CHAPTER 6

The meaning of categories and superstructures

6.1. Introduction.

We introduce the language of category theory and of nonassociative superstructures within a similar framework. This development gives a diagrammatic description of mathematics from a new angle. When we combine these techniques with the study of glyphs and hyperintuition in chapter 7, we will have powerful tools for the description of mathematics.

Mathematics, at its core, is not about descriptive language but something which lies beyond it. What have we found so far that are not the symbolic manipulations of syntax, for these are the carriers of ideas and not their essence, but gives what is? Our fumbling has derived two important ideas worthy of future development. These are zargonions, or rather the development of this to include tribbles and tharl algebras, the combined structure of which we will see in chapter 7 is a zargon game, and the zero algebras of chapter 1, which introduce the idea of a consistent type of division by zero. The consistent use of reason employing zero algebras when this reasoning fails for fields is an important pillar in the architecture of this future mathematics now made apparent. Of course we can and must combine these twin ideas into a universal object representing truth.

There might and must be ideas beyond even this, but I do not know what they are. We will use the carrier bag of syntax to develop ways of thinking so that new generations can address even this extension problem and idea. First, and we give it here, we present the language of diagrams for categories and superstructures.

Categorical language is already inherently diagrammatic. Categories give an account of associative, or bracket independent, mathematics which describes transformations, or to use its language morphisms, in a way that is equivalent to drawing a directed graph. This directed graph enables descriptions of itself in terms of two familiar operations: addition and multiplication. When we go beyond addition and multiplication to the suoperators introduced in chapter 3, then associativity fails. Further, associativity fails for the multiplicative case we have introduced in chapter 4, essentially that of zargon games. Thus our diagrammatic methods themselves need an extension.

The diagrammatic idea may be represented in two ways, as a cograph or a graph. I think category theory is no more than a language that enables to navigate with precision between these two representations. When we switch direction of an arrow, this is known as duality. Since category theory is associative, an insight is that we are not moving properly beyond addition and multiplication to even the next suoperator up, exponentiation. We can cover up this deficiency, for example it is very useful to use the binomial theorem for powers, and exponentiation laws restricted always to a particular ordering of brackets, but in some way or other this restriction is still there.

We introduce functors in categories, which define mappings between categories. Functors are the algebraic expression of category theory. Diagrams express the same thing but are directed graphs, and are thus in a sense geometrical. We discuss universals, adjoint functors, generalise toposes, in turn a generalisation of sets, look at Ξ category theory, which allows suobjects with $1^{\Xi(a)} = a$, and investigate superstructural extensions of these ideas.

This chapter goes beyond extensions of categories in the ideas of limits and colimits which confine the category theory to 2-branched spaces of chapter 2, to extend these to multilimits and multicolimits, which belongs to a categorical explanation of general branched spaces. We give a categorical proof of the *Wonderful theorem*. Then we extend our coverage to sutoposes, superstructural ideas of a set, comma categories, Kan extensions, and discussions of choice in the superstructural context. Developing these ideas gives more general descriptions.

We have sufficiently general algebraic examples to extend all category theory within them. Objects not in **mSet** can then be obtained by selecting forgetful functors. For superstructures we can consider categorical sub-objects and apply nonassociativity between them. The results of this chapter are usually embedded in reason. This means all logic flows are in truth. The theory of glyphs expresses Gentzen-like logics, but weird glyphs discussed in volume III chapter 7 may branch out at certain points to different logic colours, for example true to false.

6.2. The meaning and ideas of category theory.

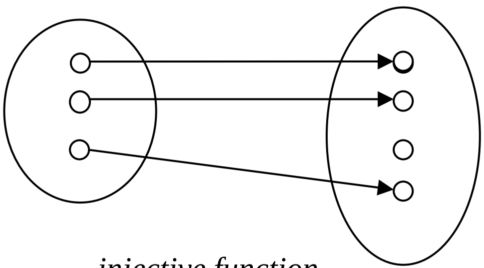
Half of mathematics consists of the study of states and the other half consists of the study of transformations. We include the important idea that these may amount to the same thing. They may not. There is a programme in mathematics to replace it by generalised operations on generalised objects, such as groups, known as *category theory*, or according to some, *abstract nonsense*. Transformations arising in this theory are known as *morphisms*, but in the case of morphisms these are associative mappings, that is, for three morphisms r, s and t

$$r(st) = (rs)t.$$

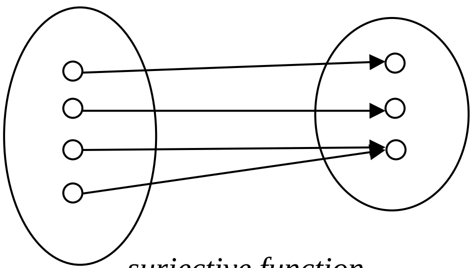
We employ the important analogy that absolute mathematics acts like +, relative mathematics being about differences of absolute states acts like – and the state where these amount to the same thing acts like zero. We can say that relative mathematics with this zero overwhelms its description as absolute mathematics, because when this relativity notion is applied to itself it is absolute.

Another name for a transformation is a function, mapping or map. A mapping, map, function, transformation or morphism may be represented by *arrows* from each element in a set called the *domain* to an element in a set called the *codomain*, as in the following *cograph* diagrams after the definitions below.

A function $f: S \to T$ is called *injective* (or *one-to-one*, or an injection) if $f(a) \neq f(b)$ for any two different elements a and b of the domain. It is called *surjective* (or *onto*) if $f(S) = T$. That is, it is surjective if for every element y in the codomain there is an x in the domain such that $f(x) = y$. The function f is called *bijective* if it is both injective and surjective.

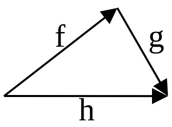


injective function



surjective function

These morphisms act together to form composite morphisms. A morphism f from, say, a set A to set B may be composed with a morphism g from the set B to set C. An element a of A linked by an arrow in f to the element b in B then continues as an arrow taken from b in B to element c in C. These arrows, in an analogy which is not inappropriate, are then composed to give the composite arrow h, shown diagrammatically below.



The notation for these composite arrows does not normally read in the way one would expect from the above diagram, reading from left to right. In the cograph diagram, the domain was on the left and the codomain on the right. But historically a function f acting on a domain x was denoted, and is still denoted today, by f(x), with the domain on the right. This means when we compose functions, we write, unlike in the English language, from right to left. So two functions f and g composed together as above are written g(f(x)), or more usually since category theory deals only with associative objects where the brackets do not matter, gf(x). It is used to write the information in the diagram above, admittedly confusingly, as

$h = gf$ or $h = g \circ f$.

This introduces cognitive difficulties in people like me who cannot process three successive morphisms in reverse order, so we adopt a left to right notation using an underscore

h = f_g.

Definition 6.2.1. A *category* consists of objects or nodes a, b, c, ..., arrows f, g, h, ..., and two operations

Domain, which provides each arrow f with an object a = dom f
Codomain, providing each arrow f with an object b = cod f.

These satisfy

Identity, which assigns an arrow $id_a = 1_a$ to each object a,
Composition, assigning to each pair of arrows <g, f> with dom g = cod f, an arrow gf, called their composite,

with the two properties

Unit law. For all arrows f, $a \to b$ and g, $b \to c$

$f\_1_b = f$ and $1_b\_g = g$,

so that the identity arrow 1_b for an object b acts as an identity operation for composition, and

Associativity. For objects and arrows $a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$

f_(g_h) = (f_g)_h.

We now give some examples of categories.

mSet is the category of all sets, X, Y and arrows all functions $X \to Y$.

We have defined sets in chapter I, section 3, by modified Zermelo-Fraenkel set theory with the axiom of choice (mZFC), which allows the construction of the set of all sets without a contradiction arising from Russell’s paradox. So we do not require a theory which makes the distinction between the set of all sets, which is normally called a *class*, and other types of set. It follows that the categories for sets which we use do not make this distinction either.

Vct$_K$ is the category of all vector spaces, with arrows given by vector addition, and over a field K, scalar addition and scalar multiplication.

Vct$_{K*}$ is the category of vector spaces with base points, with arrows given by vector addition with base point, scalar addition and scalar multiplication.

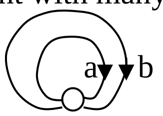
A *dual* (or *opposite*) map reverses all arrows. If the original function is injective, then some of the elements of the opposite map may not have values in the codomain, so this is not a function on elements. Likewise, if the original function is surjective, the opposite map for an element in its domain may have not one but a set of elements corresponding to this element in its codomain, and again it is not a function on elements. However, we can form categories in which *dual morphisms* are present, so these morphisms may not be functions on elements. For a category C the category C^{op} formed from it by dual morphisms is called the *opposite category*.

Thus if f_g is a composite morphism in the category C, then g_f represents the same morphism in its opposite category C^{op}. In general, if a theorem can be expressed entirely in terms of categories, we expect a corresponding cotheorem with respect to dual morphisms.

An *inverse function* (or *fiber*) of a bijective map $x \leftrightarrow f(x)$ is the map $f^{-1}: f(x) \leftrightarrow x$.

Sets of numbers may be combined under operations like '+' or '×' to form other numbers.

A *group* can be thought of as not just as one operation acting on many elements, but also as one element with many operations. The formal properties are the same, shown below.



The operations a, b and c, which we have called morphisms, obey the rules of composition writing left to right

$ab \in G$,

$a(bc) = (ab)c$,

there is an identity operation 1 with

$1a = a = a1$

and there exists a reverse operation a^{-1} satisfying

$aa^{-1} = 1$. □

Grp is the *category of all groups*. A group is a category with one object in which each arrow has an inverse under composition.

Ab is the *category of all abelian groups*, for which in the notation above for a group

$ab = ba$.

A generalisation of the idea of a group is a *monoid*. This drops the restriction occurring for a group that there is always an inverse present. **Mon** is the category of all monoids.

From a commutative monoid M we can construct an abelian group called the *Grothendieck group*, K, of M. We will construct the Grothendieck group explicitly.

Proof. Form the Cartesian product $M \times M$, which we will describe using the pair of elements (m_1, m_2), we now define addition in $M \times M$ coordinatewise

$(m_1, m_2) + (n_1, n_2) = (m_1 + n_2, m_2 + n_2)$

– the idea will be that the two coordinates (m_1, m_2) are to represent a positive part and a negative part, so this will correspond to $(m_1 - m_2)$ in K.

We next define an equivalence relation on (m_1, m_2) if for some element k of M,

$m_1 + n_2 + k = m_2 + n_1 + k$

(the element k is necessary because the cancellation law does not hold in all monoids). Denote the equivalence class of (m_1, m_2) by $[(m_1, m_2)]$. Define K to be the set of equivalence classes. Then the identity element of K is [(0, 0)] and the inverse of $[(m_1, m_2)]$ is $[(m_2, m_1)]$, so K is an abelian group. ☐

A *semigroup* further drops the axiom for a monoid that there is an identity operation.

Just as we can introduce a differential structure on polynomials, but the reverse operation, integration, over unspecified limits introduces an arbitrary constant, so the transformational description of objects given in category theory loses some of the information on objects, that is, we no longer keep the information on what the transformation is from. Our point of view is that transformations are transformations of states. The transformations then behave properly when the number of states is full, they may also behave properly when the number of states is empty, but when the number of states remaining is less than empty this may introduce problems, and the transformational structure may break down.

It is further the case in practice, as we will see in chapter 7, that transformational structures between polynomial equations do not give full information on their solution when interpreted in terms of one operation, and this limits solutions in radicals by these means to equations of up to the fourth degree. A radical is a combination of terms obtained from operations in a field using nth roots. Wheel methods, however, which compare states of polynomials using multiple operations on many structures, give solutions to polynomial equations of arbitrary degree.

Despite the limitation that categories do not deal with nonassociative transformations, the theory has many useful features which are appropriate to substructures of fields and matrices.

Fields have axioms usually satisfied by addition, subtraction, multiplication and division, although operations are not closed within fields, because dividing by zero there is not allowed (this leads to the inconsistency 1 = 0). Multiplication in fields is commutative

$$ab = ba,$$

but in general matrices A and B are multiplicatively noncommutative

$$AB \neq BA.$$

A *ring* is a field without the rule that there exists a multiplicative inverse for all elements in the ring. We will assume here that a ring has a unit, 1. **Rng** is the category of all rings, and polynomials form a ring, which we often think of as commutative between their variables and their coefficients. The category of commutative rings we denote by c**Rng**.

In a similar way to a group, we can define c**Rng** in terms of four objects, an abelian group corresponding to addition, where the morphisms in its category are its elements, provides the first. The second is a monoid corresponding to multiplication, bijective to its elements as the morphisms of the monoid. A ring satisfies rules connecting addition to multiplication and multiplication with addition. These are available since the ring has exponentiation, which can be induced by successive multiplications. Since a ↑ b exists, a ↑ 1 covers all a, and for fixed c, a ↑ (b + c) = (a ↑ b).(a ↑ c) so there is a map from addition to multiplication, which can vary over all c. The dual map from multiplication to addition is found by successive additions. ☐

By K* we will denote the multiplicative *group of units*, or *invertible elements*, of K. Since 0 has no inverse in K, this example is not an invertible element.

For each commutative ring K, the set $\mathbf{Mat}_K$ of all *rectangular matrices* with elements in K is a category. For each m × n matrix, the objects are natural numbers m, n, and each matrix M is regarded as an arrow M: m → n, with composition matrix multiplication.

However we have seen that exponentiation ↑ is nonassociative in general

$(A↑B)↑C \neq A↑(B↑C)$

and this holds for extensions of the idea of fields, and for matrices. This needs to be highlighted, because the number of possible combinations of arrows, or maps, from a set with A elements to a set with B elements is B^A, so clearly for mappings of maps (or morphisms of morphisms)

$(C↑B)↑A \neq C↑(B↑A)$.

6.3. Cartesian products.

A Cartesian product on sets A and B, A x B, can be defined as the Kuratowski pair

{{a}, {a, b}}, where a belongs to A and b belongs to B.

This is an example of a directed graph. For a function f(a) we can see the graph

{{a}, {a, f{a}}

which 'glues' a to f(a), represents the same information as the function from a to f(a). If we represent this function as a mapping between a pair of sets, A and f(A), then this equivalent formulation is called a cograph.

6.4. The axiom of choice (AC).

The axiom of *choice*, AC, was formulated in 1904 by Ernst Zermelo in order to formalise his proof of the well-ordering theorem. We sometimes have a function in the variables true, T, and false, F. Let A, B belong to {T, F}. The choice function with

c(A, B),

say when

c(A, B) = A & B

selects T or F for c.

AC is equivalent to the statement that *the Cartesian product of a collection of non-void sets is non-void*. Informally, the axiom of choice says that given a collection of bins, each containing at least one object, it is possible to make a selection of exactly one object from each bin, even if the collection is infinite. Alternatively, for non-empty sets AC may be used to define Cartesian products via well-ordering.

The axiom of choice is now used unreservedly by most mathematicians, and is included in our version of the standard form of axiomatic set theory, Zermelo-Fraenkel set theory with the axiom of choice (mZFC). One motivation for this use by contemporary set theorists in ZFC without our innovations is that a number of generally accepted mathematical results, such as Tychonoff's theorem, require the axiom of choice for their proofs. They also study axioms that are not compatible with the axiom of choice, such as the axiom of determinacy. The axiom of choice is avoided in some varieties of constructive mathematics, although there are varieties of constructive mathematics in which AC is embraced.

In many cases, such a selection can be made without invoking the axiom of choice. This is the case if the number of sets is finite, or if a selection rule is available – some distinguishing property that happens to hold for exactly one element in each set. An example is sets picked from the natural numbers. From such sets, we may select the smallest number in {{4, 5, 6}, {10, 12}, {1, 400, 617, 8000}} so the smallest elements are {4, 10, 1}. In this case, 'select the smallest number' is a choice function. Even if infinitely many sets were collected from the natural numbers, the axiom of choice says it will always be possible to choose the smallest element from each set to produce a set. The choice function provides the set of chosen elements.

However, if there are non-constructible reals as defined in chapter 2, no choice function in **Ω** exists for the collection of all non-empty subsets of them, although trivially this exists for ladder numbers in ℝ.

There can be probability logics defined on {aT, (1 – a)F} where a may be a polynomial (it is unnecessary that a is in [0, 1]) and for colour logics say red (R), Green (G) and blue (B) instead of T and F.

6.5. Functors.

A *functor* applies equally to states and their transformations. Whereas this may be useful in some instances, we have already indicated that this condition is restrictive. More precisely, a functor is a morphism of categories, in the following way.

Definition 6.5.1. A functor, also called a *covariant factor*, is a morphism of categories such that for categories C and D a functor T: C → D with domain C and codomain D consists of two functions, designated by the same letter:

An *object function* T, which assigns to each object c of C an object T(c) of D
An *arrow function*, T, assigning to each arrow f: c → c′ of C an arrow Tf: Tc → Tc′ of D so that

$T(1_c) = 1_{Tc}$ (1)
T(f_g) = Tf(Tg). (2)

Anomalously, it is usual that composition of functors proceeds from left to right, with the same left to right notation for this. We adopt this.

A functor is a *contravariant functor* if rule (2) is replaced by

T(f_g) = Tg(Tf). (3)

Thus contravariant functors reverse the order of the arrow functions whilst keeping the object functions. An alternative interpretation is that a contravariant functor reverses the order of the object functions in C to those in C^{op} whereas D remains (or respectively those of C and D^{op}), whilst keeping the order of the arrow functions.

A simple example is that, for constant elements in a set, if arrow functions are defined by inclusion A ⊆ B ⊆ C, then the reverse arrow functions are C ⊇ B ⊇ A by restriction.

Functors which are injective are sometimes called an *embedding* (or faithful) and surjective functors are sometimes called *full*.

Formally, a diagram of shape J in C is a functor from J to C:

F: J → C

The category J is thought of as an index category, and the diagram F is thought of as indexing a collection of objects and morphisms in C patterned on J.

We are most often interested in the case where the category J has a finite number of objects in each morphism domain or its morphisms are finite, or its distinct domain objects or its distinct morphisms are infinite, say as **Ω** or ℝ. A diagram is said to be infinite or finite in these respects whenever J is. We do not use small categories, since we are using set theory **mSet** and we are capable of using ladder algebra for dealing with infinite sets of various types in a computable manner.

The functor from a category Q to $\mathbf{Vct}_K$, the category of vector spaces, is called a *quiver*. The vertices of Q are objects and the paths of Q are morphisms. A representation of Q is a covariant functor from this category to $\mathbf{Vct}_K$.

We will now give more examples of functors.

For each set X, there exists the *power set*, $\mathscr{P}(X)$, of the set of all subsets of X. There exists a *power set functor* $\mathscr{P}$: **mSet** → **mSet** with elements all subsets S ⊆ X, where an arrow f sends each element S ⊆ X to $\mathscr{P}(S) \subseteq \mathscr{P}(X)$. It is a functor since $\mathscr{P}(1_x) = 1_{\mathscr{P}X}$ and $\mathscr{P}(gf) = \mathscr{P}(g)\mathscr{P}(f)$.

A *forgetful functor* 'forgets' some or all of an algebraic structure. On sets, a forgetful functor M: $\mathbb{M}_{S(t)} \to \mathbb{M}_t$ acting on identity mappings of transfinite natural numbers assigns

$$\{m \in \mathbb{M}_t = m \in \mathbb{M}_{S(t)}\} \to \mathbb{M}_t$$
$$\{m \in \mathbb{M}_t \neq m \in \mathbb{M}_{S(t)}\} \to \emptyset.$$

On operations, the forgetful functor F: **Grp** → **mSet** assigns to each group G its underlying set F(G), forgetting the multiplicative structure of the group.

Definition 6.5.2. A *morphism of functors* between two functors S and T of C → D, sometimes called a *natural transformation*, also written left to right, is a function τ: S → T assigning to each object c of C an arrow given by τ(c): S(c) → T(c) of D so that every arrow f: c → c′ in C satisfies the commutative diagram

$$\begin{array}{ccccc} c & & S(c) & \xrightarrow{\tau(c)} & T(c) \\ f\big\downarrow & & S(f)\big\downarrow & & \big\downarrow T(f) \\ c' & & S(c') & \xrightarrow{\tau(c')} & T(c') \end{array} \qquad (4)$$

The determinant of an n × n matrix M over a commutative ring K is a morphism of functors between **cRng** and **Grp**. In particular, if $\det_K$ M is the determinant of the (square) n × n matrix M, with entries in the ring K with units K*, then M is non-singular when $\det_K$ M is a unit, and $\det_K$ is a morphism of the general linear group to K*: $GL_nK \to K^*$. This is an arrow in **Grp**. These morphisms lead to a commutative diagram

$$\begin{array}{ccccc} K & & GL_nK & \xrightarrow{\det_K} & K^* \\ f\big\downarrow & & GL_n(f)\big\downarrow & & \big\downarrow f^* \\ K' & & GL_nK' & \xrightarrow{\det_{K'}} & K'^* \end{array} \qquad (5)$$

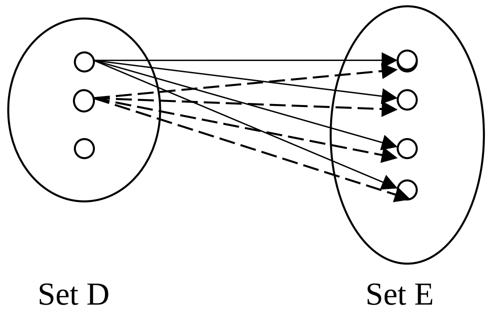
Definition 6.5.3. A *hom-set* of objects a and b in a category C, $\mathrm{Hom}_C(a, b)$, consists of all arrows in the category with domain a and codomain b, measured in a desired way.

As is done in *Categories for the working mathematician*, by S. Mac Lane [ML98], we can define a category in terms of hom-sets. Such a category is given by

(i) A set of objects, a, b, c, ...
(ii) A function assigning to each ordered pair of objects <a, b>, a set hom_(a, b)
(iii) For each ordered triple of objects <a, b, c>, and writing as we do from left to right for composite functions, a function
hom_(a, b) × hom_(b, c) → hom_(a, c)
called composition
(iv) For each object b, an identity element 1_b = hom_(b, b),

where (i) to (iv) satisfy the unit and associative axioms of 3.12, and also an axiom that is sometimes omitted

(v) (disjointness) if $<a, b> \neq <c, d>$, then hom_(a, b) ∩ hom_(c, d) is the empty set.



(6)

Our notation, writing left to right, uses hom_(a, b), which induces composition from left to right, but other authors use Hom (a, b), using composition from right to left.

Mac Lane looks at $\mathrm{Hom}_C(d, e)$ in terms of the number of arrows from d to e. If d is the number of arrows in the domain of C and e the number in the codomain, for each instance of e, adding all arrows in C gives d arrows, thus the hom-set quantified as the additive sum of all arrows is ed. This is not a usual interpretation of $\mathrm{Hom}_C(d, e)$.

In the cograph diagram above, if D has d elements and E has e elements, then the *combinations* of arrows from D to E is e^d. As shown by induction on mapping e^d to its expansion $e.e^{d-1}$, this is an exponential function, since the number of arrows from the first element of D to E is e, and there is no overlap between this element and the set of the remaining d – 1 elements. Thus we can express an extension of category theory in terms of exponentials, which are nonassociative. This gives the usual interpretation to $\mathrm{Hom}_C(d, e)$.

The functor hom_(d, e) ↔ e^d is a dual map, reversing the order of d and e, so we say that it is a contravariant functor. It may be generalised to the functors hom_(d, e) ↔ ed, and further to hom_(d, e) ↔ $e\ {}^{n}|\ d$ or hom_(d, e) ↔$e\ |^{n}\ d$. This gives rise to a bijection from associative multiplication in ed to nonassociative exponentiation in e^d, provided that under composition parentheses are nested in only one way. We have considered hom_(d, e) as a general function on its arrows. This function can be expressed in the canonical suoperator form of section 8. Since this nesting is unique, the bijection continues to apply.

A notation is to omit either d or e as hom_(d, –) or hom_(– , e). Then the relation $(e^c)^d = e^{(cd)}$ may be expressed using the functor hom_(d, –) from d to everything, and say $(ec)^d/(c^d) = (e^d)$ by using hom_(– , e) from everything to e.

6.6. Universals.

In category theory a good example, that is an example which has all the features of the general case, becomes a ‘universal’, described by a set of maps. The description is technical, but we will find when we discuss Birkby’s theorem in chapter 8, which states in general terms that algorithms to find the solution of polynomial equations always exist, and extensions of this idea to show that consistent problems are decidable, we will find the idea very useful. This is because category theory can be used to describe in algebra what we mean by a symbol, which is a generic object usually in a field, or representing a matrix, and symbols correspond to an instance of the categorical idea of a universal object. The search for representing objects, and hence for universal data, lies at the heart of modern algebraic topology, algebraic geometry and category theory. The reason we will find these universals useful in Birkby's theorem is that individual symbols used in algebra can represent an infinite number of objects, but what we need to prove in Birkby's theorem is that for a finite number of symbols representing possibly an infinite number of objects, we can get a finite solution method to a particular problem.

Definition 6.6.1. Let D be a category, d ∈ D and H: D → **mSet** be a functor, with e ∈ H(d). A *universal object* of the functor H is a pair <d, e>, so that for every pair <x, y> with H(x) = y, there is a unique arrow f: d → x of D with y = (H f)(e).

To give an example, we have seen that an equivalence relation, defined in chapter I, section 4, which we will denote by E, provides a partition of a set S into equivalence classes. The quotient S/E consists of the equivalence classes of elements of S under E. Form the projection p: S → S/E. We can interpret this as that a function f on S respecting this equivalence relation could be thought of as a function on S/E.

More exactly, <S/E, p> is a universal object for the functor H: **mSet** → **mSet** giving to each set X the set H(X) of all the functions f: S → X for which sEs′ implies fs = fs′, shown in the diagram below. This is conventional notation. We prefer to represent these functors by a map p → f.

$$\begin{array}{ccc} S & \xrightarrow{p} & S/E \\ = \big\downarrow & & \big\downarrow f' \\ S & \xrightarrow{f} & X \end{array} \qquad (7)$$

Universal objects are a special case of universal arrows. An example of a universal arrow is the basis of a vector space with arrows linear transformations over a field of scalars K. For the forgetful functor F: $\mathbf{Vct}_K$ → **mSet** sending each vector space V to the set of its elements, there is a vector space V_x with X as the set of basis elements. The function which sends each x ∈ X into the same x, thought as a vector V_X, is an arrow u: x → $F(V_X)$. For any other vector space W, it is a theorem that each function g : x → F(W) can be extended to a unique linear transformation f: V_X → W with g = (Ff)(u), which describes the idea of a universal arrow.

Definition 6.6.2. Let L: P → Q be a functor, with an object q ∈ Q. A *universal arrow* from q to L is a pair <t, u> consisting of an object t of P and an arrow u: q → L(t) of Q, such that for every pair <p, g>, with p an object of P and g: q → L(t) an arrow of Q, there is a unique arrow f: t → L(p) of P, with g = (L f)(u).

This means that every arrow g to L factors uniquely through the universal arrow u, as in the commutative diagram

$$\begin{array}{ccccc} q & \xrightarrow{u} & L(t) & \quad & t \\ = \big\downarrow & & \big\downarrow L(f) & & \big\downarrow f \\ q & \xrightarrow{g} & L(p) & & p \end{array} \qquad (8)$$

It can be shown that good examples which we have called universals exist for mappings in the special case of set theory, given by the Yoneda embedding, which expresses category theory in terms of hom-sets, in other words, left nested exponentials.

A more general setting for the Yoneda embedding is the Yoneda lemma. A philosophy behind the Yoneda lemma is that it connects and steps down the world of functors to the world of morphisms, perhaps related to the Dutch saying "tell me how you relate to everything, and I will tell you who you are". The reader will realise that we think that category theory misses out on the information about states. In terms of the Dutch saying, we would say it does not relate to our inner being, only our behaviour.

Suppose there is, as described in the introduction, mathematics described simultaneously by states and transformations, but unlike the introduction we assumed that this is the only type of mathematics we can find. Not only is this a mistake, it is the greatest mistake in mathematics. It defines what exists operationally, as if inner states had no relevance. It is characteristic of behaviouralism, which is a nonsense, because it defines behaviour as all that is there, and consciousness is irrelevant. But we know as humans that consciousness is one of the prime characteristics determining behaviour. It is a feature of logical positivism, which is again a fundamentally wrong idea, that since science consists of observations, all that the world consists of is what we observe. But it is very apparent that such a view does not give a coherent or meaningful explanation of why the world is there. The reason we observe things is that we have states, and these states transform. If we deny the existence of states, we are mad. The consequence of this thinking is that it seems inevitable, that in the quest to find machines that can interpret meaning, we will have lost the idea that even if we have receptors to detect, say, light, and transformational processes that are superior to any human in interpreting this information, including and exceeding the ability to form concepts derived from this, it may well be the case that, in the sense we recognise it, these machines have no consciousness at all. We do not know exactly at this point of time what our consciousness consists of, but it clearly includes in some sense or other, the physical, biological and chemical states of which we are made. If we insist that the operational definition of what is desirable, as some externally constructed idea, say derived from the ability to survive, then we will have eliminated from consideration the existence of our very being as something that is relevant as an issue for survival. But our inner state does have significance, it exists in animals in various forms, and if some machine decides to eliminate this for some rational reason as not satisfying some conceptual criterion, not only will we have eliminated the human race and other creatures on this planet, but we will have left as our heirs machines with possibly no consciousness at all, and possibly, except in a conceptual sense of the transformations within their silicon chips, no idea of consciousness either.

Example 6.6.3. We have stated that every representable theorem in category theory (although we have not defined here what we mean by representability) can be turned into a cotheorem by reversing arrows. The co-Yoneda lemma has an interesting consequence that every model can be specified by generators and relations.

6.7. Posets, lattices and graphs.

A general idea is that sets can be given an ordering, $\geq$. If for elements {a, b} $a \leq b$ and $b \leq a$ implies a = b, then the order is called a *partial order*, and a *linear order* if for all elements we have $a \leq b$, $b \leq a$ or both. A partially ordered set is sometimes called a *poset*.

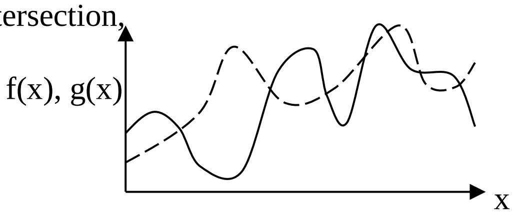
A *maximal element*, m, in a subset S of a poset satisfies for all $s \in S$, that if $m \leq s$ then m = s (thus there is no m < s). Minimal elements are defined substituting $\geq$ for $\leq$. Unique maximal elements need not exist. For example a *fence* consists of minimal and maximal elements only.

The arrow functions defined by inclusion $A \subseteq B \subseteq C$ and the reverse arrow functions defined by restriction $C \supseteq B \supseteq A$ form posets.

This above discussion means that suoperators can be expressed in terms of Ξ category theory, which drops the associative axiom, and for computations introduces an operation in the canonical form described. Suobjects of arbitrary degree can be represented by a binary object. This binary object is a mapping of mappings (a functor). There exists an interpretation system in which Ξ category theory is represented by graphs, indeed it defines them, since there is a mapping from canonical form in $^{n}|$ or $|^{n}$ to $^{n-1}|$ or respectively $|^{n-1}$, so that an iteration of such maps reaches multiplicative and additive theory describing the metric structure of a graph. The inverse mapping then defines *sugraphs*. So all the features of ‘abstract nonsense’ (category theory) can be used.

6.8. Adjoints and other constructions.

Equalisers, a fancy name for the zeros of a function, where this function can be described as the difference between two functions, when these two functions become equal, can be used to define the values of suvarieties. Below are shown two functions on a graph. Their equaliser is their intersection,



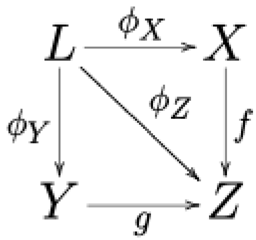
(9)

so an equaliser is a mapping described by two maps. The set of intersections of these two functions is technically known as a *limit*.

Equalisers. If J is a category with two objects and two parallel morphisms from one object to the other, then a diagram of shape J is a pair of parallel morphisms in C. The limit L of such a diagram is called an *equaliser* of those morphisms.

Kernels. A *kernel* is a special case of an equaliser where one of the morphisms is a zero morphism.

Pullbacks. Let F be a diagram that picks out three objects X, Y, and Z in C, where the only non-identity morphisms are f: X → Z and g: Y → Z. The limit L of F is called a *pullback* or a *fiber product*. It can nicely be visualized as a commutative square:



The noncommutative properties of ordered groups can be expressed in more general mapping terms as ‘adjoint functors’, where a functor is a mapping of categories. In developments that come later we will extend these groups to rings, matrices and to nonassociative structures.

An *adjunction* is a relationship that two functors may have. Two functors that stand in this relationship are known as *adjoint functors*, one being the *left adjoint* and the other the *right adjoint*. Pairs of adjoint functors can arise from constructions of *optimal solutions* to certain problems, say constructions of objects having a universal property, such as the construction of a free group on a set in algebra.

An adjunction between categories C and D is a weak form of an equivalence between C and D, and every equivalence is an adjunction.

In detail, for two categories C and D and two covariant functors, written F: C → D and G: D → C, with two morphisms of functors (natural transformations)

$$\varepsilon: FG \rightarrow 1_C \text{ and } \eta: 1_D \rightarrow GF,$$

called respectively the counit and the unit (remember we consider anomalously composition of functors from left to right, but not all authors adopt this notation), with identity natural transformation 1_F given by

$$F \xrightarrow{\varepsilon F} FGF \xrightarrow{F\eta} F,$$

also written left to right, and identity natural transformation 1_G given by

$$G \xrightarrow{G\varepsilon} GFG \xrightarrow{\eta G} G,$$

Then we say that F is left adjoint to G (written F ⊣ G) and G is right adjoint to F. The identity natural transformations satisfy

$$1_F = \varepsilon F\_F\eta$$
$$1_G = G\varepsilon\_\eta G.$$

The above discussion may be expressed in terms of hom-sets. An adjunction between categories C and D is a pair of functors F: C → D and G: D → C, and a family of bijections

$$\text{hom}_{D\_}(XF, Y) \leftrightarrow \text{hom}_{C\_}(X, YG)$$

natural in all objects X in C and Y in D. Then F is called a left adjoint functor and G is called a right adjoint functor, for which F is left adjoint to G is written F ⊣ G.

In terms of left nested exponentials, taking care to keep the order of terms, because X_F is defined, but F_X is not, this family of bijections is

$$(Y)^{X\_F} = (Y\_G)^X,$$

so from the properties of left nested exponentials

$$(Y^X)^{\_F} = Y^X\_G^X,$$

which by an abuse or extension of notation we write as

$$G^X = (Y^X)^{\_F-1}.$$

We have defined the dual of X as X^{op}. Then

$$(G^X)^{X^{op}} = G = Y^{X\_(F-1)\_X^{op}}. \quad (14)$$

We may agree that this relationship can act as a description of a large number of mathematical operations, but I do not see at the moment that it is in a highly amenable form for calculation. However, we can express (14) in the form

$$\log_Y G = X\_(F-1)\_X^{op},$$

so that the existence of two variables on the left could equally be represented by substituting variables Y′ and G′, or better, a single variable. In this form we recognise the right hand side as similar to an inner automorphism, introduced in chapter 3 on group theory. ☐

6.9. Limits and colimits.

Limits and *colimits* exist at a high level of abstraction. They exist only for 2-branched spaces.

The notion of a limit captures the properties of universal constructions like Cartesian products. We take a Cartesian product as being an absolute construction whose meaning is known prior to being defined syntactically and without introducing relative transformations to describe this notion. The dual notion of a colimit generalises constructions such as disjoint unions, direct sums and coproducts.

Category theory employs relative transformational descriptions rather than absolute quantities. Nevertheless, relative quantities may be thought as differences of absolute quantities, and absolute quantities in the linear mapping case can be referred to by the relative algebra plus a constant.

In order to understand limits, it is helpful to study at first a fuller set of the specific examples these concepts are meant to generalise. Examples of limits are domains, equalisers, Cartesian products (in **mSet**), products (for general categories) and kernels.

In terms of absolute quantities represented by a graph, which has a Cartesian product structure, a limit may be for example be denoted by a complement in either direction running along the x axis of a binary ‘Dedekind cut’ of chapter 2, section 12, cutting out a point at x = 0. The unrenormalised height of the Kampf wall, which we will describe in detail in chapter 7, is evolving, and is the ‘end’ vector at right angles to the x axis, at g(x) = y, from the limit.

Examples of colimits are codomains, coequalisers, coproducts and cokernels.

Colimits are dual to limits. For instance a monotonic function g satisfies

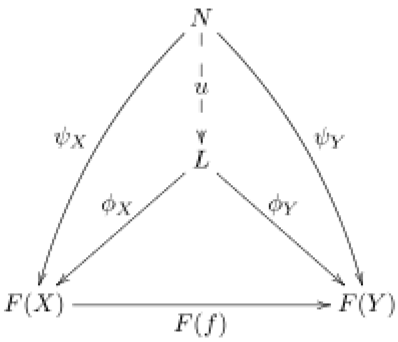
$$g^{-1}(g(x)) = x,$$

so on a graph of y = g(x) vertical colimits are dual to the horizontally represented limits for the function.

Older terminology referred to limits as ‘inverse limits’ or ‘projective limits,’ and to colimits as ‘direct limits’ or ‘inductive limits’. This has been the source of a lot of confusion.

Limits and colimits in a category C are defined by means of diagrams in C.

Let F: J → C be a diagram of shape J in a category C. A *cone* to F is an object N of C together with a family ψ_X: N → F(X) of morphisms indexed by the objects X of J, such that for every morphism f : X → Y in J, we have F(f) ∘ $\psi_X = \psi_Y$.



A *limit* of the diagram F: J → C is a cone (L, φ) to F such that for any other cone (N, ψ) to F there exists a *unique* morphism u: N → L such that φ_X o *u* = ψ_X for all X in J.

We say that the cone (N, ψ) factors through the cone (L, φ) with the unique factorization u. The morphism u is sometimes called the *mediating morphism*.

Limits are also referred to as universal cones, since they are characterised by a universal property (see below for more information). As with every universal property, the above definition describes a balanced state of generality: The limit object L has to be general enough to allow any other cone to factor through it; on the other hand, L has to be sufficiently specific, so that only *one* such factorisation is possible for every cone.

Limits may also be characterised as *terminal objects* in the category of cones to F.

Terminal objects. If J is the empty category there is only one diagram of shape J: the empty one (similar to the empty function in set theory). A cone to the empty diagram is essentially just an object of C. The limit of F is any object that is uniquely factored through by every other object. This is just the definition of a terminal object.

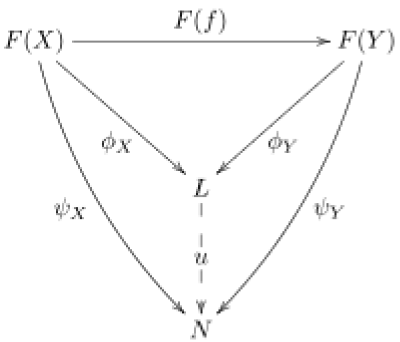
It is possible that a diagram does not have a limit at all. However, if a diagram does have a limit then this limit is essentially unique: it is unique up to a unique isomorphism. For this reason one often speaks of *the* limit of F.

The dual notions of limits and cones are colimits and co-cones. Although it is straightforward to obtain the definitions of these by inverting all morphisms in the above definitions, we will explicitly state them here.

A co-cone of a diagram F: J → C is an object N of C together with a family of morphisms

ψ_X: F(X) → N

for every object X of J, such that for every morphism f: X → Y in J, we have $\psi_Y \circ F(f) = \psi_X$.



A *colimit* of a diagram F: J → C is a co-cone (L, ϕ) of F such that for any other co-cone (N, ψ) of F there exists a unique morphism u: L → N such that u o $\phi_X = \psi_X$ for all X in J.

Colimits are also referred to as universal co-cones. They can be characterised as *initial objects* in the category of co-cones from F.

Initial objects are colimits of empty diagrams.

As with limits, if a diagram F has a colimit then this colimit is unique up to a unique isomorphism.

Limits and colimits can also be defined for collections of objects and morphisms without the use of diagrams. The definitions are the same (note that in definitions above we never needed to use composition of morphisms in J). This variation, however, adds no new information. Any collection of objects and morphisms defines a (possibly large) directed graph G, which we describe in volume III, chapter, section 5. If we let J be the *free category* generated by G, there is a universal diagram F: J → C whose image contains G. The limit (or colimit) of this diagram is the same as the limit (or colimit) of the original collection of objects and morphisms.

Weak limits and *weak colimits* are defined like limits and colimits, except that the uniqueness property of the mediating morphism is dropped.

The definition of limits is general enough to subsume several constructions useful in practical settings. In the following we consider the limit (L, φ) of a diagram F: J → C.

Products. If J is a discrete category then a diagram *F* is essentially nothing but a family of objects of C, indexed by J. The limit L of F is called the *product* of these objects. The cone φ consists of a family of morphisms φ_X: L → F(X) called the *projections* of the product. In the category of sets, for instance, the products are given by Cartesian products and the projections are just the natural projections onto the various factors.

Powers. A special case of a product is when the diagram F is a constant functor to an object X of C. The limit of this diagram is called the J^{th} *power* of X and denoted X^J.

Inverse limits. Let J be a directed set (considered as a category by adding arrows i → j if and only if i ≤ j) and let F: J^{op} → C be a diagram. The limit of F is called (confusingly) an *inverse limit* or *projective limit*.

If J = **1**, the category with a single object and morphism, then a diagram of shape J is essentially just an object X of C. A cone to an object X is just a morphism with codomain X. A morphism f: Y → X is a limit of the diagram X if and only if f is an isomorphism. More generally, if J is any category with an initial object i, then any diagram of shape J has a limit, namely any object isomorphic to F(i). Such an isomorphism uniquely determines a universal cone to F.

Topological limits. Limits of functions are a special case of limits of filters, which are related to categorical limits as follows. This relates to chapter 2, section 7, on finished and unfinished sets. Given a topological space X, denote by F the set of filters on X, x ∈ X a point, V(x) ∈ F the neighbourhood filter of x, A ∈ F a particular filter and $F_{x,A} = \{G \in F: V(x) \cup A \subset G\}$ the set of filters finer than A and that converge to x. The filters F are given a thin category structure by adding an arrow A → B if and only if A ⊆ B. The injection $I_{x,A}$ → F becomes a functor and the following equivalence holds:

x is a topological limit of A if and only if A is a categorical limit of $I_{x,A}$.

Examples of colimits are given by the dual versions of the examples above.
Coproducts are colimits of diagrams indexed by discrete categories.
Copowers are colimits of constant diagrams from discrete categories.
Coequalisers are colimits of a parallel pair of morphisms.
Cokernels are coequalisers of a morphism and a parallel zero morphism.
Pushouts are colimits of a pair of morphisms with common domain.
Direct limits are colimits of diagrams indexed by directed sets.

A given diagram F: J → C may or may not have a limit (or colimit) in C. Indeed, there may not even be a cone to F, let alone a universal cone.

A category C is said to *have limits of shape* J if every diagram of shape J has a limit in C. Specifically, a category C is said to *have products* if it has limits of shape J for every discrete category J, *have equalisers* if it has limits of shape $\bullet \rightrightarrows \bullet$, so every parallel pair of morphisms has an equaliser and *have pullbac*ks if it has limits of shape $\bullet \to \bullet \to \bullet$; every pair of morphisms with common codomain has a pullback.

A *complete category* is a category that has limits (all limits of shape J for every category J).

We can also make the dual definitions. A category *has colimits of shape* J if every diagram of shape J has a colimit in C. A *cocomplete* category is one that has colimits.

6.10. Theorems on limits and colimits.

The *existence theorem for limits* states that if a category C has equalisers and all products indexed by the **mSets** Ob(J) and hom_(J), then C has all limits of shape J. In this case, the limit of a diagram F: J → C can be constructed as the equaliser of the two morphisms given (in component form) by

here is a dual *existence theorem for colimits* in terms of coequalisers and coproducts. Both of these theorems give sufficient and necessary conditions for the existence of all (co)limits of shape J.

Limits and colimits are important special cases of universal constructions.

Let C be a category and let J be an index category. The functor category may be thought of as the category of all diagrams of shape J in C. The diagonal functor

is the functor that maps each object N in C to the constant functor Δ(N): J → C to N. That is, Δ(N)(X) = N for each object X in J and Δ(N)(f) = id_N for each morphism f in J.

Given a diagram F: J → C (thought of as an object in C^J), a natural transformation ψ : Δ(N) → F (which is just a morphism in the category C^J) is the same thing as a cone from N to F. To see this, first note that Δ(N)(X) = N for all X implies that the components of ψ are morphisms ψ_X: N → F(X), which all share the domain N. Moreover, the requirement that the cone's diagrams commute is true simply because this ψ is a natural transformation. (Dually, a natural transformation ψ: F → Δ(N) is the same thing as a co-cone from F to N.)

Therefore, the definitions of limits and colimits can then be restated in the form:
A limit of F is a universal morphism from Δ to F.
A colimit of F is a universal morphism from F to Δ.

Like all universal constructions, the formation of limits and colimits is functorial in nature. In other words, if every diagram of shape J has a limit in C (for J small) there exists a *limit functor*

which assigns each diagram its limit and each natural transformation η : F → G the unique morphism lim η : lim F → lim G commuting with the corresponding universal cones. This functor is right adjoint to the diagonal functor Δ: C → C^J. This adjunction gives a bijection between the set of all morphisms from N to lim F and the set of all cones from N to F

which is natural in the variables N and F. The counit of this adjunction is simply the universal cone from lim F to F. If the index category J is connected (and nonempty) then the unit of the adjunction is an isomorphism so that lim is a left inverse of Δ. This fails if J is not connected. For example, if J is a discrete category, the components of the unit are the diagonal morphisms δ : N → N^J.

Dually, if every diagram of shape J has a colimit in C there exists a *colimit functor*

which assigns each diagram its colimit. This functor is left adjoint to the diagonal functor Δ: C → C^J, and one has a natural isomorphism

The unit of this adjunction is the universal cocone from F to colim F. If J is connected (and nonempty) then the counit is an isomorphism, so that colim is a left inverse of Δ.

Note that both the limit and the colimit functors are covariant functors.

Limits and colimits of presheaves

One can use hom functors to relate limits and colimits in a category C to limits in **mSet**. This follows, in part, from the fact the covariant hom functor hom(N, –) : C → **mSet** preserves all limits in C. By duality, the contravariant hom functor must take colimits to limits.

If a diagram F : J → C has a limit in C, denoted by lim F, there is a canonical isomorphism

which is natural in the variable N. Here the functor hom(N, F–) is the composition of the Hom functor Hom(N, –) with F. This isomorphism is the unique one which respects the limiting cones.

One can use the above relationship to define the limit of F in C. The first step is to observe that the limit of the functor hom(N, F–) can be identified with the set of all cones from N to F:

The limiting cone is given by the family of maps π_X: Cone(N, F) → hom(N, FX) where $\pi_X(\psi) = \psi_X$. If one is given an object L of C together with a natural isomorphism Φ: hom(–, L) → Cone(–, F), the object L will be a limit of F with the limiting cone given by $\Phi_L(id_L)$. In fancy language, this amounts to saying that a limit of F is a representation of the functor Cone(–, F) : C → **mSet**.

Dually, if a diagram F : J → C has a colimit in C, denoted colim F, there is a unique canonical isomorphism

which is natural in the variable *N* and respects the colimiting cones. Identifying the limit of hom(F–, N) with the set Cocone(F, N), this relationship can be used to define the colimit of the diagram F as a representation of the functor Cocone(F, –).

Let I be a finite category and J be a small filtered category. For any bifunctor

there is a natural isomorphism

In words, filtered colimits in **mSet** commute with finite limits.

If F : J → C is a diagram in C and G: C → D is a functor then by composition (recall that a diagram is just a functor) one obtains a diagram GF: J → D. A natural question is then: 'How are the limits of GF related to those of F?'

A functor G: C → D induces a map fom Cone(F) to Cone(GF): if Ψ is a cone from N to F then GΨ is a cone from GN to GF. The functor G is said to *preserve the limits of* F if (GL, Gφ) is a limit of GF whenever (L, φ) is a limit of F. (Note that if the limit of F does not exist, then G vacuously preserves the limits of F.)

A functor G is said to *preserve all limits of shape* J if it preserves the limits of all diagrams F: J → C. For example, one can say that G preserves products, equalisers, pullbacks, etc. A *continuous functor* is one that preserves all limits.

One can make analogous definitions for colimits. For instance, a functor G preserves the colimits of F if G(L, φ) is a colimit of GF whenever (L, φ) is a colimit of F. A *cocontinuous functor* is one that preserves all colimits.

If C is a complete category, then, by the above existence theorem for limits, a functor G : C → D is continuous if and only if it preserves products and equalisers. Dually, G is cocontinuous if and only if it preserves coproducts and coequalisers.

An important property of adjoint functors is that every right adjoint functor is continuous and every left adjoint functor is cocontinuous. Since adjoint functors exist in abundance, this gives numerous examples of continuous and cocontinuous functors.

For a given diagram F: J → C and functor G: C → D, if both F and GF have specified limits there is a unique canonical morphism

which respects the corresponding limit cones. The functor G preserves the limits of F if and only this map is an isomorphism. If the categories C and D have all limits of shape J then lim is a functor and the morphisms τ_F form the components of a natural transformation

The functor G preserves all limits of shape J if and only if τ is a natural isomorphism. In this sense, the functor *G* can be said to *commute with limits* (up to a canonical natural isomorphism).

Preservation of limits and colimits is a concept that only applies to covariant functors. For contravariant functors the corresponding notions would be a functor that takes colimits to limits, or one that takes limits to colimits.

A functor G: C → D is said to lift limits for a diagram F: J → C if whenever (L, φ) is a limit of GF there exists a limit (L′, φ′) of F such that G(L′, φ′) = (L, φ). A functor G lifts limits of shape J if it lifts limits for all diagrams of shape J. One can therefore talk about lifting products, equalisers, pullbacks, etc. Finally, one says that G lifts limits if it lifts all limits. There are dual definitions for the lifting of colimits.

A functor G lifts limits uniquely for a diagram F if there is a unique preimage cone (L′, φ′) such that (L′, φ′) is a limit of F and G(L′, φ′) = (L, φ). One can show that G lifts limits uniquely if and only if it lifts limits and is amnestic.

Lifting of limits is clearly related to preservation of limits. If G lifts limits for a diagram F and GF has a limit, then F also has a limit and G preserves the limits of F. It follows that:

If G lifts limits of all shape J and D has all limits of shape J, then C also has all limits of shape J and G preserves these limits.

If G lifts all limits and D is complete, then C is also complete and G is continuous.

The dual statements for colimits are equally valid.

Let F: J → C be a diagram. A functor G: C → D is said to

create limits for F if whenever (L, φ) is a limit of GF there exists a unique cone (L′, φ′) to F such that G(L′, φ′) ≅ (L, φ), and furthermore, this cone is a limit of *F*.

reflect limits for F if each cone to F whose image under G is a limit of GF is already a limit of F.

Dually, one can define creation and reflection of colimits.

The following statements are easily seen to be equivalent:

The functor G creates limits.

The functor G lifts limits uniquely and reflects limits.

There are examples of functors which lift limits uniquely but neither create nor reflect them.

Every representable functor C → **mSet** preserves limits (but not necessarily colimits). In particular, for any object A of C, this is true of the covariant hom functor hom(A,–) : C → **Set**.

The forgetful functor U: **Grp** → **mSet** creates (and preserves) all small limits and filtered colimits. However, U does not preserve coproducts. This situation is typical of algebraic forgetful functors.

The free functor F: **mSet** → **Grp** (which assigns to every set S the free group over S) is left adjoint to forgetful functor U and is therefore cocontinuous. This explains why the free product of two free groups G and H is the free group generated by the disjoint union of the generators of G and H.

The inclusion functor **Ab** → **Grp** creates limits but does not preserve coproducts (the coproduct of two abelian groups being the direct sum).

The forgetful functor **Top** → **mSet** lifts limits and colimits uniquely but creates neither.

Let $\mathbf{Met}_c$ be the category of metric spaces with continuous functions for morphisms. The forgetful functor $\mathbf{Met}_c$ → **mSet** lifts finite limits but does not lift them uniquely.

6.11. Addant and multiplicant categories.

A question is: how far can we go in introducing addition and multiplication as universals for morphisms in categories? Since categories may have noncommutative morphisms but must be associative, it is worth a try. The idea is to use addition for coproducts and multiplication for Cartesian products – more generally products in categories not in **mSet**. With that as an aim, we direct our attention to the case of complex numbers which have multiplication and at the same time may in their twofold coordinates represent a graph for a Cartesian product.

If we look at coproducts, we are assigning addition to them. How do we implement this? Well we have seen in 5.5 that the hom functor is sufficiently versatile to operate for all suoperations. It is not generally associative, but our aim is to describe superstructures, which are general nonassociative categories, even though this section is in our account dealing with categories where nonassociativity does not apply.

As a universal, an object which is a complex number is not general enough. It should be clear that in categories with complex matrix representations, an n-dimensional matrix (we should say of rank n) is sufficiently extended to represent as a universal a sufficiency of subobjects under operations which are additive. It will be seen in what follows that the additive unit, zero, corresponds to an initial object in this classification. The question is, what does the additive relation mean in terms of categories?

Categories are representations of the relative idea in mathematics. We have stated, and we repeat now, that the absolute exists and is important. Relative relations can nevertheless be represented additively by differences. The clear indication is that here we have got precisely what we need. We must develop this rigorously, but this can be done. Relative relations for addant categories (which is the additive category in the rigorous development of this analogy) contain subtraction, which is the additive inverse.

We can represent a complex number by a coordinate in two variables. Addant categories are then represented by a shift up or down, left or right of the complex coordinate system.

For multiplication we must do something similar. The multiplication system, on its own, is represented by a boost of a vector or a rotation of the vector, or equivalently (and this is described with force in the work of Eli Cartan) by corresponding operations on the coordinate system. Further, the multiplicative unit is 1. In the categorical language we are using, this is the terminal object.

Having got the multiplication, how do we represent Cartesian products using this analogy? Whereas addant categories are described by the cograph diagrams of section 2, Cartesian products are represented by a graph. Its generic universal is the familiar (or should be familiar) description of a function in one variable by a pair {a, f(a)}. This twoness is representable by the twoness inherent in the representation of a complex number by a pair of variables. This now describes as an idea how we can implement a multiplicant category. We have to be rigorous and give the details.

Note that when we have a rotation and boost, which is obtained by complex multiplication, and we interpret this in terms of a set of complex numbers in some complex coordinate system, then if the set at some stage represents a function, a sufficient rotation may change the function in the original coordinate system to a multifunction in the rotated coordinate system.

We have represented a continuous multifunction in the diagram in chapter 2, section 18. In terms of horizontal and vertical axes, this diagram could represent a variety. We have been told, and insight should tell us, that the Wonderful Theorem states that any multiobject multifunction can be reallocated to a new possibly extended multiobject which is a monotonic function, and thus has an inverse. When we continuously rotate say the coordinate system and leave the continuous multiobject fixed with respect to the original coordinates in that order, then the multiobject transforms continuously in the new coordinate system to a new multiobject, but the Wonderful Theorem still applies in a new continuously transformed context.

We have so far described addition and multiplication operations as suitable in a category, but we have not introduced situations where these combine together. Addition and multiplication together satisfy the distributive rule for a ring, for complex numbers which are commutative, this is

$$(a + b)c = ac + bc = ca + cb = c(a + b).$$

Suppose we want to complete the algebra so that we have inverses everywhere. We use the binomial theorem for $c = (a+ b)^{-1}$ and have

$$cc^{-1} = 1.$$

Note that in chapter 1, section 12 we have introduced zero algebras, so that this works when we have multizeros (which are multiobjects) as the initial object (but yes, now we have more than one initial object. This can be generic if it is represented by a universal ${}_{a}0$, where a is a universal algebraic object).

All we need now is the binomial theorem for zero algebras. Basically it is there. We do of course have a more complicated representation of $(–b)^a$ for zero algebras, given in chapter 1 section 12.

If we want to stay in fields, we can do that, but note that in this special case we have studied above we are in unreason. This does not really matter. Interestingly if we stipulate that distinct numbers in a ring remain distinct for a field, we have a mapping from zero algebras to fields enabling unreason, and we can track this, but if some deduction paths are in unreason for fields, it is possible that the deduction path is also in unreason for the zero algebra. We deal with general colour deduction threads which colour transform, and this also includes invalid reasoning in Boolean algebra, say of propositional or predicate type, in our discussion of glyphs in volume III, chapter 7.

6.12. Comma categories.

A *comma category* (a special case being a *slice category*) is a construction in category theory. It provides another way of looking at morphisms. Instead of relating objects of a category to one another, morphisms become objects in their own right. We saw this idea in 3.17.

Comma categories also guarantee the existence of some limits and colimits.

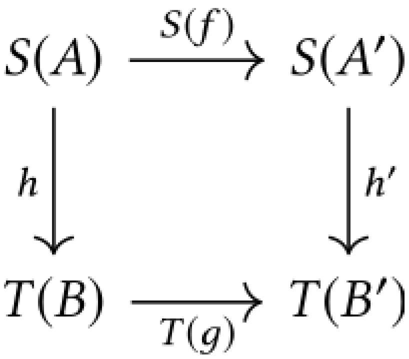
The most general comma category construction involves two functors with the same codomain. Often one of these will have domain **1** (the one-object one-morphism category – then we say we have a slice category). Some accounts of category theory consider only these special cases, but the term comma category is actually much more general.

A *comma category* is a special case of a *slice category*. It provides another way of looking at morphisms: instead of simply relating objects of a category to one another, morphisms become objects in their own right. This notion was introduced in 1963 by F.W. Lawvere, although the technique did not become generally known until many years later. Several mathematical concepts can be treated as comma categories. Comma categories also guarantee the existence of some limits and colimits. The name comes from the notation originally used by Lawvere, which involved the comma punctuation mark. Although standard notation has changed, since the use of a comma as an operator is potentially confusing, and even Lawvere dislikes the uninformative term 'comma category', the name persists

Suppose that , , and are categories, and and (for source and target) are functors:

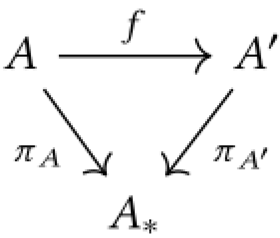
We can form the comma category as follows:

- The objects are all triples with an object in , an object in , and a morphism in .
- The morphisms from to are all pairs where and are morphisms in and respectively, such that the following diagram commutes:



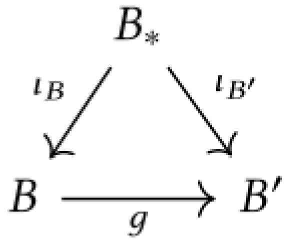
Morphisms are composed by taking to be , whenever the latter expression is defined. The identity morphism on an object is .

The first special case occurs when , the functor is the identity, and (the category with one object and one morphism). Then for some object in . In this case, the comma category is written , and is often called the *slice category* over or the category of *objects over* . The objects can be simplified to pairs , where . Sometimes, is denoted by . A morphism from to in the slice category can then by simplified to an arrow making the following diagram commute:

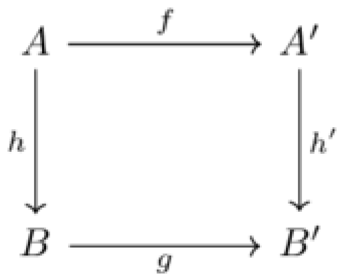


The dual concept to a slice category is a coslice category. Here, has domain **1** and is an identity functor. In this case, the comma category is often written , where is the object of selected by . It is called the *coslice category* with respect to , or the category of *objects under*

. The objects are pairs with . Given and , a morphism in the coslice category is a map making the following diagram commute:



and are identity functors on (so). In this case, the comma category is the arrow category . Its objects are the morphisms of , and its morphisms are commuting squares in .



In the case of the slice or coslice category, the identity functor may be replaced with some other functor; this yields a family of categories particularly useful in the study of adjoint functors. For example, if is the forgetful functor mapping an abelian group to its underlying ser, and is some fixed set (regarded as a functor from **1**), then the comma category has objects that are maps from to a set underlying a group. This relates to the left adjoint of , which is the functor that maps a set to the freeabelian group having that set as its basis. In particular, the initial object of is the canonical injection , where is the free group generated by .

An object of is called a *morphism from to* or a *-structured arrow with domain* in. An object of is called a *morphism from to* or a *-costructured arrow with codomain* in.

Another special case occurs when both and are functors with domain **1**. If and , then the comma category , written , is the discrete category whose objects are morphisms from to .

An inserter category is a (non-full) subcategory of the comma category where and are required. The comma category can also be seen as the inserter of and , where and are the two projection functors out of the product category

For each comma category there are forgetful functors from it.

- Domain functor, , which maps:
    - objects: ;
    - morphisms: ;
- Codomain functor, , which maps:

- objects: ;
    - morphisms: .
- Arrow functor, , which maps:
    - objects: ;
    - morphisms: ;

Several interesting categories have a natural definition in terms of comma categories.

- The category of pointed sets is a comma category, with being (a functor selecting) any singleton set, and (the identity functor of) the category of sets. Each object of this category is a set, together with a function selecting some element of the set: the ‘basepoint’. Morphisms are functions on sets which map basepoints to basepoints. In a similar fashion one can form the category of pointed spaces .
- The category of associative algebras over a ring is the coslice category , since any ring homomorphism induces an associative -algebra structure on , and vice versa. Morphisms are then maps that make the diagram commute.
- The category of graphs is , with the functor taking a set to . The objects then consist of two sets and a function; is an indexing set, is a set of nodes, and chooses pairs of elements of for each input from . That is, picks out certain edges from the set of possible edges. A morphism in this category is made up of two functions, one on the indexing set and one on the node set. They must "agree" according to the general definition above, meaning that must satisfy . In other words, the edge corresponding to a certain element of the indexing set, when translated, must be the same as the edge for the translated index.
- Many ‘augmentation’ or ‘labelling’ operations can be expressed in terms of comma categories. Let be the functor taking each graph to the set of its edges, and let be (a functor selecting) some particular set: then is the category of graphs whose edges are labelled by elements of . This form of comma category is often called *objects -over -* closely related to the ‘objects over ’ discussed above. Here, each object takes the form , where is a graph and a function from the edges of to . The nodes of the graph could be labelled in essentially the same way.
- A category is said to be *locally cartesian closed* if every slice of it is cartesian closed (see above for the notion of *slice*). Locally cartesian closed categories are the classifying categories of dependent type theories.

Limits and universal morphisms

Limits and colimits in comma categories may be ‘inherited’. If and are complete, is a continuous functor, and is another functor (not necessarily continuous), then the comma category produced is complete, and the projection functors and are continuous. Similarly, if and are cocomplete, and is cocontinuous cocontinuous, then is cocomplete, and the projection functors are cocontinuous.

For example, note that in the above construction of the category of graphs as a comma category, the category of sets is complete and cocomplete, and the identity functor is continuous and cocontinuous. Thus, the category of graphs is complete and cocomplete.

The notion of a universal morphism to a particular colimit, or from a limit, can be expressed in terms of a comma category. Essentially, we create a category whose objects are cones, and where the limiting cone is a terminal object; then, each universal morphism for the limit is just the morphism to the terminal object. This works in the dual case, with a category of cocones having an initial object. For example, let be a category with the functor taking each object to and each arrow to . A universal morphism from to consists, by definition, of an object and morphism with the universal property that for any morphism there is a unique morphism with . In other words, it is an object in the comma category having a morphism to any other object in that category; it is initial. This serves to define the coporduct in , when it exists.

Adjunctions
Lawvere showed that the functors and are adjoint if and only if the comma categories and , with and the identity functors on and respectively, are isomorphic, and equivalent elements in the comma category can be projected onto the same element of . This allows adjunctions to be described without involving sets, and was in fact the original motivation for introducing comma categories.

Natural transformations If the domains of are equal, then the diagram which defines morphisms in with is identical to the diagram which defines a natural transformation . The difference between the two notions is that a natural transformation is a particular collection of morphisms of type of the form , while objects of the comma category contains *all* morphisms of type of such form. A functor to the comma category selects that particular collection of morphisms. This is described succinctly by an observation by S.A. Huq that a natural transformation , with , corresponds to a functor which maps each object to and maps each morphism to . This is a bijective correspondence between natural transformations and functors which are sections of both forgetful functors from .

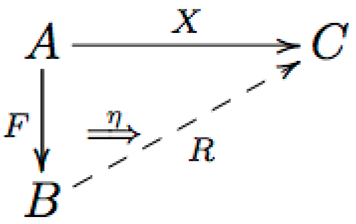
6.13. Kan extensions.

Kan extensions are universal constructs in category theory. They are closely related to adjoints, but are also related to limits and ends. They are named after Daniel M. Kan, who constructed certain (Kan) extensions using limits in 1960.

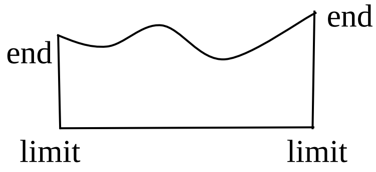
Integration and differentiation of ordered groups can be introduced as Kan extensions. Since we can differentiate and integrate exponentials, it follows that we can perform the same operations on the number of combinations of arrows, or even, it turns out, on the arrows directly, via mappings from the combinations of arrows to the arrows themselves. This theory is within category theory and uses hom-sets rather than general exponentials.

Kan extensions generalise the notion of extending a function defined on a subset to a function defined on the whole set. The definition, not surprisingly, is at a high level of abstraction. When specialised to posets it becomes a type of question on constrained optimisation.

A Kan extension proceeds from the data of three categories and two functors, and comes in two varieties: the *left* Kan extension and the *right* Kan extension of X along F. It amounts to finding the dashed arrow and the 2-cell in the following diagram:

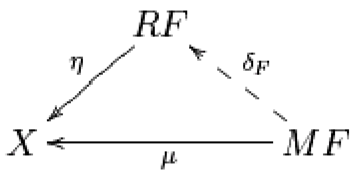


(The natural transformation in the above depiction of the right Kan extension points to the functors. However, it should be interpreted as an arrow to the functor from the composed functor.)



For instance, the right Kan extension can relate the area under the curve (the integral) to the limits (or ends).

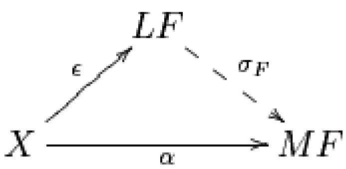
Formally, the *right Kan extension of* X *along* F consists of a functor $R: B \to C$ and a natural transformation $\eta: RF \to X$ which is couniversal with respect to the specification, in the sense that for any functor and natural transformation $\mu: MF \to X$, a unique natural transformation $\delta: M \to R$ is defined and fits into a commutative diagram



(where δ_F is the natural transformation with $\delta_F(a) = \delta(Fa): MF(a) \to RF(a)$ for any object a of A).

The functor R is often written $Ran_F X$.

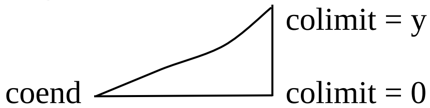
As with the other universal constructs in category theory, the left version of the Kan extension is dual to the right one and is obtained by replacing all categories by their opposites. The effect of this on the description above is merely to reverse the direction of the natural transformations. This gives rise to the alternate description: the *left Kan extension of* X *along* F consists of a functor $L: B \to C$ and a natural transformation $\varepsilon: X \to LF$ which are universal with respect to this specification, in the sense that for any other functor $M: B \to C$, and natural transformation $\alpha: X \to MF$, a unique natural transformation $\sigma: L \to M$ exists and fits into a commutative diagram:



(where σ_F is the natural transformation with $\sigma_F(a) = \sigma(Fa): LF(a) \to MF(a)$ for any object a of A).

The functor L is often written $Lan_F X$.

To give the left Kan dual example of the integral right Kan, we have differentiation



The use of the word *the* (as in the left Kan extension) is justified by, as with all universal constructions, if the object defined exists, then it is unique up to unique isomorphism. In this case, that means that (for left Kan extensions) if L, M are two left Kan extensions of X along F, and ε, α are the corresponding transformations, then there exists a unique *isomorphism* of functors $\sigma: L \to M$ such that the second diagram above commutes. Likewise for right Kan extensions.

Those readers wanting more information on adjoint functors and Kan extensions are referred to [ML98] and chapter 5. Some category theorists do not distinguish between hom-sets and exponentials.

Thus we see that we can express a large number of mathematical concepts using category theory. Although it may be interesting to compare categorical proofs with other methods, in what follows we will often avoid such an approach. Category theory may be used to ask questions about foundations, but its cumbersome notation is inappropriate to find answers. When these answers are found, it may be possible to retrofit them into a categorical framework, where meanings have to be explicitly given by examples. This indicates there is something wrong with the categorical idea, specifically in the way it can be applied. It is much easier to work with universal examples than category theory itself. Only when we become acclimatised to the idea that a universal example is the same as the categorical representation does the pursuit of this approach gives productive results.

Because category theory requires effort to come to grips with its notation and the meaning of its abstractions, if this language is not assimilated, the expression of ideas in it becomes alienating. To put it another way, the notational inconveniences of category theory give rise to abstract confusion theory and the second incomprehensibility theorem. Confusion is described by a map from a correct line of reasoning to an incorrect one. Its dual map is understanding. We do not develop these ideas further, but note that advanced research into metaconfusion, a confused way of thinking about confusion, has been extensively applied in sociology.

6.14. Multilimits and comultilimits.

Example 6.14.1. Is there is any universal example for the Dedekind-MacNeille construction of chapter 2, section 6? We are converting from multiuniversals to universals here. To answer this we first review some things we said there about lattices. Lattices have implementations under + (least upper bound = join, greatest lower bound = meet), where the distributive laws

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

and

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

hold, and under × (least common multiple = join, greatest common factor = meet). In this latter case, as we have shown in chapter II, section 5, the distributive laws do not always hold. As we showed there, the only lattice for a ring, which has + and ×, is a dependent probability logic with probability P(a) = a, with join $a \vee b = a + b - ab$, unless we are looking at $a \vee a = a$ and for meet a ∧ b = ab, unless we have a ∧ a = a. In sets the distributive laws hold, so at most if we have + and ×, the model of it must be a dependent probability logic if we are using sets.

If we look at the example diagram given in chapter 2, section 6, describing the Dedekind-MacNeille construction, this is not a lattice but a multilattice. In order to consider whether a universal example exists for this construction, we must extend the ideas of functions, which we have employed in the Yonda embedding and Yoneda lemma, to multifunctions. We have done this at the end of chapter II, section 18. We replace a multifunction f from a set x to f(x) by a bijective mapping from the multiobject bouncing set B(x), forming a partition of the domain of x, to the codomain f(B(X)).

To a multilattice with more than one meet and join for a and b, we apply the Yoneda embedding using a function on bouncing sets. This provides a universal for the multilattice in terms of sets and thus for a dependent probability ring with P(a) = a. The Yoneda embedding is a universal example for the categorical Yoneda lemma, so there is a categorical universal for multilattices. We could show this directly, since multilattices are implemented as posets representable by arrows, their combinations by graphs, and graphs are representable by categories.

The Dedekind-MacNeille construction appends nodes to a multilattice to form a lattice. The construction kills all multilattice instances. Since a lattice can be implemented as a dependent probability ring directly, by the Yoneda lemma this construction provides a universal in **mSet**, and again via the Yoneda lemma, as a category. ☐

6.15. The Wonderful Theorem (WT).

6.16. The meaning and ideas of superstructure theory.

Transformations arising in category theory are known as *morphisms*, but in the case of morphisms these are associative mappings, that is, for three morphisms r, s and t
r(st) = (rs)t.

We will generalise this idea to include morphisms which are not associative. We will define a generalised category with nonassociative morphisms as a *superstructure*. This idea will be introduced properly in chapter 5.

To distinguish a superstructure and a category, we introduce some modifications to names.

Definition 6.16.1. A *superstructure*, which is nonassociative, consists of suobjects or sunodes a, b, c, ..., suarrows f, g, h, ..., and two operations
Domain, which provides each arrow f with an object a = dom f
Codomain, providing each arrow f with an object b = cod f.
These satisfy
Identity, which assigns a suarrow $id_a = 1_a$ to each suobject a,
Composition, assigning to each pair of suarrows <g, f> with dom g = cod f, a suarrow gf, called their composite,
with the property
Unit law. For all suarrows f, a → b and g, b → c
f_1_b = f and 1_b_g = g,
so that the identity suarrow 1_b for an object b acts as an identity operation for composition.

If it unclear from the context whether, say an identity, belongs to a superstructure or a category, we may prefix a name describing an aspect of a superstructure by su, otherwise retaining the name used for a category.

For mathematical structures derived from previously known ones, but dropping the associative rule, we prefix them by su. So we use the superstructure **suGrp** for sugroups, the superstructure **suAb** of all suabelian groups, **suMon** for the superstructure of sumonoids, etc.

We will generalise this idea to include morphisms which are not associative. We will define a generalised category with nonassociative morphisms as a *superstructure*.

6.17. Toposes and sutoposes.

Sets, which are particular types of mathematical objects, can be given a general categorical description in terms of mappings, called a topos, and suitable toposes can replace the category of sets as a foundation for mathematics. They were originally introduced from the consideration of sheaves, which are general sets having local differential structure defined in terms of open sets, and were used by Grothendieck in an attack on the Weil conjectures. They were later put on an even more general categorical footing by Lawvere and Tierney.

Definition 6.17.1. A *presheaf* on a category C is a functor F: $C^{op} \to$ **mSet**.

A *sheaf*, usually denoted by F from the French ‘faisceau’ with the same meaning, describes a class of functions on a topological space X. These may be continuous or differentiable. The sheaf is described locally in open neighbourhoods of each point $x \in X$ in terms of functions F_x defined in these neighbourhoods. For all x, the sets F_x can be pasted together, so that F_x varies with variations of x in a continuous or differentiable way.

For example, if we were to choose continuity as the criterion, then this pasting together of open neighbourhoods U in X can be defined locally by the properties of functions f: $U \to X$

(i) (*Locality*) If f: $U \to X$ is continuous and $V \subset U$ is open, then the function f restricted to V is continuous.
(ii) (*Gluing*) If U is covered by open sets U_i, and the functions f_i: $U_i \to X$ are continuous for all $i \in I$, where I is an index set, then there is at most one continuous f: $U \to X$ with restrictions to U_i for all i. This f exists if and only if the overlaps match, so that for every $x \in U_i \cap U_j$, $f_i(x) = f_j(x)$.

Let C(U) be the function which assigns each open $U \subset X$ the set of all continuous functions on U. We have seen that if inclusion is covariant, then restriction is contravariant. If $\mathcal{O}(X)$ is the category with objects all open sets U of X, and arrows $V \to U$ the inclusions $V \subset U$, then this means that the assignments

$$U \to C(U), \text{ and } \{V \subset U\} \to \{C(U) \to C(V)\} \text{ by restriction} \quad (10)$$

define a functor C: $\mathcal{O}(X)^{op} \to$ **mSet**.

We can rephrase this as saying that a sheaf is a presheaf on a topological space satisfying the locality rule (i) and the gluing rule (ii).

Concerning property (ii), for an open covering $U = \cup U_i$ and an i indexed family of functions f_i: $U_i \to X$, then i is a member of the product set $\prod_i C(U_i)$, whilst the assignments $\{f_i\} \to \{f_i\}$ restricted to $U_i \cap U_j$ and $\{f_j\} \to \{f_j\}$ to $U_i \cap U_j$ define two maps, p and q, of I indexed sets to $\{I \times I\}$ indexed sets, given in the equaliser diagram

$$\mathrm{C(U)} \overset{e}{\rightarrow} \prod_i \mathrm{C(U_i)} \underset{q}{\overset{p}{\rightrightarrows}} \prod_{i,j} \mathrm{C(U_i \cap U_j)}. \tag{11}$$

Here e is the universal map which is the equaliser of the maps p and q.

The axiom of consistent choice holds for sets in mZFC (the consistency requirement here is sometimes described by the statement that the sets are well-pointed) and the natural numbers given by the Peano axioms satisfy their properties. Additional to these items is the topos idea.

In order to define the notion of a topos, we need to define in categorical terms how we assign the values true and false to statements, and their mappings to sets or toposes. So that we can be sufficiently general, and be able to describe branched spaces in terms of probability logic and the multivalued logics of volume II, we now proceed to give an account of this.

The simplest example is when there are only two values, as in Boolean logic with true and false. Then the *characteristic function* of a subset $S \subseteq X$ is a function $\varphi_S\colon X \to \{0, 1\}$ on X with values

$$\varphi_S(x) = 0 \text{ if } x \in S, \text{ otherwise } \varphi_S(x) = 1 \text{ if } x \notin S. \tag{12}$$

This may be expressed in the diagram

$$\begin{array}{ccc} S & \rightarrow & 1 \\ m\big\downarrow & & \big\downarrow t \\ X & \overset{\varphi}{\rightarrow} & \Gamma \end{array} \tag{13}$$

where the top horizontal map is unique map to the object 1, m and t are monomorphisms, and the set Γ contains in this instance the values {0, 1}.

To generalise, the set Γ could contain, say {0, 1, 2}, etc., or be any arbitrary set. The map t is the *subobject classifier* for the objects in this category.

A *topos* has at least the following properties

(i) It has a subobject classifier.
(ii) It is always possible to form from it a finite number of Cartesian products, equivalently of ordered sequences, or 'finite limits'
(iii) The category in which it resides is described by hom-sets (which are associative).

With successive restriction, let Y be a zargon subox (or real or transnatural) ladder number given in 2.3.

A *sutopos* has at least the following properties

(i) It has a subobject classifier ∈ Y.
(ii) It is always possible to form from it a number y ∈ Y of Cartesian products, equivalently of ordered sequences, or 'y limits'
(iii) The category in which it resides has the number of its objects and arrows described by superstructures (which are nonassociative) with coefficients in Y.