Chapter 4

Category Theory

4.1 Categories and Functors

4.1.1 Motivation

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.

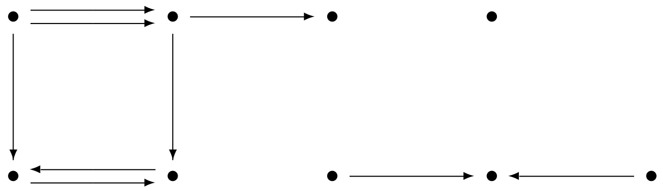
The notion of category that we will define shortly amounts to just such a "partial" monoid with multiple identities. The elements of the partial monoid are called morphisms, and the entities between which the morphisms run are called objects.

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.

4.1.2 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***.

4.1.3 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

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

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.

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 $D_E(E)$.

4.1.4 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. Now 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 $u, v \in V$.

4.1.5 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.

4.1.6 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 $(x_1 \wedge x_2) \vee (x_1 \wedge x_3)$ from 3 to 1, and $(x_1, x_2, \ldots, x_n)$ is is the identity morphism 1_n at n. In such a category certain parallel terms may be identified, e.g. $(x_1 \wedge x_2) \vee (x_1 \wedge x_3)$ and $x_1 \wedge (x_2 \vee x_3)$, which both go from 3 to 1. 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. (To make this a category we must postulate For each n 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 $F(G)$.

4.1.7 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: $\mathbb{Z} \to \mathbb{R}$. 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.

4.1.8 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'.

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 $C \overset{1_C}{\to} C$ takes each object and morphism of C to itself. The composition $C \overset{F}{\to} D \overset{G}{\to} E$, namely $C \overset{GF}{\to} E$, 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 $C \overset{F}{\to} D \overset{G}{\to} C$ 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.

4.1.9 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 $C^\circ = (M, t, s, \check{c})$. Here $\check{c}$ denotes the converse of c, satisfying $\check{c}(g, f) = c(f, g)$. This corresponds to reversing the directions of the morphisms, and hence of composition. Evidently $(C^\circ)^\circ = C$ and $U(C^\circ) = U(C)^\circ$.

But what is a reversed morphism? When C is an abstract category, namely a graph with a composition law, no conceptual problem arises in reversing an edge: the resulting reversed edge is just another morphism. But when C is a concrete category whose morphisms are functions between sets, it is natural to ask what function is obtained by reversing a function.

For the special case where the function is a bijection from X to Y, its reverse can be taken to be its inverse, a function from Y to X. But not all functions are bijections. Consider the morphisms in $\mathbf{Set}^\circ$ from set X to the singleton $\{0\}$. These are the functions from $\{0\}$ to X, one per element of X. But there can only be one function to a singleton, so these morphisms cannot be distinct functions from X to $\{0\}$. It would seem that we have to settle for treating the reverse of a concrete morphism as just an abstract morphism.

One method of making reversed functions more concrete is to recall that a function $f : X \to Y$ is a special case of a binary relation from X to Y, whose converse is a binary relation from Y to X. Then if C is any category of sets and binary relations, C° has the same objects as C while its morphisms are the converses of the morphisms of C. $\mathbf{Set}^\circ$ can then be understand as an instance of this construction, having morphisms that are binary relations but not in general functions.

Another "concretization" of a reversed function is its inverse image function $f^{-1} : 2^Y \to 2^X$, defined as $f(Y') = \{x \in X | f(x) \in Y\}$. Here the reverse of a function is a function. To interpret f^{-1} as a function between sets we must also replace each set X in C by its power set 2^X.

Unlike the former method, the latter generalizes to any locally small category C (one with small homsets). Choose an object z of C. To each object x of C associate the set $C(x, z)$. To each morphism $f : x \to y$ of C associate the function $C(f, z) : C(y, z) \to C(x, z)$ defined as $C(f, z)(g) = gf$ where $g : y \to z$ (and hence $gf : x \to z$). It is now readily verified that these associations define a functor $C(-, z)$ from C° to **Set**. Such a functor constitutes a representation of the objects of C as sets and the morphisms of C as functions. The representation is *faithful* when distinct morphisms are represented in this way as distinct functions. This is the same thing as saying that the functor $C(-, z)$ is faithful. For example if $C = \mathbf{Set}$ then $C(-, z)$ is faithful if and only if the set z has at least two elements. Taking z to be $\{0, 1\}$ then gives the representation of the preceding paragraph.

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*** $f : x \to y$ is a morphism for which there exists a morphism $g : y \to x$ such that $gf = 1_x$ and $fg = 1_y$. Two objects with an isomorphism 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.

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 **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.

4.1.10 Exercises

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

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

3. Show that 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,

4. 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.)

5. Define "subcategory" from the heterogeneous viewpoint.

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

7. Show that the category **Grp** of all groups and group homomorphisms is a full subcategory of **Mon**.

8. Show that **Set**$^\circ$ is not isomorphic to **Set**, but is isomorphic to a subcategory of **Set**.

9. Show that every isomorphism is a bimorphism.

10. Show that in **Set** every bimorphism is an isomorphism.

11. (i) Show that in a finite monoid, viewed as a category, every monic is an isomorphism. (ii) Exhibit a monoid containing a bimorphism that is not an isomorphism. (Hint: consider functions on the set of integers that operate by adding a constant.)

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

4.2 Limits

4.2.1 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 **Cat** to **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.

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.

4.2.2 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, $X_1 \overset{p_1}{\leftarrow} X_1{\times}X_2 \overset{p_2}{\rightarrow} X_2$) 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 1 *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 $x_1 \overset{p_1}{\leftarrow} z \overset{p_2}{\rightarrow} x_2$ and $x_1 \overset{p'_1}{\leftarrow} z' \overset{p'_2}{\rightarrow} x_2$ 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 $p_1p'p = p'_1p = p_1$ and $p_2p'p = p'_2p' = p_2$. But we also have $p_11_z = p_1$ and $p_21_z = p_2$. 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.

Proposition 2 *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_1p_1, q_2p_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_1g_{12}, q_2g_{12}), and let $g : y \to z$ represent (g_{12}, g_3), as (p_1g, p_2g). Hence g represents $(g_i : y \to x_i)_{i=1,2,3}$, as (q_1p_1g, q_2p_1g, p_2g). Now for any other such representative $g' : y \to z$, p_1g' 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_1p_1, q_2p_1, p_2) is universal. ∎

Corollary 1 *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 $x_1 \overset{p_1}{\to} z \overset{p_2}{\leftarrow} x_2$ is called a *coproduct* of x_1 and x_2 when for every diagram $x_1 \overset{p'_1}{\to} z' \overset{p'_2}{\leftarrow} x_2$ 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.*

4.2.3 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.

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 $p_1g = g_1$ and $p_2g = g_2$.

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 $p_3 = f_1p_1 = f_2p_2$.

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***.

4.2.4 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 $f_eg_{se} = g_{te}$ 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.

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 18 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 3 *A category with all equalizers, and all products of up to α objects, α any ordinal, has all limits of diagrams with up to α edges.*

Proof:

and let $D = (a : V \to O(C), f : E \to M(C))$ be

Let D be a diagram in category C on graph $G = (V, E, s, t)$ $(s, t : E \to V)$ consisting of functions $a : V \to O(C)$, $f : E \to M(C)$. The first function labels each vertex u of G with object a_u of C and the second labels each edge e of G with morphism f_e of C.

Obtain from D the following two discrete diagrams. Take D' to be $(a : V \to O(C), \emptyset)$, the diagram resulting from deleting the edges of D. (So D' is a diagram in C on the discrete graph $(V, \emptyset, \emptyset, \emptyset)$.) Take $D'' = (a \circ t : E \to O(C), \emptyset)$, a diagram in C on graph discrete graph $(E, \emptyset, \emptyset, \emptyset)$ consisting of the codomains appearing in D, with each edge e of G giving rise to codomain $a_{t(e)}$.

Let p, q be products of D', D'' respectively, with associated projections $p_u : p \to a_u$, $u \in V$, $q_e : q \to a_{t(e)}$, $e \in E$, as morphisms of C.

Now construct two cones both from p to D''. For the first cone, take its e-th projection to be the $t(e)$-th projection of the product of D', namely $p_{t(e)} : p \to a_{t(e)}$. Since q is the universal cone to D'', this cone to D'' determines a unique map $k : p \to q$ commuting with the projections to D'', which we can view as one function representing the family $\langle p_{t(e)} \rangle_{u<U}$.

For the second cone, take its e-th projection to be $f_e p_{s(e)} : p \to a_{t(e)}$. This cone to D'' similarly determines a unique map $h : p \to q$ commuting with the projections to D'', which we can view as one function representing the family $\langle f_e p_{s(e)} \rangle_{e \in E}$.

We now claim that the equalizer $e : d \to p$ of h and k, with projections $p_i e$, is a limit of D.

Let r be any cone to D. We show that there is a unique map from r to e commuting with the projections to D. Now r is also a cone to D', whence there exists a unique map $s : r \to p$ commuting with the projections to D', i.e.

$$p_i s = r_i. \qquad (1)$$

Now consider the cone from r to D'' whose projections are $r_{t(e)} : r \to a_{t(e)}$. Setting i to $t(e)$ in (1) gives $p_{t(e)} s = r_{t(e)}$ for all $u < U$. But $q_u k = p_{t(e)}$ since k commutes with those projections to D'', so $q_u k s = p_{t(e)} s = r_{t(e)}$, that is, ks commutes with *those* projections to D'', and hence is the unique such map from r to q Setting i to $s(e)$ in (1) gives $p_{s(e)} s = r_{s(e)}$, whence $f_u p_{s(e)} s = f_u r_{s(e)} = r_{t(e)}$, for all $u < U$. But $f_u p_{s(e)} = q_u h$, so $q_u h s = r_{t(e)}$, that is hs commutes with those projections to D'' and hence is the unique such map from r to q. But ks also enjoys that distinction so $hs = ks$. This makes $s : r \to p$ a cone to the diagram $h, k : p \to q$, whence there exists a unique map from r to e commuting with the projections to D' and hence D. ∎

4.2.5 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° are just colimits in C and vice versa, doubling at one stroke the number of categories both whose limits and colimits we understand.

Proposition 4 *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*** or ***zero*** 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 **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 $f(*_X) = *_Y$.

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 $\leq_Y$.

4.2.6 Exercises

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

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

3. (i) Show that a category with binary products has n-ary products for all finite $n \geq 1$. (ii) Show that n-ary products are unique up to isomorphism.

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

5. 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**.

6. 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. Show that equalizers are unique up to isomorphism.

8. 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**.

9. Show that pullbacks and pushouts are unique up to isomorphism.

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

11. 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.

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

13. 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.

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

15. Define colimit.

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

17. Explain why equalizers and pullbacks are limits.

18. 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.

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

20. 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?

21. Show that products, coproducts, and equalizers in the category **Pos** of partially ordered sets and their monotone maps have as their underlying sets those fo the corresponding constructs in **Set**. Give an example showing where this fails for coequalizers.

22. Show that any variety has all limits.

23. In the category **AbMon** of abelian monoids, show that the product of a pair of abelian monoids is isomorphic to their coproduct.

4.3 Natural Transformations

4.3.1 Basics of 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 **Set**.

The analogous notion for **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 **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 **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 $\mathbf{Set}$, $F : \mathbf{Set}^2 \to \mathbf{Set}$ 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 $\mathbf{Set}$, C is $\mathbf{Set}^3$, D is $\mathbf{Set}$, 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 × in **Set**: each $\alpha\times_{XYZ}$ is a bijection, that is, an isomorphism in **Set**, making $\alpha\times$ a natural isomorphism.

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

4.3.2 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 $\pi1_{XY} : X \times Y \to X$ and $\pi2_{XY} : X \times Y \to Y$ 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.

4.3.3 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 $\text{ob}(\mathbf{0}) = \emptyset$ 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 $(C_d)_{d \in \text{ob}(D)}$ 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 $\ulcorner F(1) \urcorner$; we abbreviate $(\ulcorner F(1) \urcorner \downarrow G)$ 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, $(C \downarrow \ulcorner x \urcorner)$) 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 $C^{\mathbf{2}}$.

4.3.4 Algebraic Theories

In this section we give a formal definition of "algebraic theory," explore a few of its drier properties, then ease up a bit while we relate the concept to the examples and to the intuitive notion of "theory," and finally consider some additional aspects of theories.

Definition 1 *An algebraic theory T is a skeletal category with all finite products.*

(Recall that a category is skeletal just when every isomorphism is an automorphism.)

In particular T must have a final object as the empty product, whence T is nonempty.

Example 1 *(***Fincard**^{op} *and* **1** *are algebraic theories, where* **Fincard***, for finite cardinals, denotes the full subcategory of* **Set** *consisting of one set of each cardinality. These are the trivial and inconsistent theories respectively. A consistent theory is any algebraic theory other than the inconsistent one.*

Definition 2 *A theory morphism is a functor between theories that preserves finite products. The category of all algebraic theories and their theory morphisms is denoted* **Th***.*

Example 2 *The trivial and inconsistent theories are respectively the initial and final objects of* **Th***.*

So far, so unmotivated. While **1** is obviously boring, **Fincard**^{op} looks like it might be a bit more glamorous, despite its unprepossessing nickname "trivial."

The idea is that **Fincard**^{op} consists of formal functions between tuples. Think of a formal function not as an actual function but as a computer program for computing that function. For example the morphism $f : X^3 \to X^2$ might be the formal function $f : (x_0, x_1, x_2) \mapsto (x_2, x_0)$, or it might be $g : (x_0, x_1, x_2) \mapsto (x_1, x_1)$. In LISP we would write

```
(DEFUN F (X0 X1 X2) (LIST X2 X0))
```

and

```
(DEFUN G (X0 X1 X2) (LIST X1 X1)).
```

If X were "instantiated" say by the set ω of natural numbers then such functions would correspondingly be instantiated as functions between powers of ω, with the first example mapping $(3, 99, 2)$ to $(2, 3)$, $(5, 11, 11)$ to $(11, 5)$, etc., and the second mapping them to $(99, 99)$, $(11, 11)$, etc. Each such formal function from X^m to X^n can be seen to be defined by an actual function from n (as an ordinal, i.e. initial segment of ω) to m. The above examples can then be seen to be defined by functions from 2 to 3: in the first example it maps 0 to 2 and 1 to 0, while in the second it maps both 0 and 1 to 1. That things got turned around here explains the *op* in **Fincard**^{op}.

The n formal functions from X^n to X are now easily recognized as the n projections of X^n onto X. But of course; X^n is the n-th power of X, as we said at the beginning. To economize on subscripts, let us denote the projections (and their images when no confusion results) by $x, y, z, x', y', \ldots$ in that order.

Exercise 1 *The unique theory morphism from the trivial theory to any consistent algebaic theory is bijective on objects and injective on morphisms. (Hence in any algebraic theory the powers of X are either all distinct or all identical.*

So a consistent theory amounts to the trivial theory together with some additional morphisms.

The theory B of Boolean algebras provides an ideal example, in part because it is so easily defined. Simply take B to be the category whose objects are the finite powers of the doubleton $\{0, 1\}$, and whose morphisms are all functions between them. (So this is a countably infinite full subcategory of **Set**.) This category is closed under finite products; indeed the product of i (the name we shall use in this context for the cartesian power 2^i) with j is just $i + j$, with the evident projections.

The trivial theory had just one morphism from 1 to 1. $B(1, 1)$ adds to this the two constant functions and the twist map or Boolean negation. In $B(2, 1)$ the two projections are now joined by 14 more binary operations. In general $B(m, 1)$ has 2^{2^m} m-ary Boolean operations, and more generally still $B(m, n)$ has $(2^n)^{2^m}$ n-tuples of m-ary Boolean operations. (One way to think about subsets of $\mathbf{n}$ is as bit vectors of length n. Thus the Boolean function $x \wedge y$ maps 11 to 1 and the other three bit vectors to 0, while the two projections each pick out one of the two bits.)

What makes this category a theory in the equational sense is that the theory embodies the equations of Boolean algebra. It does this by having different compositions being equal, which is just a thin disguise for different expressions being equal.

The examples with polynomials are somewhat different in character if we regard polynomials as syntactic objects. But this presents no problem: just define composition to be formal substitution; the composition of the pair of polynomials $(x + y, x - y)$ (a morphism from 2 to 2) with xy (a morphism from 2 to 1) is the polynomial $(x + y)(x - y)$. However, we can also construct this category in a more semantic way like our Boolean algebra example: take the objects to be direct powers of Z (the ring of integers), or for that matter of R (the ring of reals), and the morphisms to be all integer-coefficient polynomials with either integer or real values of the variables respectively. This category is isomorphic to the syntactically defined one. Note that we can't take powers of just any old ring; if we take the objects to be direct powers of the ring of integers mod 2 (so $x + y$ turns into $x \oplus y$ – exclusive-or – and xy into $x \wedge y$) – surprise! We get instead the theory of Boolean algebras, that is, a category isomorphic to the one above of functions between power sets, a reminder that Boolean algebras are "the same thing" as Boolean rings.

Now consider the two functions $x \vee y$ and $y \vee x$, morphisms of B, each mapping A^2 to A. These "two" are actually one, a fact expressed by the Boolean identity $x \vee y = y \vee x$. Similarly the composition of $(x, y \vee z) : A^3 \to A^2$ with $x \wedge y : A^2 \to A$ in B is the same function as the composition of $(x \wedge y, x \wedge z) : A^3 \to A^2$ with $x \vee y : A^2 \to A$, corresponding to the identity $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (conjunction distributes over disjunction). But this fact is independent of the choice of A; it depends only on A being a Boolean algebra.

Now let us justify the term "theory." An equational theory such as the theory of commutative rings makes connections such as $(x + y)(x - y) = x^2 - y^2$. We have already seen that the composition of $(x + y, x - y)$ with xy expresses $(x + y)(x - y)$; moreover the composition of (x^2, y^2) with $x - y$ expresses $x^2 - y^2$. But if this is the category of polynomials then these two expressions must be equal, which is to say that the following diagram must commute.

$$\begin{tikzcd}
2 \arrow[r, "{(x+y,x-y)}"] \arrow[d, "{(x^2,y^2)}"'] & 2 \arrow[d, "xy"] \\
2 \arrow[r, "x-y"] & 1
\end{tikzcd}$$

This leads us to the notion of a *presentation* of a theory. A presentation consists of (i) a subset of the morphisms of the theory all having codomain 1, called its *operations*, with arity of each operation being given by its domain; and (ii) a binary relation on the free category generated by those operations together with all their products and all projections, called its *axioms*. The theory so presented is then the quotient of the free category by the least congruence containing the axioms. When the set of operations and the binary relation are both finite the theory is called *finitely axiomatizable.*

For the theory of commutative rings, the usual choice of operations are 0, $x + y$, xy, and $-x$. If we were more economically minded we might pick instead 0, $x - y$ and xy. For either choice there is a finite set of such axioms as $x - x = 0$ or $x - (x - y) = x$.

The inconsistent theory corresponds to an equational theory whose one axiom is $x = y$. It may also be characterized as the equational theory of the empty class, and also of the class containing just one algebra which in turn contains just one element (the unique zeroth direct power of algebras drawn from the empty class), constituting the class of all models of this theory.

An *algebra* is a functor $A : T \to \mathbf{Set}$ from a theory T to **Set** which preserves all finite products. The functor supplies an interpretation of X as a set $A(X)$ and of the morphisms from m to n as n-tuples of functions from $A(X)^n$ to $A(X)$. Given two such algebras, the homomorphisms between them are exactly the natural transformations between them. The naturality property realizes the requirement that homomorphisms commute with the operations of the algebras.

The notion of an algebra may be generalized to other domains by using some category other than **Set**. For example the finite-product-preserving functors from the theory of monoids to **Cat** are the (strictly) monoidal categories, which are categories equipped with an associative binary operation on objects.

4.3.5 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) $s_1s_2 = s_1t_2 = s_2s_1 = t_2s_1 = s_1$, $t_1s_2 = t_1t_2 = s_2t_1 = t_2t_1 = t_1$;

(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 $1 \leq i \leq n$, $F : (X, s_i, t_i, c_i) \to (X', s'_i, t'_i, c'_i)$ 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.

4.4 Adjunctions

An isomorphism together with its inverse constitute a pair of morphisms whose composition, in either order, is an identity. That is, the composition of any two consecutive morphisms in

$$\begin{tikzcd} a \arrow[r, "f"] & b \arrow[r, "g"] & a \arrow[r, "f"] & b \arrow[r, "g"] & a \end{tikzcd}$$

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

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

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

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

These compositions are made using the following vertical compositions:

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

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}.

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

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 $X^\circ \times A$ 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 $X^\circ \times A$, 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.

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

Examples

1. 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 the unit $\eta_x : x \to UFx$ of the adjunction to be the function associating each generator with the corresponding element of (the underlying set of) the free algebra generated by x, and take the counit $\varepsilon_a : FUa \to a$ 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.

2. Consider the set $\mathbb{Z}$ of integers and the set $\mathbb{R}$ of reals, each with their usual order, hence each a category. The function $fl : \mathbb{Z} \to \mathbb{R}$ 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.

4.4.1 Equivalence of three definitions of adjunction

In this section we give three definitions of adjunction and prove their equivalence. Common to these definitions are a pair of categories $\mathcal{X}$ and $\mathcal{A}$ and a pair of functors $F : \mathcal{X} \to \mathcal{A}$ and $U : \mathcal{A} \to \mathcal{X}$. As an instance we may take $\mathcal{X}$ and $\mathcal{A}$ to be the categories of sets and monoids respectively, F the free functor turning a set Σ into the free monoid Σ^* on Σ, and U the forgetful functor taking a monoid to its underlying set.

Definition 3 *An adjunction $F \dashv U$ is a natural isomorphism $\varphi_{xa} : \mathcal{X}(F(x), a) \to \mathcal{A}(x, U(a))$.*

Here $\mathcal{X}(F(x), a)$ and $\mathcal{A}(x, U(a))$ are functors $\mathcal{X}^\circ \times \mathcal{A} \to \mathbf{Set}$. Naturality of a transformation between these functors means that for each morphism $(f : x' \to x, h : a \to a')$ of $\mathcal{X}^\circ \times \mathcal{A}$ the following diagram (in **Set**) commutes.

$$\begin{tikzcd}
\mathcal{A}(F(x),a) \arrow[r, "\mathcal{A}(F(f),h)"] \arrow[d, "\varphi_{xa}"] & \mathcal{A}(F(x'),a') \arrow[d, "\varphi_{x'a'}"] \\
\mathcal{X}(x,U(a)) \arrow[r, "\mathcal{X}(f,U(h))"] & \mathcal{X}(x',U(a'))
\end{tikzcd}$$

Definition 4 *An adjunction $F \dashv U$ is a coordinated isomorphism θ of comma categories $(F \downarrow \mathcal{A})$ and $(\mathcal{X} \downarrow U)$, where "coordinated" means that the isomorphism commutes with the projections from the comma categories to $\mathcal{X}$ and $\mathcal{A}$.*

Such an isomorphism amounts to putting the following pairs of comma category morphisms in bijection.

$$\begin{tikzcd}
F(x) \arrow[r, "F(f)"] \arrow[d, "g"] & F(x') \arrow[d, "g'"] & & x \arrow[r, "f"] \arrow[d, "\varphi(g)"] & x' \arrow[d, "\varphi(g')"] \\
a \arrow[r, "h"] & a' \arrow[r, phantom, "\overset{\theta_{g,g'}}{\mapsto}" pos=0.5, shift left=6ex] & & U(a) \arrow[r, "U(h)"] & U(a')
\end{tikzcd}$$

The significance of coordination is that θ leaves $f : x \to x'$ and $h : a \to a'$ unchanged while transforming the arrows of the objects of the comma categories (vertical in the diagrams) and replacing the F's above with U's below.

Definition 5 *An adjunction between functors $F : \mathcal{X} \to \mathcal{A}$ and $U : \mathcal{A} \to \mathcal{X}$ is a pair of natural transformations $\eta_x : x \to U(F(x))$ and $\epsilon_a : F(U(a)) \to a$ satisfying the* triangle identities

$$\epsilon_{F(x)} F(\eta_x) = 1_{F(x)} \qquad (\triangle 1)$$

$$U(\epsilon_a)\eta_{U(a)} = 1_{U(a)}. \qquad (\triangle 2)$$

This definition presents an adjunction as a higher-dimensional isomorphism, namely as two identity 3-cells, one between the identity 2-cell on F and the 2-cell $\epsilon_{F(x)}F(\eta_x) = 1_{F(x)}$, the other between the identity 2-cell on U and the 2-cell $U(\epsilon_a)\eta_{U(a)}$. In an ordinary isomorphism, one dimension down, η and ϵ are identity 2-cells.

Theorem 2 *Definitions 1-3 are equivalent.*

Proof: In fact the definitions are constructively equivalent, in the sense that every adjunction is fully determined by the data of any one of these three definitions. We prove this by showing, for any given adjunction, how to transform the data of definition i specifying that adjunction into the corresponding data of definition $i + 1$ (mod 3).

(i) Definition 1 → Definition 2.

Given φ as per Definition 1, we take the object part of the functor θ to be simply φ, and take the morphism part to map each morphism of $(F \downarrow \mathcal{A})$ as follows.

$$\begin{tikzcd} F(x) \arrow[r, "F(f)"] \arrow[d, "g"] & F(x') \arrow[d, "g'"] \\ a \arrow[r, "h"] & a' \end{tikzcd} \quad \overset{\theta}{\mapsto} \quad \begin{tikzcd} x \arrow[r, "f"] \arrow[d, "\varphi(g)"] & x' \arrow[d, "\varphi(g')"] \\ U(a) \arrow[r, "U(h)"] & U(a'). \end{tikzcd}$$

It is straightforward to see that this is a functor.

For this last square to be a morphism of $(\mathcal{X} \downarrow U)$ it suffices that it commute, i.e. that $U(h)\varphi_{xa}(g) = \varphi_{x'a'}(g')f$. This follows by diagram chases around two instances of the square in Definition 1 starting with the element g of $\mathcal{A}(F(x), a)$. For the first instance we take $x' = x$ and $f = 1_x$. Going round the top, $\mathcal{A}(F(1_x), h)$ sends g to hg thence to $\varphi_{x,a'}(hg)$. The other way sends g to $\varphi_{xa}(g)$ thence to $U(h)\varphi_{xa}(g)$, so we have $U(h)\varphi_{xa}(g) = \varphi_{xa'}(hg)$. For the second instance we take $a = a'$ and $h = 1_{a'}$, and the corresponding chase yields $\varphi_{xa'}(g'F(f)) = \varphi_{x'a'}(g')f$. But by the commutativity of the second last square, $hg = g'F(f)$, and we now have $U(h)\varphi_{xa}(g) = \varphi_{xa'}(hg) = \varphi_{xa'}(g'F(f)) = \varphi_{x'a'}(g')f$ as desired.

Since φ is a bijection, so is θ. But while it is immediate that φ^{-1} is a bijection, the status of θ^{-1} as itself an isomorphism of categories requires more work. Inspection shows it to be a functor. The argument that it maps morphisms of $(\mathcal{X} \downarrow U)$ to morphisms of $(F \downarrow \mathcal{A})$ is a straightforward dualization of the forward argument, and we are done.

(ii) Definition 2 → Definition 3

We construct the transformations η and ϵ from θ according to

$$\begin{aligned} \eta_x &= \theta(1_{F(x)}) \\ \epsilon_a &= \theta^{-1}(1_{U(a)}). \end{aligned}$$

Here $1_{F(x)}$ is in full the object $(x, 1_{F(x)}, F(x))$ of $(F \downarrow \mathcal{A})$, while $1_{U(a)}$ is similarly the object $(U(a), 1_{U(a)}, a)$ of $(\mathcal{X} \downarrow U)$. (Recall that objects of $(F \downarrow G)$ are triples $(x, g : F(x) \to G(y), y)$.)

We now show that η and ϵ are natural and satisfy the triangle identities.

For naturality of η we have

$$\begin{tikzcd}
F(x) \arrow[r, "F(f)"] \arrow[d, "1_{F(x)}"] & F(x') \arrow[d, "1_{F(x')}"] \\
F(x) \arrow[r, "F(f)"] & F(x')
\end{tikzcd}
\overset{\varphi}{\mapsto}
\begin{tikzcd}
x \arrow[r, "f"] \arrow[d, "\eta_x"] & x' \arrow[d, "\eta_{x'}"] \\
U(F(X)) \arrow[r, "U(F(f))"] & U(F(x'))
\end{tikzcd}$$

For naturality of ϵ we have

$$\begin{tikzcd}
F(U(a)) \arrow[r, "F(U(f))"] \arrow[d, "\epsilon_a"] & F(U(a')) \arrow[d, "\epsilon_{a'}"] \\
a \arrow[r, "f"] & a'
\end{tikzcd}
\overset{\varphi}{\mapsto}
\begin{tikzcd}
U(a) \arrow[r, "U(f)"] \arrow[d, "1_{U(a)}"] & U(a') \arrow[d, "1_{U(a')}"] \\
U(a) \arrow[r, "U(f)"] & U(a')
\end{tikzcd}$$

For $\triangle 1$ we have

$$\begin{tikzcd}
F(U(a)) \arrow[r, "F(1_{U(a)})"] \arrow[d, "1_{F(U(a))}"] & F(U(a)) \arrow[d, "\epsilon_a"] \\
F(U(a)) \arrow[r, "\epsilon_a"] & a
\end{tikzcd}
\overset{\varphi}{\mapsto}
\begin{tikzcd}
U(a) \arrow[r, "1_{U(a)}"] \arrow[d, "\eta_{U(a)}"] & U(a) \arrow[d, "1_{U(a)}"] \\
U(F(U(a))) \arrow[r, "U(\epsilon_a)"] & U(a)
\end{tikzcd}$$

Dually for $\triangle 2$ we have

$$\begin{tikzcd}
F(x) \arrow[r, "F(\eta_x)"] \arrow[d, "1_{F(x)}"] & F(U(F(x))) \arrow[d, "\epsilon_{F(x)}"] \\
F(x) \arrow[r, "1_{F(x)}"] & F(x)
\end{tikzcd}
\overset{\varphi}{\mapsto}
\begin{tikzcd}
x \arrow[r, "\eta_x"] \arrow[d, "\eta_x"] & U(F(x)) \arrow[d, "1_{U(F(x))}"] \\
U(F(x)) \arrow[r, "U(1_{F(x)})"] & U(F(x))
\end{tikzcd}$$

(iii) Definition 3 → Definition 1

From η and ϵ we define transformations $\varphi_{xa} : \mathcal{X}(F(x), a) \to \mathcal{A}(x, U(a))$ and $\psi_{xa} : \mathcal{A}(x, U(a)) \to \mathcal{X}(F(x), a)$ as follows.

$$\begin{aligned} \varphi_{xa}(g : F(x) \to a) &= U(g)\eta_x \\ \psi_{xa}(g' : x \to U(a)) &= \epsilon_a F(g') \end{aligned}$$

We now show that φ and ψ are natural, and mutually inverse, i.e. $\psi = \varphi^{-1}$, whence φ is a natural isomorphism. We argue naturality as follows.

$$\begin{tikzcd} \mathcal{A}(F(x), a) \arrow[r, "\mathcal{A}(F(f), h)"] \arrow[d, "\varphi_{xa}"] & \mathcal{A}(F(x'), a') \arrow[d, "\varphi_{x'a'}"] \\ \mathcal{X}(x, U(a)) \arrow[r, "\mathcal{X}(f, U(h))"] & \mathcal{X}(x', U(a')) \end{tikzcd}$$

Starting with $g : F(x) \to a$, chasing the diagram round the top takes us to $hgF(f)$ and then $U(hgF(f))\eta_{x'}$, which is $U(h)U(g)U(F(f))\eta_{x'}$. Around the other way we pass through $U(g)\eta_x$ to arrive at $U(h)U(g)\eta_x f$.

Now by naturality of η we have

$$\begin{tikzcd} x' \arrow[r, "f"] \arrow[d, "\eta_{x'}"] & x \arrow[d, "\eta_x"] \\ U(F(x')) \arrow[r, "U(F(f))"] & U(F(x)) \end{tikzcd}$$

We now show that $\psi_{xa}\varphi_{xa}$ and $\varphi_{xa}\psi_{xa}$ are identities. Expanding $\psi_{xa}\varphi_{xa}(g)$, we must show that $\epsilon_a F(U(g)\eta_x) = g$. From the naturality of η we have

$$\begin{tikzcd} F(U(F(x))) \arrow[r, "F(U(g))"] \arrow[d, "\epsilon_{F(x)}"] & F(U(a)) \arrow[d, "\epsilon_a"] \\ F(x) \arrow[r, "g"] & a \end{tikzcd}$$

Hence

$$\begin{array}{rcl}\psi(\varphi(g)) & = & \epsilon_a F(U(g)\eta_x) \\ & = & \epsilon_a F(U(g))F(\eta_x) \\ & = & g\epsilon_{F(x)}F(\eta_x) \\ & = & g\end{array}$$

The other inverse is argued dually. ∎

4.4.2 Exercises

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

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

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

4. 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 $\sigma : F \to G$, $\tau : G \to H$ $\sigma' : F' \to G'$, $\tau' : G' \to H'$.

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

4.5 Monads

Let $\mathcal{C}$ be a category. A ***monad on*** $\mathcal{C}$ is a triple (T, η, μ) consisting of a functor $T : \mathcal{C} \to \mathcal{C}$, a natural transformation $\eta : I_{\mathcal{C}} \to T$ called the *unit*, and a natural transformation $\mu : T^2 \to T$ (where T^2 denotes the composition of T with itself) called the *multiplication*, making the following diagrams commute.

$$\begin{tikzcd} T^3 \arrow[r, "\mu\circ T"] \arrow[d, "T\circ\mu"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"] & T \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "\eta\circ T"] \arrow[dr, equal] & T^2 \arrow[d, "\mu"] & T \arrow[l, "T\circ\eta"'] \arrow[dl, equal] \\ & T & \end{tikzcd}$$

Here $T\circ\mu$, $\mu\circ T$, $\eta\circ T$, and $T\circ\eta$ are all horizontal compositions. That is, $(T\circ\mu)_x = T(\mu_x)$, $(\mu\circ T)_x = \mu_{T(x)}$, $(\eta\circ T)_x = \eta_{T(x)}$, and $(T\circ\eta)_x = T(\eta_x)$ for all objects x of $\mathcal{C}$.

Now T is a functor, but if we pretend for the moment that it is a set, then $\mu : T^2 \to T$ would be a binary operation while $\eta : I \to T$ would be a zeroary operation or constant. The first diagram would then express the associativity of this operation, while the second would say that η was the identity for μ.

But although T is not a set, it *is* an object, namely an object of the category $\mathcal{C}^{\mathcal{C}}$ of endofunctors on $\mathcal{C}$, whose morphisms are all natural transformations between those endofunctors. Furthermore composition of these endofunctors is an associative binary operation on $\mathcal{C}^{\mathcal{C}}$, which we can think of as a kind of product, allowing us to view T^2 as the product of T with itself.

This perspective makes (T, μ, η) a *monoid object in the category of endofunctors on* $\mathcal{C}$. (For comparison an ordinary monoid is a monoid object in the category of sets with product taken to be ordinary cartesian product, namely a set together with an associative binary operation and an identity element.)

Examples

1. *The identity monad on $\mathcal{C}$.* For any category $\mathcal{C}$, $(I_{\mathcal{C}}, 1_{I_{\mathcal{C}}}, 1_{I_{\mathcal{C}}})$, i.e. the identity functor on $\mathcal{C}$ and two copies of the identity natural transformation on that functor, is easily seen to satisfy the monad conditions.

2. *The lift monad.* Let $T : \mathbf{Set} \to \mathbf{Set}$ be the functor $X + 1$ taking each set X to $X + 1$ defined as $X \cup \{X\}$ (assuming as usual that sets cannot contain themselves—the important thing is for $X + 1$ to adjoin *some* new element to X). Take $\eta : I_{\mathbf{Set}} \to T$ to be the transformation whose X-th component $\eta_X : X \to X + 1$ satisfies $\eta_X(x) = x$. Define $\eta_X : (X+1)+1 \to X+1$ as $\mu_X(x) = x$ for $x \in X$, and $\mu_X(X) = \mu_X(X+1) = X$. Thus η is the evident embedding of X in $X + 1$ while μ identifies the two new elements in $X + 1 + 1$.

It is easily verified that η and μ are natural. The associativity square commutes because identifying three elements can be done two at a time in any order. The left identity triangle commutes because its top edge embeds $X+1$ in $X+1+1$ by sending X to X, which μ then identifies with $X+1$. The right identity triangle commutes similarly except that X is sent to $X + 1$ which does not change the outcome of the identification.

3. *The Kleene star monad.* Let $T : \mathbf{Set} \to \mathbf{Set}$ be the functor taking each set X to the set X^* of finite strings on X, and taking each function $f : X \to Y$ to the function $T(f) : X^* \to Y^*$ which sends each word $w \in X^*$ to the result of replacing each letter x in w by the letter $f(x)$. (Had we taken X^* and Y^* to be monoids, $T(f)$ would have been a homomorphism, but we are here treating them as mere sets and so $T(f)$ is merely a function.) Let $\eta : I_{\mathbf{Set}} \to T$ be the transformation whose X-th component is $\eta_X : X \to X^*$ defined as mapping each element $x \in X$ to the string whose one letter is x. And let $\mu : T^2 \to T$ be the transformation whose X-th component $\mu_X : X^{**} \to X^*$ takes each string of strings in X^{**} to their concatenation ("flatten").

The naturality of each of η and μ is easily verified. The associativity square amounts to the assertion that flattening a string of strings of strings over X to a string over X yields the same string whether the flattening starts from the outside or the inside. For example $((ab)(a))((a)(bb))$ flattens to $abaabb$ via either $(ab)(a)(a)(bb)$ (remove the outside parentheses first) or $(aba)(abb)$ (the inside parentheses). And the top edges of the identity triangle amount to parenthesizing respectively the individual letters of each string in X^* to make it a string of X^{**}, so abb becomes $(a)(b)(b)$, or parenthesizing the whole string, so abb becomes (abb). In either case μ strips the parentheses out again to give back the original string.

4. *The power set monad.* Let $\wp : \mathbf{Set} \to \mathbf{Set}$ be the covariant power set functor which maps $f : X \to Y$ to the direct image function $\wp(f) : \wp(X) \to \wp(Y)$, namely the function mapping each subset $X' \subseteq X$ to $\{f(x) | x \in X'\}$. This example is as for Example 3 with sets in place of strings, i.e. we write $\{a, b\}$ instead of ab and do not distinguish it from either $\{b, a\}$ or $\{a, b, b\}$. The definitions of η and μ, and the verification of the monad conditions, are treated entirely analogously. Note that μ is simply arbitrary ("big") union $\bigcup$.

5. *The term monad.* Given any variety, let $T : \mathbf{Set} \to \mathbf{Set}$ be the functor sending each set X to the set of all terms on X (the elements of the free algebra $F(X)$), and sending each function $f : X \to Y$ to the function $T(f) : T(X) \to T(Y)$ which, given a term t in $T(X)$, produces the term in $T(Y)$ obtained by substituting $f(x)$ for each occurrence of x in t for all $x \in X$. Taking Example 3 to be the case of this for the variety $\mathbf{Mon}$ of monoids, the appropriate generalization of that example makes η_X the function that maps each $x \in X$ to the term x, and makes μ_X the function that flattens each term of terms to a term by performing the evident substitutions. (If we picture $T(T(X))$ as a tree with a boundary separating the upper term from the lower terms, then μ simply erases that boundary.) The generalization then extends in the obvious way to the verification of the monad conditions.

Examples 1 and 2 are also cases of the term monad, for respectively the empty signature (where the only terms on X are the elements of X themselves) and the signature consisting of a single constant (the variety of pointed sets).

Whether Example 4 is a case of Example 5 is a very nice question. That we understood it as a small twist on Example 3 suggests that it should be. All we need is a family of operations corresponding to concatenation and identity for monoids. But if we insist that the signature of a variety contain only finitely many operations, or less stringently that it be indexed by some set, then Example 4 cannot be understood as arising from any variety. If however we allow a signature to be indexed by a proper class then this allows us to view the category of complete semilattices as a variety whose signature consists of a proper class of operations $\bigvee_Y$ ranging over all sets Y, with each such operation taking as argument a Y-tuple (defined as a function from Y to L) of elements of the given complete semilattice L and returning its sup. Example 4 then arises as the case of Example 5 for the variety of complete semilattices.

6. Given an adjunction (F, U, η, ϵ) (the data of Definition 3 of an adjunction, with $F : \mathcal{C} \to \mathcal{D}$ and $U : \mathcal{D} \to \mathcal{C}$), the triple $(UF, \eta, U \circ \epsilon \circ F)$ is a monad on $\mathcal{C}$. Example 5 is the special case of this where $\mathcal{C} = \mathbf{Set}$, $\mathcal{D}$ is a variety, and F and U are the associated free and forgetful functors constituting the left and right adjoints of an adjunction between $\mathcal{C}$ and $\mathcal{D}$.

We now verify the monad conditions. Certainly UF is an endofunctor on $\mathcal{C}$. Furthermore $\eta : I \to UF$ and $U \circ \epsilon \circ F : UFUF \to UF$ are natural, being built from naturals by horizontal composition, and are of the appropriate types, respectively $I \to T$ and $T^2 \to T$.

To verify the associativity square, note that its vertices are functors and its edges natural transformations, composed at the upper right and lower left corners by vertical composition. The first of these composites can be expanded as the 2-category diagram

$$\begin{tikzcd}
\mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \\
\mathcal{C} \arrow[dr, "F"] & \mathcal{D} \arrow[ur, "U"] & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[ur, "U"] & \mathcal{C} \\
& \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[rr, equal] & & \mathcal{D} \arrow[ur, "U"] &
\end{tikzcd}$$

and the second as

$$\begin{tikzcd}
\mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \\
\mathcal{C} \arrow[dr, "F"] & \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[ur, "U"] & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & \mathcal{D} \arrow[ur, "U"] & \mathcal{C}. \\
& \mathcal{D} \arrow[rr, equal] & & \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[ur, "U"] &
\end{tikzcd}$$

But each of these compose to the same natural transformation from $UFUFUF$ to UF, namely

$$\begin{tikzcd}
\mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \\
& \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & \; & \mathcal{D} \arrow[ur, "U"] &
\end{tikzcd}$$

Hence the two paths of the associativity square are equal by the interchange law.

Finally we show the two identity triangles commute by expanding $\mu(\eta \circ T)$ as the 2-cell diagram

$$\begin{tikzcd}
\mathcal{C} \arrow[dr, "F"] & & \mathcal{C} \arrow[dr, "F"] \arrow[rr, equal] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \\
& \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] & {} & \mathcal{D} \arrow[ur, "U"] \arrow[u, phantom, "\Downarrow \eta"] &
\end{tikzcd}$$

and similarly $\mu(T \circ \eta)$ as

$$\begin{tikzcd}
\mathcal{C} \arrow[dr, "F"] \arrow[rr, equal] & {} & \mathcal{C} \arrow[dr, "F"] \arrow[d, phantom, "\Downarrow \epsilon"] & & \mathcal{C} \\
& \mathcal{D} \arrow[ur, "U"] \arrow[rr, equal] \arrow[u, phantom, "\Downarrow \eta"] & {} & \mathcal{D} \arrow[ur, "U"] & .
\end{tikzcd}$$

The two triangle identities in Definition 3 of adjunction ensure that each of these composites is the identity natural transformation on $UF = T$.. (So we obtained the identity triangles from the triangle identities.)

Our six examples form a pyramid whose base consists of Examples 1-4 and whose apex is Example 6, the most general among these examples. We now show that Example 6 is in fact the most general among *all* monads.

Theorem 3 *Every monad (T, η, μ) on $\mathcal{C}$ is isomorphic to some monad $(UF, \eta, U \circ \epsilon \circ F)$ where (F, U, η, ϵ) is an adjunction, with $\mathcal{C}$ the domain of F and the codomain of U.*

The burning question here is, what to take as the codomain $\mathcal{D}$ of F (and hence domain of U) for this adjunction? We give two proofs, named for their respective discoverers. In both proofs the monad can be understood as an equational theory and F can be understood as mapping objects of $\mathcal{C}$ to their associated free objects (free algebras when $\mathcal{C} = \mathbf{Set}$), and U as its associated forgetful functor. The proofs are distinguished by their choice of $\mathcal{D}$, respectively the pseudovariety (just the free algebras, suitably understood) and the variety (all the algebras, again suitably understood) associated with the equational theory. Both categories are constructed using nothing but the monad (T, η, μ) itself.

Proof: (Kleisli) For the category $\mathcal{D}$, Kleisli takes the objects to be those of $\mathcal{C}$. The morphisms are taken to be all morphisms of $\mathcal{C}$ of the form $\{f : a \to T(b)\}$, with $s_{\mathcal{D}}(f) = a$ ($= s_{\mathcal{C}}(f)$) and $t_{\mathcal{D}}(f) = b$ (which is not in general $t_{\mathcal{C}}(f)$). Given $f : a \to T(b)$ and $g : b \to T(c)$, the composite gf is defined as $\mu_c T(g) f$, that is, $a \xrightarrow{f} T(b) \xrightarrow{T(g)} T(T(c)) \xrightarrow{\mu_c} T(c)$. The identity at a is defined as η_a.

For the object part of F and U, take F to be the identity on objects of C and U to be T. For the morphism part, take $F(a \xrightarrow{f} b)$ to be $a \xrightarrow{\eta_a} T(a) \xrightarrow{T(f)} T(b)$, and take $U(a \xrightarrow{f} b)$ (bearing in mind that $a \xrightarrow{f} b$ in $\mathcal{D}$ is $a \xrightarrow{f} T(b)$ in $\mathcal{C}$) to be $T(a) \xrightarrow{T(f)} T(T(b)) \xrightarrow{\mu_b} T(b)$. As part of Exercise 1 below, $UF = T$.

Lastly, take the unit η of the adjunction to be that of the monad, and take the counit ϵ to have for its a-th component ϵ_a the identity morphism $1_{T(a)}$. As Exercise 1, complete the proof. ∎

Proof: (Eilenberg-Moore)

We begin by specifying the category $\mathcal{D}$. Given a monad (T, η, μ) on $\mathcal{C}$, define a ***T-algebra*** to be a pair (a, e) where a is an object of $\mathcal{C}$ called the *carrier*, and $e : T(a) \to a$ is a morphism of $\mathcal{C}$, called the *evaluation map*, such that the following diagrams commute.

$$\begin{tikzcd}
T(T(a)) \arrow[r, "T(e)"] \arrow[d, "\mu_a"'] & T(a) \arrow[d, "e"] & a \arrow[r, "\eta_a"] \arrow[dr, "1"'] & T(a) \arrow[d, "e"] \\
T(a) \arrow[r, "e"] & a & & a
\end{tikzcd}$$

Define a ***homomorphism*** of T-algebras to be a morphism $f : a \to b$ of $\mathcal{C}$ making the following diagram commute.

$$\begin{tikzcd}
T(a) \arrow[r, "e_a"] \arrow[d, "\mathcal{T}(f)"'] & a \arrow[d, "f"] \\
T(b) \arrow[r, "e_b"] & b
\end{tikzcd}$$

Homomorphisms of T-algebras compose vertically in the evident way. The identity on $T(a) \xrightarrow{e} a$ is 1_a, the identity on a. This completes the definition of the category $\mathcal{D}$.

Now the algebras of the variety in Example 5 constitute an example of T-algebras so constructed. By consideration of this connection the reader should now be able to construct (F, U, η, ϵ) (Exercise 2). ■

Exercises

1. Complete the Kleisli proof.
2. Complete the Eilenberg-Moore proof.