \times Y) \times Z$ are isomorphic. We formalize this by having the law take the form of a specific family of functions $\alpha \times_{XYZ} : X \wedge (Y \wedge Z) \to (X \wedge Y) \wedge Z$, indexed by objects X, Y, Z of $\mathbf{Set}^3$, each defined as $\alpha \times_{XYZ} (x, (y, z)) = ((x, y), z)$. Similarly the closest we come to having an associativity law for disjoint union is the family of functions $\alpha{+}_{XYZ} : X + (Y + Z) = (X + Y) + Z$ each satisfying $\alpha +_{XYZ} (1, x) = (1, (1, x))$, $\alpha +_{XYZ} (2, (1, y)) = (1, (2, y))$, and $\alpha +_{XYZ} (2, (2, z)) = (2, z)$.

Likewise the poset law $x \wedge y \leq x$ when transferred to **Set** must be turned into a specific function from $X \times Y$ to X. A suitable such function is the projection function $\pi 1_{XY} : X \times Y \to X$, defined as $\pi_i(x, y) = x$. Likewise $x \leq x \vee y$ may be turned into the inclusion $i1_{XY} : X \to X + Y$.

Categorical logic permits new laws to be inferred from old, for example by composition. From premises $\pi 1_{XY}$ and $i1_{XY}$ we may infer the law $i1_{XY} \pi 1_{XY} : X \times Y \to X + Y$ by composing the premises. This generalizes the inference rule of transitivity of $\leq$.

Another way to relate $X \times Y$ to $X + Y$ is via $i2_{XY} \pi 2_{XY} : X \times Y \to X + Y$, passing through Y instead of X. This gives us two different laws relating the same pair of objects $X + Y$ and $X \times Y$.

We see therefore that a law consists of a family of morphisms from $F(x_1, \ldots, x_n)$ to $G(x_1, \ldots, x_n)$, one morphism for each $x_1, \ldots, x_n$. We take the index set of this family to consist of the objects of some category C. For example if each x_i ranges over the objects of a category C_i then C would be $\prod_i C_i$. We then formalize the terms $F(x_1, \ldots, x_n)$ and $G(x_1, \ldots, x_n)$ as functors $F, G : C \to D$.

With these motivations in mind we now define natural transformation.

Given parallel functors $F, G : C \to D$, a *transformation* $\tau : F \to G$ is a function mapping each object x of C to a morphism τ_x of D.

For the family $\pi 1_{XY} : X \times Y \to X$ of projections, C is $\mathbf{Set}^2$, D is **Set**, $F : \mathbf{Set}^2 \to \mathbf{Set}$ is $X \times Y$ and $G : \mathbf{Set}^2 \to \mathbf{Set}$ is X (projection onto the first coordinate of $\mathbf{Set}^2$). This makes $\pi 1$ a transformation. In our example of associativity of $\times$ in **Set**, C is $\mathbf{Set}^3$, D is **Set**, F is $X \wedge (Y \wedge Z)$, and G is $(X \wedge Y) \wedge Z$, making $\alpha\times$ a transformation.

A transformation is called *natural* when for each morphism $f : x \to x'$ of C the following square commutes.

$$\begin{tikzcd} F(x) \arrow[r, "F(f)"] \arrow[d, "\tau_x"'] & F(x') \arrow[d, "\tau_{x'}"] \\ G(x) \arrow[r, "G(f)"] & G(x') \end{tikzcd}$$

For the transformation $\pi 1$, this square becomes

$$\begin{tikzcd} X \times Y \arrow[r, "f \times g"] \arrow[d, "\pi 1_{XY}"'] & X' \times Y' \arrow[d, "\pi 1_{X'Y'}"] \\ X \arrow[r, "f"] & X' \end{tikzcd}$$

We establish its commutativity by observing that for each pair (x, y), applying $f \times g$ to yield $(f(x), g(y))$ and then projecting out $f(x)$ has the same effect as projecting out x and applying f to the result. Hence $\pi 1$ is natural.

This form of argument is called a *diagram chase.* We picked an element (x, y) and "chased" it around the possible paths of the diagram to show that always led to the same element at the end.

For the transformation $\alpha\times$, the corresponding square is

$$\begin{tikzcd}
X \times (Y \times Z) \arrow[r, "f \times (g \times h)"] \arrow[d, "\alpha\times_{XYZ}"'] & X' \times (Y' \times Z') \arrow[d, "\alpha\times_{X'Y'Z'}"'] \\
(X \times Y) \times Z \arrow[r, "(f \times g) \times h"] & (X' \times Y') \times Z'.
\end{tikzcd}$$

Here we chase the element $(x,(y,z)$ of $X \times (Y \times Z)$ around the diagram. Over the top we pass via $(f(x),(g(y),h(z)))$ to arrive at $((f(x),g(y)),h(z))$. Down the left we pass via $((x,y),z)$ to arrive again at $((f(x),g(y)),h(z))$. Hence $\alpha\times$ is natural.

When $F, G : C \to D$ are monotone maps between ordered sets, every transformation from F to G is automatically natural, since all diagrams in an ordered set commute.

A *natural isomorphism* is a natural transformation each of whose morphisms is an isomorphism of D. For example any equational law in a poset whose two terms are expressible as functors is a natural isomorphism. A more interesting example in **Set** is given by associativity of $\times$ in **Set**: each $\alpha\times_{XYZ}$ is a bijection, that is, an isomorphism in **Set**, making $\alpha\times$ a natural isomorphism.

Exercises. 1. Show that $\alpha+$ is a natural isomorphism.

2. Formulate commutativity laws for cartesian product and disjoint union as natural isomorphisms.

We will encounter many more examples of natural transformations when we come to adjunctions.

8.2 Composition of Natural Transformations

Any two natural transformations $\sigma : F \to G$ and $\tau : G \to H$ between three parallel functors $F, G, H : C \to D$ compose "vertically" in the obvious way, as suggested by the following diagram.

$$\begin{tikzcd}
F(x) \arrow[r, "F(f)"] \arrow[d, "\sigma_x"'] & F(x') \arrow[d, "\sigma_{x'}"'] \\
G(x) \arrow[r, "G(f)"] \arrow[d, "\tau_x"'] & G(x') \arrow[d, "\tau_{x'}"'] \\
H(x) \arrow[r, "H(f)"] & H(x')
\end{tikzcd}$$

That the outer square of this diagram commutes follows from the commutativity of the two inside squares. This composition can be seen to be associative.

Moreover, for each functor $F : C \to D$ there is an obvious choice of identity natural transformation on F, defined by $\tau_x = 1_{Fx}$.

Hence the functors from C to D constitute the objects of a category, denoted D^C, whose morphisms are the natural transformations between such functors. That is, we have equipped the homset $\hom(C, D)$ with morphisms to make it a "homcategory."

For example the two families $\pi 1_{XY} : X \times Y \to X$ and $\pi 2_{XY} : X \times Y \to Y$ of projections compose vertically with the respective families $i1_{XY} : X \to X+Y$ and $i2_{XY} : Y \to X+Y$ of injections to yield two distinct natural transformations from $X \times Y$ to $X+Y$.

Every natural isomorphism $\tau : F \cong G$ has an inverse $\tau^{-1} : G \cong F$ obtained by inverting each of its morphisms. The vertical composition $\tau^{-1}\tau : F \to F$ can then be seen to consist solely of identity morphisms, making it the identity natural transformation on F. Hence natural isomorphisms are isomorphisms of functors, and indeed could have been defined as such. That we did not is because we did not then have the notion of vertical composition.

Besides this vertical composition there is also a "horizontal" composition, which is slightly more complicated. Let $F, G : A \to B$ and $F', G' : B \to C$ be two pairs of parallel functors. Given natural transformation $\tau : F \to G$ and $\tau' : F' \to G'$, the naturality of τ' requires the commutativity of the square

$$\begin{tikzcd}
F'Fx \arrow[r, "F'\tau_x"] \arrow[d, "\tau'_{Fx}"'] & F'Gx \arrow[d, "\tau'_{Gx}"'] \\
G'Fx \arrow[r, "G'\tau_x"] & G'Gx
\end{tikzcd}$$

That is, τ and τ' can be applied in either order, i.e. independently.

We then define the horizontal composition $\tau' \circ \tau$ of τ' and τ to be the diagonal of this square, $\tau_{Gx}F'(f) = G'(f)\tau_{Fx}$.

For an example of horizontal composition consider the two functors $(W \times X) \times (Y \times Z)$ and $(Z \times Y) \times (X \times W)$ from $\mathbf{Set}^4$ to $\mathbf{Set}$, and for each object (W, X, Y, Z) of $\mathbf{Set}^2$ let $\delta_{WXYZ} : (W \times X) \times (Y \times Z) \to (Z \times Y) \times (X \times W)$ be the function (morphism of $\mathbf{Set}$) defined by $\delta_{WXYZ}((w,x),(y,z)) = ((z,y),(x,w))$. It can be seen that this defines a natural transformation δ between these two functors.

We may decompose δ horizontally as follows. Let $(W \times X, Y \times Z)$ and $(X \times W, Z \times Y)$ be two functors from $\mathbf{Set}^4$ to $\mathbf{Set}^2$, and define κ_{WXYZ} to be the pair of functions (c_{WX}, c_{YZ}), defining a natural transformation κ between these two functors. Then it may be seen that $\delta = c \circ \kappa$, where $c_{UV} : U \times V \to V \times U$ is commutativity of Cartesian product as defined above.

Exercises. 1. Show that the horizontal composition of two such natural transformations is a natural transformation.

2. Given parallel functors $F, G, H : A \to B$ and $F', G', H' : B \to C$, prove the following *interchange law* for four natural transformations $\sigma : F \to G$, $\tau : G \to H$ $\sigma' : F' \to G'$, $\tau' : G' \to H'$.

$$(\tau'\sigma') \circ (\tau\sigma) = (\tau' \circ \tau)(\sigma' \circ \sigma)$$

8.3 Functor Categories

We gave as algebraic motivation for natural transformations the notion of the category C^D with objects all functors $F : D \to C$ and morphisms all natural transformations $\tau : F \to G$ between such functors. We call this the *functor category* C^D. In this section we look at some commonly encountered functor categories where one of C or D is small.

We begin with $D = \mathbf{0}$, the empty category. Now $\mathbf{0}$ is initial in $\mathbf{Cat}$, having exactly one functor $F_C : \mathbf{0} \to C$ to each category C. Hence $C^{\mathbf{0}}$ has one object. A natural transformation $\tau : F_C \to F_C$ is a function from $\mathrm{ob}(\mathbf{0}) = \emptyset$ to the set of morphisms of C. Again there is only one such function, whence $C^{\mathbf{0}}$ has only one morphism, the identity morphism on its one object. Hence $C^{\mathbf{0}}$ is isomorphic to the category $\mathbf{1}$.

The functor from $\mathbf{0}$ to C is distinct from the functor from $\mathbf{0}$ to some other category B, in that they have distinct targets. Hence $C^{\mathbf{0}}$ is not equal to $B^{\mathbf{0}}$, and by implication not equal to $\mathbf{1}$. One might *define* $\mathbf{1}$ to be $C^{\mathbf{0}}$ for a particular C, say $\mathbf{0}$, in which case we would have a single exception to this implication at that C.

Now let us consider $\mathbf{1}$, having object 1. Let $F : C^{\mathbf{1}} \to C$ be the functor defined as application of its argument to 1. It takes each functor $G : \mathbf{1} \to C$ to the object $G(1)$ of C, and each natural transformation $\tau : G \to G'$ to the morphism $\tau_1 : G(1) \to G'(1)$ of C, preserving source, target, composition, and identities as required of a functor. Moreover it hits every object and morphism of C. Hence it is an isomorphism and $C^{\mathbf{1}}$ is isomorphic to C.

Let 2 denote the discrete two-object category. The functor category C^2 has for its objects pairs of objects of C, and for its morphisms pairs of morphisms of C. This describes the category $C \times C$. We have already been using the notation C^2 as a convenient abbreviation for $C \times C$ without promising any connection with the notion of functor category. Fortunately then, these two usages of C^2 coincide.

More generally, for any discrete category D, C^D is the D-th power of C, that is, the product of the constant family $(C_d)_{d \in \mathrm{ob}(D)}$ with index set the objects of D and all members $C_d = C$.

Such powers are easily described without reference to functor categories. We took advantage of this early on in describing such operations as product in a category C as a functor $\times : C^2 \to C$. Other operations such as composition and pullback however were not so conveniently described. For example we held the fort by describing composition as a partial binary operation on morphisms, with domain messily restricted by the condition that the source of one argument had to match up with the target of the other. Functor categories permit a tidier description of such domains. For example the domain of composition in C becomes the functor category $C^{\mathbf{3}}$ of triangles in C, while its codomain is $C^{\mathbf{2}}$, the category of morphisms in C. Similarly pullback in C is a functor $C^{>} \to C$ mapping wedge-shaped diagrams in C to objects of C.

we can be more explicit about the names of the domains of such operations, allowing us to describe them more insightfully as functors between suitable functor categories. The messiness of partiality then disappears!

Let us now move towards more general powers. We start with the basic nondiscrete category $\mathbf{2}$, the two-object chain. The functor category $C^{\mathbf{2}}$ has for its objects morphisms $f : w \to x$ of C and for its morphisms commuting squares $\mathcal{S}$ of C of the form

$$\begin{tikzcd} w \arrow[r, "f"] \arrow[d, "h"'] & x \arrow[d, "k"] \\ y \arrow[r, "g"'] & z \end{tikzcd}$$

These squares $\mathcal{S}$ have their source and target at their top and bottom respectively, compose vertically in the obvious way, and are identities just when their vertical sides are identities.

Given functors $F : A \to C$, $G : B \to C$, the *comma category* $(F{\downarrow}G)$ is the subcategory of $A \times C^{\mathbf{2}} \times B$ whose morphisms $(f, \mathcal{S}, g)$ are such that the left and right sides of the square $\mathcal{S}$ in C are respectively $F(f)$ and $G(g)$. $(F{\downarrow}G)$ amounts to the variable homset $\hom(F(a), G(b))$ in C, with a, b ranging over $A \times B$. When A and B (and hence $A \times B$) are discrete, i.e. just sets, so is $(F{\downarrow}G)$. In particular when A and B are both $\mathbf{1}$ then $(F{\downarrow}G)$ amounts to the homset $\hom(F(1), G(1))$, the constant homset.

When F is the identity functor $1_C : C \to C$ (so $A = C$) we write it as C (the homogeneous view). In particular $(C{\downarrow}C) = C^{\mathbf{2}}$. When $A = \mathbf{1}$, F is the name $\lceil F(1) \rceil$; we abbreviate $(\lceil F(1) \rceil{\downarrow}G)$ as $(F(1){\downarrow}G)$ when there is no ambiguity. All of the above applies equally to G. For any object x of C we refer to $(C{\downarrow}x)$ (properly, $(C{\downarrow}\lceil x \rceil)$) as the category of *objects over x*. $(C{\downarrow}x)$ is also called the *slice category C/x* of objects over x. Dually $(x{\downarrow}C)$ is called the category of *objects under x*.

The objects and morphisms of C/x amount to those of C suffixed with $\xrightarrow{f} x$ for all possible such f's. Each object y becomes various objects $y \xrightarrow{f} x$ of C/x and each morphism $y \xrightarrow{g} y'$ of C becomes various morphisms $y \xrightarrow{g} y' \xrightarrow{f} x$ of C/x. Note that the latter are not just morphisms $y \xrightarrow{fg} x$ of C since they include the information as to how fg decomposes into f and g, i.e. these morphisms of C/x are triangles in C terminating in x.

The category $\mathbf{3}$ is the three-object chain $1 \to 2 \to 3$, a triangle. The functor category $C^{\mathbf{3}}$ has for its objects all triangles $x \xrightarrow{f} y \xrightarrow{g} z$ in C and for its morphisms all commuting diagrams

$$\begin{tikzcd}
x \arrow[r, "f"] \arrow[d, "i"'] & y \arrow[r, "g"] \arrow[d, "j"'] & z \arrow[d, "k"'] \\
x' \arrow[r, "f'"] & y' \arrow[r, "g'"] & z'
\end{tikzcd}$$

whose categorical structure parallels that of the morphisms of $C^{\mathbf{2}}$.

8.4 2-categories

We just saw that the homsets of the category **Cat** of all small categories could be equipped with morphisms, in this case natural transformations between functors, satisfying an interchange principle. A category with homsets so equipped is called a *2-category*, and the morphisms between its morphisms are called *2-cells*. By back formation we could call morphisms 1-cells and objects 0-cells, though this is not usually done.

The homogeneous objects-are-morphisms view of categories yields the most straightforward definition of a 2-category. A *2-category* is a set $(X, s_1, t_1, c_1, s_2, t_2, c_2)$ such that

(i) for each $i = 1, 2$, (X, s_i, t_i, c_i) is a category (presented homogeneously);

(ii) $s_1 s_2 = s_1 t_2 = s_2 s_1 = t_2 s_1 = s_1$, $t_1 s_2 = t_1 t_2 = s_2 t_1 = t_2 t_1 = t_1$;

(iii) $s_1(x) = t_1(x')$ implies $s_2(c_1(x, x')) = c_1(s_2(x), s_2(x'))$ and $t_2(c_1(x, x')) = c_1(t_2(x), t_2(x'))$;

(iv) (Interchange) $s_2(x) = t_2(y)$, $s_2(x') = t_2(y')$, $s_1(x) = t_1(x')$ imply $c_1(c_2(x, y), c_2(x', y')) = c_2(c_1(x, x'), c_1(y, y'))$.

The composition c_1 is horizontal composition and c_2 is vertical composition. The set of morphisms or 1-cells is given by either of $s_2(X)$ or $t_2(X)$. Likewise the objects or 0-cells are given by $s_1(X)$ or $t_1(X)$. Just as the morphisms are a subset of the 2-cells, so are the objects a subset not only of the 2-cells but of the morphisms; in fact $s_1(X) = s_1(s_2(X))$, etc.

The category **Cat** provides the basic example of a 2-category. The objects are categories, the morphisms are functors, and the 2-cells are natural transformations between functors.

For a quite different example take the category of all rectangular matrices over say the integers, with multiplication as its vertical composition and "juxtaposition" as its horizontal composition. The juxtaposition $M \circ N$ of an $m \times m'$ matrix M with an $n \times n'$ matrix N is the $(m + n) \times (m' + n')$ matrix obtained by positioning the two matrices, displayed as usual, with the lower right corner of M touching the upper left corner of N, padded out with zeros in the upper right and lower left to form a rectangular matrix. The interchange law may now be verified, and amounts to a special case of the rule for block multiplication.

An *n-category* is a structure $(X, s_1, t_1, c_1, \ldots, s_n, t_n, c_n)$ such that $(X, s_i, t_i, c_i, s_j, t_j, c_j)$ is a 2-category for all $1 \leq i < j \leq n$.

An *n-functor* $F : (X, s_1, t_1, c_1, \ldots, s_n, t_n, c_n) \to (X', s'_1, t'_1, c'_1, \ldots, s'_n, t'_n, c'_n)$ is a function $F : X \to X'$ such that for $1 \le i \le n$, $F : (X, s_i, t_i, c_i) \to (X', s'_i, t'_i, c'_i)$ is a functor. That is, an n-functor is just a function between n-categories that is a functor at each level, involving no interaction between levels.

8.5 Adjunctions

An isomorphism pair is a pair of morphisms whose composition, in either order, is an identity. For example $f : a \to b$ and $g : b \to a$ form an isomorphism pair just when both gf and fg are identities, namely 1_a and 1_b respectively. That is, the composition of any two consecutive morphisms in

$$a \xrightarrow{f} b \xrightarrow{g} a \xrightarrow{f} b \xrightarrow{g} a$$

is an identity.
An adjunction between categories X, A is a pair $\eta : 1_X \to UF$, $\varepsilon : FU \to 1_A$ of natural transformations

$$\begin{tikzcd}
X \arrow[rr, equal] \arrow[dr, "F"'] & \arrow[d, phantom, "\Downarrow\eta"] & X \\
& A \arrow[ur, "U"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& X \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow\varepsilon"] & \\
A \arrow[ur, "U"] \arrow[rr, equal] & {} & A
\end{tikzcd}$$

whose composition, in either order, is an identity natural transformation. The two compositions are

$$\begin{tikzcd}
X \arrow[rr, equal] \arrow[dr, "F"'] & \arrow[d, phantom, "\Downarrow\eta"] & X \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow\varepsilon"] & \\
& A \arrow[ur, "U"'] \arrow[rr, equal] & {} & A
\end{tikzcd}
\qquad
\begin{tikzcd}
& X \arrow[rr, equal] \arrow[dr, "F"'] \arrow[d, phantom, "\Downarrow\varepsilon"] & \arrow[d, phantom, "\Downarrow\eta"] & X \\
A \arrow[ur, "U"] \arrow[rr, equal] & {} & A \arrow[ur, "U"'] &
\end{tikzcd}$$

These compositions are made using the following vertical compositions:

$$
\begin{tikzcd}
X \arrow[rr, equal] \arrow[dr, "F"'] & {} \arrow[d, phantom, "\Downarrow\eta"] & X \arrow[dr, "F"] & \\
X \arrow[dr, "F"'] & A \arrow[ur, "U"'] & X \arrow[dr, "F"] & A \\
& A \arrow[ur, "U"] \arrow[rr, equal] & {} \arrow[u, phantom, "\Downarrow\varepsilon"] & A
\end{tikzcd}
\qquad
\begin{tikzcd}
& X \arrow[rr, equal] \arrow[dr, "F"'] & {} \arrow[d, phantom, "\Downarrow\eta"] & X \\
A \arrow[ur, "U"] & X \arrow[dr, "F"] & A \arrow[ur, "U"'] & X \\
A \arrow[ur, "U"'] \arrow[rr, equal] & {} \arrow[u, phantom, "\Downarrow\varepsilon"] & A \arrow[ur, "U"] &
\end{tikzcd}
$$

and the four constituents of these vertical compositions are made with horizontal compositions, namely $F\eta$, εF, ηU, and $U\varepsilon$.

Just as we could form a long chain $gfgfgf$ of morphisms any consecutive two of which compose to form an identity, so can we form a long chain of alternating natural transformations any two consecutive members of which paste to form an identity, either 1_{1_X} or 1_{1_A}.

$$
\begin{tikzcd}
X \arrow[rr, equal] \arrow[dr, "F"'] & {} \arrow[d, phantom, "\Downarrow\eta"] & X \arrow[rr, equal] \arrow[dr, "F"] & {} \arrow[d, phantom, "\Downarrow\eta"] & X \arrow[dr, "F"] & \\
& A \arrow[ur, "U"] \arrow[rr, equal] & {} \arrow[u, phantom, "\Downarrow\varepsilon"] & A \arrow[ur, "U"] \arrow[rr, equal] & {} \arrow[u, phantom, "\Downarrow\varepsilon"] & A
\end{tikzcd}
$$

An isomorphism of categories can then be seen to be the special case of an adjunction where η and ε are identity natural transformations. For to say that $\eta : 1_X \to UF$ is an identity is to say that $UF = 1_X$.

We call F and U respectively the *left adjoint* and *right adjoint* of the adjunction, and η and ε respectively the *unit* and *counit*.

In a partial order with all finite nonempty joins, we may describe $x \vee y$ as a monotone operation satisfying $x \leq x \vee y$, $y \leq x \vee y$, and $x \vee x \leq x$. This description may be formulated as the adjunction $(\vee, \Delta, \eta, \varepsilon)$ having left adjoint $\vee : P^2 \to P$, right adjoint $\Delta : P \to P^2$ the diagonal functor defined as $\Delta(x) = (x, x)$, unit $\eta : 1_{P^2} \to \Delta\vee$ the law $(x, y) \leq^2 (x \vee y, x \vee y)$, namely the first two inequations defining $x \vee y$, and counit $\varepsilon : \vee\Delta \to 1_P$ the law $x \vee x \leq x$, namely the third inequation.

Now let us verify the two pasting conditions. The first pasting, $(\varepsilon \circ 1_F)(1_F \circ \eta)$, here $x \vee y \leq (x \vee y) \vee (x \vee y) \leq x \vee y$, must be $1_{x \vee y}$ since ordered sets only have one morphism between any two objects. The same argument applies to the other pasting, here $(x, x) \leq (x \vee x, x \vee x) \leq (x, x)$.

The adjunction contains all the information present in our original description, since the information that $\vee$ is monotonic is implicit in the assumption that $\vee$ is a functor, and the three inequations are given by the unit and counit.

This description is actually a definition of $x \vee y$, up to isomorphism in the case of a preorder and "on the nose" in the case of a partial order. To see this, let z and z' be two values for $x \vee y$ satisfying the three conditions, for any given x, y. Hence we have $x \leq z$, $y \leq z$, so by monotonicity of $\vee$, $x \vee y \leq z \vee z \leq z$, whence $z' \leq z$. Similarly $z \leq z'$, so z and z' must be isomorphic in a preorder and identical in a partial order.

An equivalent definition of an adjunction is a natural isomorphism φ between the functors $\hom(Fx, a)$ and $\hom(x, Ua)$. These are parallel functors in that they both map $X^{\mathrm{op}} \times A$ to **Set**. That φ is a natural isomorphism means that every morphism $\varphi_{x,a} : \hom(Fx, a) \to \hom(x, Ua)$ is a bijection of homsets, and that for every morphism of $X^{\mathrm{op}} \times A$, that is, for every pair $(h, k) : (x, a) \to (x', a')$ of morphisms $h : x' \to x$, $k : a \to a'$, the following diagram commutes.

$$\begin{tikzcd}
\hom(Fx, a) \arrow[r, "\hom(Fh, k)"] \arrow[d, "\varphi_{xa}"'] & \hom(Fx', a') \arrow[d, "\varphi_{x'a'}"'] \\
\hom(x, Ua) \arrow[r, "\hom(h, Uk)"] & \hom(x', Ua')
\end{tikzcd}$$

For another example consider the set $\mathbf{Z}$ of integers and the set $\mathbf{R}$ of reals, each with their usual order, hence each a category. The function $fl : \mathbf{Z} \to \mathbf{R}$ takes the integer a to the corresponding real a. It has a left adjoint $\lceil x \rceil$ taking the real x to the ceiling of x, that is, the least integer greater or equal to x.

The unit of this adjunction is the law $x \leq fl\lceil x \rceil$, and the counit is $\lceil fl(a) \rceil \leq a$. The natural bijection between $\hom(Fx, a)$ and $\hom(x, Ua)$ is in this case given as exercise 1.2.4-3c of Volume I of Knuth, *The Art of Computer Programming*, which asserts

$$\lceil x \rceil \leq n \text{ if and only if } x \leq n$$

where $\lceil x \rceil$ denotes the least integer greater or equal to x.

The basic example of an adjunction is provided by free algebras. Take X to be the category of sets and A some category of algebras, e.g. the category of all monoids and their monoid homomorphisms. Take $U : A \to X$ to be the functor Ua which returns the underlying set of a given algebra a. Take $F : X \to A$ to be the functor which returns the free algebra Fx generated by a given set x. For each set x take $\eta_x : x \to UFx$ to be the function associating each generator with the corresponding element of (the underlying set of) the free algebra generated by x, and take $\varepsilon_a : FUa \to a$ to be the evaluation function which maps each term of the free algebra on the underlying set of a (thinking of these as values of variables) to the result of evaluating that term at those values.