

Category Theory

A Gentle Introduction

Peter Smith
University of Cambridge

Version of January 29, 2018

©Peter Smith, 2018

This PDF is an early incomplete version of work still very much in progress. For the latest and most complete version of this Gentle Introduction and for related materials see the [Category Theory page](#) at the Logic Matters website.

Corrections, please, to ps218 at cam dot ac dot uk.

Contents

Preface	ix
1 The categorial imperative	1
1.1 Why category theory?	1
1.2 From a bird's eye view	2
1.3 Ascending to the categorial heights	3
2 One structured family of structures	4
2.1 Groups	4
2.2 Group homomorphisms and isomorphisms	5
2.3 New groups from old	8
2.4 'Identity up to isomorphism'	11
2.5 Groups and sets	13
2.6 An unresolved tension	16
3 Categories defined	17
3.1 The very idea of a category	17
3.2 Monoids and pre-ordered collections	20
3.3 Some rather sparse categories	21
3.4 More categories	23
3.5 The category of sets	24
3.6 Yet more examples	26
3.7 Diagrams	27
4 Categories beget categories	30
4.1 Duality	30
4.2 Subcategories, product and quotient categories	31
4.3 Arrow categories and slice categories	33
5 Kinds of arrows	37
5.1 Monomorphisms, epimorphisms	37
5.2 Inverses	39
5.3 Isomorphisms	42
5.4 Isomorphic objects	44
	iii

Contents

6	Initial and terminal objects	46
6.1	Initial and terminal objects, definitions and examples	47
6.2	Uniqueness up to unique isomorphism	48
6.3	Elements and generalized elements	49
7	Products introduced	51
7.1	Real pairs, virtual pairs	51
7.2	Pairing schemes	52
7.3	Binary products, categorially	56
7.4	Products as terminal objects	59
7.5	Uniqueness up to unique isomorphism	60
7.6	‘Universal mapping properties’	62
7.7	Coproducts	62
8	Products explored	66
8.1	More properties of binary products	66
8.2	And two more results	67
8.3	More on mediating arrows	69
8.4	Maps between two products	71
8.5	Finite products more generally	73
8.6	Infinite products	75
9	Equalizers	76
9.1	Equalizers	76
9.2	Uniqueness again	79
9.3	Co-equalizers	80
10	Limits and colimits defined	83
10.1	Cones over diagrams	83
10.2	Defining limit cones	85
10.3	Limit cones as terminal objects	87
10.4	Results about limits	88
10.5	Colimits defined	90
10.6	Pullbacks	91
10.7	Pushouts	94
11	The existence of limits	96
11.1	Pullbacks, products and equalizers related	96
11.2	Categories with all finite limits	100
11.3	Infinite limits	102
11.4	Dualizing again	103
12	Subobjects	104
12.1	Subsets revisited	104
12.2	Subobjects as monic arrows	105

12.3	Subobjects as isomorphism classes	106
12.4	Subobjects, equalizers, and pullbacks	107
12.5	Elements and subobjects	109
13	Exponentials	110
13.1	Two-place functions	110
13.2	Exponentials defined	111
13.3	Examples of exponentials	113
13.4	Exponentials are unique	116
13.5	Further results about exponentials	117
13.6	Cartesian closed categories	119
14	Group objects, natural number objects	123
14.1	Groups in Set	123
14.2	Groups in other categories	125
14.3	A very little more on groups	127
14.4	Natural numbers	128
14.5	The Peano postulates revisited	129
14.6	More on recursion	131
15	Functors introduced	135
15.1	Functors defined	135
15.2	Some elementary examples of functors	136
15.3	What do functors preserve and reflect?	138
15.4	Faithful, full, and essentially surjective functors	140
15.5	A functor from Set to Mon	142
15.6	Products, exponentials, and functors	143
15.7	An example from algebraic topology	145
15.8	Covariant vs contravariant functors	147
16	Categories of categories	149
16.1	Functors compose	149
16.2	Categories of categories	150
16.3	A universal category?	151
16.4	‘Small’ and ‘locally small’ categories	152
16.5	Isomorphisms between categories	154
16.6	An aside: other definitions of categories	156
17	Functors and limits	159
17.1	Diagrams redefined as functors	159
17.2	Preserving limits	160
17.3	Reflecting limits	164
17.4	Creating limits	166
18	Hom-functors	167

Contents

18.1	Hom-sets	167
18.2	Hom-functors	169
18.3	Hom-functors preserve limits	170
19	Functors and comma categories	174
19.1	Functors and slice categories	174
19.2	Comma categories	175
19.3	Two (already familiar) types of comma category	176
19.4	Another (new) type of comma category	177
19.5	An application: free monoids again	178
19.6	A theorem on comma categories and limits	180
20	Natural isomorphisms	182
20.1	Natural isomorphisms between functors defined	182
20.2	Why ‘natural’?	183
20.3	More examples of natural isomorphisms	186
20.4	Natural/unnatural isomorphisms between objects	191
20.5	An ‘Eilenberg/Mac Lane Thesis’?	193
21	Natural transformations	195
21.1	Natural transformations	195
21.2	Vertical composition of natural transformations	198
21.3	Horizontal composition of natural transformations	199
22	Functor categories	202
22.1	Functor categories defined	202
22.2	Functor categories and natural isomorphisms	203
22.3	Hom-functors from functor categories	204
22.4	Evaluation and diagonal functors	205
22.5	Cones as natural transformations	206
22.6	Limit functors	207
23	Equivalent categories	210
23.1	The categories \mathbf{Pfn} and \mathbf{Set}_* are ‘equivalent’	210
23.2	\mathbf{Pfn} and \mathbf{Set}_* are not isomorphic	212
23.3	Equivalent categories	213
23.4	Skeletons and evil	216
24	The Yoneda embedding	219
24.1	Natural transformations between hom-functors	219
24.2	The Restricted Yoneda Lemma	222
24.3	The Yoneda embedding	223
24.4	Yoneda meets Cayley	225
25	The Yoneda Lemma	229

25.1	Towards the full Yoneda Lemma	229
25.2	The generalizing move	230
25.3	Making it all natural	231
25.4	Putting everything together	233
25.5	A brief afterword on ‘presheaves’	234
26	Representables and universal elements	235
26.1	Isomorphic functors preserve the same limits	235
26.2	Representable functors	236
26.3	A first example	237
26.4	More examples of representables	239
26.5	Universal elements	240
26.6	Categories of elements	242
26.7	Limits and exponentials as universal elements	244
27	Galois connections	245
27.1	(Probably unnecessary) reminders about posets	245
27.2	An introductory example	246
27.3	Galois connections defined	248
27.4	Galois connections re-defined	251
27.5	Some basic results about Galois connections	252
27.6	Fixed points, isomorphisms, and closures	253
27.7	One way a Galois connection can arise	255
27.8	Syntax and semantics briefly revisited	255
28	Adjoints introduced	257
28.1	Adjoint functors: a first definition	257
28.2	Examples	259
28.3	Naturality	263
28.4	An alternative definition	264
28.5	Adjoints and equivalent categories	269
29	Adjoints further explored	272
29.1	Adjunctions reviewed	272
29.2	Two more theorems!	273
29.3	Adjunctions compose	273
29.4	The uniqueness of adjoints	275
29.5	How left adjoints can be defined in terms of right adjoints	276
29.6	Another way of getting new adjunctions from old	280
30	Adjoint functors and limits	282
30.1	Limit functors as adjoints	282
30.2	Right adjoints preserve limits	284
30.3	Some examples	286
30.4	The Adjoint Functor Theorems	287

Contents

Bibliography

290

Preface

The project This Gentle Introduction is very much still work in progress, so there are chapters at different levels of development and with different degrees of integration with what's around them. So far, at least in a rough and ready way, we cover the basic notions of elementary category theory – explaining the very idea of a category, then treating limits, functors, natural transformations, representables, adjunctions. The long-term plan is (possibly) to say something about categorial logic, explore categories of sets, and even edge towards some initial themes in topos theory.

But considerations of length will soon begin to weigh, because we do take things pretty slowly. Experience suggests that getting a really secure understanding by going at a rather gentle pace when first encountering categorial ways of thinking makes later adventures exploring beyond the basics very much more manageable.

I imagine one reader to be a mathematics student who wants a clear introduction to categorial ideas without having to take on an industrial-strength graduate course (or else who wants a helping hand while tackling the beginnings of such a course). Another reader might be a philosopher interested in the foundations of mathematics (and knowing a smidgin of mathematics) who wants to know what the categorial fuss is about.

What do you need to bring to the party? You obviously can't be well placed to appreciate how category theory gives us a story about the ways in which different parts of modern abstract mathematics hang together if you really know *nothing* beforehand about modern mathematics!

But don't be scared off. In this Gentle Introduction we try to presuppose a bare minimum. If you know just a little e.g. about what a group is, what a Boolean algebra is, what a topological space is, and some similar bits and pieces, then you should cope fairly easily. And if a few later illustrative examples pass you by, don't panic. I usually try to give multiple illustrations of important concepts and constructs; so feel free simply to skip those examples that happen not to work so well for you.

Theorems as exercises There are currently no exercises in what follows – or at least, there are none explicitly labeled as such. However, almost all the proofs of

Preface

theorems in basic category theory are very straightforward. Surprisingly often, you just have to ‘do the obvious thing’: there’s little ingenious trickery needed at the outset. So you can think of almost every theorem as in fact presenting you with an exercise which you could, even should, attempt in order to fix ideas. The ensuing proof which I spell out is then the answer (or at least, *an* answer) to the exercise. For a few tougher theorems, I give preliminary hints about how the argument ought to go.

Notation and terminology I try to keep to settled notation and terminology, and where there are standard alternatives often mention them too: what I say here should therefore be easy to relate to other modern discussions of the same material.

‘Iff’, as usual, abbreviates ‘if and only if’. In addition to using the familiar ‘ \square ’ as an end-of-proof marker, I use ‘ \triangle ’ as an end-of-definition marker.

On the current version These notes are really in three main parts that are not ideally integrated (I’ve had to hang fire, with everything in an unsatisfactory and unfinished state, while I complete another book project). The Preface and Chapters 1 and 3 have been rewritten. Chapter 2 is all new. Then Chapters 4 to 27 are from a version now two years old; and the end of the Gentle Introduction from Chapter 28 was written maybe two years before that. The joins will most certainly show! I hope to get back to this project by the end of 2018.

Thanks! Andrew Bacon, Malcolm F. Lowe and Mariusz Stopa very kindly sent embarrassingly long lists of corrections to the previous version. A lot of the mistakes were obvious typos, but there were also enough mislabelled arrows or fumbling of notation mid-proof and the like that I should certainly apologize to readers who found themselves scratching their heads in puzzlement! I had further corrections from David Ozonoff, Zoltán Tóth, and Adrian Yee. Warm thanks to everyone!

1 The categorial imperative

1.1 Why category theory?

Modern pure mathematics explores abstract structures and their interrelationships (that's not the whole story, of course, but it is unquestionably one central part of the story). These mathematical structures cluster in families. Take such a family of structures together with the structure-preserving maps between them. Then – and here's a fundamental insight – we can think of this family as forming a *further* structure, a structure-of-structures, something else to explore mathematically.

Take a standard example. Any particular group is a structure which comprises some objects equipped with a binary operation defined on them, where the operation obeys familiar axioms. But we can also think of a whole family of groups, together with appropriate maps between them – namely the homomorphisms which preserve group structure – as forming a further structure-of-structures. We will fill out this very abstract sketch a little in the next chapter.

Here's another example, described for now at the same level of arm-waving generality. Take some particular objects equipped with e.g. a partial order: these constitute a simple mathematical structure. Change the objects and/or the partial ordering and we get more such structures. So now we can get a family of such partially-ordered structures together with order-preserving maps between them. We can then think of these as now constituting a second kind of structure-of-structures.

A third example. Any particular topological space is a structure, this time comprising some points equipped with a topology. But again, a whole family of topological spaces, together with appropriate maps between them – this time, the continuous functions which preserve topological structure – forms another structure-of-structures.

In each of these three cases, then, we can not only investigate particular basic structures (particular groups, particular ordered objects, particular topological spaces), but we can *also* explore structures-of-structures (structured families of groups, structured families of partially ordered collections of objects, and families of topological spaces). Moreover, as a further step, we will want to go on to consider the interrelations between these various structures-of-structures.

The categorial imperative

That will involve considering another level of structure-preserving maps, so-called functors, linking these structures-of-structures.

That's not the end of it. Going up another level of abstraction, we will find ourselves wanting to talk e.g. about operations which map one functor to another while preserving their functorial character (in ways we will explain).

So here is one mathematical imperative: to explore these layers of increasing abstraction (a task which will appeal to a certain cast of mind – though certainly not to all mathematicians!). And evidently, if we *are* going to set out on such an exploration, we will want a framework for dealing with them in a disciplined and illuminating way. Category theory provides exactly what we need, at least as we first set out.

This isn't what category theory was specifically designed for (it emerged from much more specific explorations in the area of homological algebra and algebraic topology). However, category theory's basic ideas and constructions do have very wide application and provide a general toolkit for systematically probing structures-of-structures and even structures-of-structures-of-structures. And it is category theory in this general role that will be our main concern in these notes, together with its connections (eventually) with two other disciplines with different kinds of generality, logic and set theory.

1.2 From a bird's eye view

But what do we gain by ascending through these levels of abstraction and by developing tools for imposing some order on what we find?

For a start, we should get a richer conceptual understanding of how various parts of mathematics relate to each other. We might even say that, in *one* sense of that contested label, this will be a 'philosophical' gain. Many philosophers, pressed for a crisp characterization of their discipline, like to quote a famous remark by Wilfrid Sellars,

The aim of philosophy, abstractly formulated, is to understand how things in the broadest possible sense of the term hang together in the broadest possible sense of the term. (Sellars, 1963, p. 1)

Category theory indeed provides us with a unifying framework for exploring in depth some ways in which a lot of mathematics hangs together. That's why it should be of central interest both to mathematicians interested in the conceptual shape of their discipline and also to philosophers of mathematics.

But note, category theory does much more than give us a good way of relating aspects of structures that we already know about. As Tom Leinster so nicely puts it, the theory

... takes a bird's eye view of mathematics. From high in the sky, details become invisible, but we can spot patterns that were impossible to detect from ground level. (Leinster, 2014, p. 1)

From its highly abstract vantage point, category theory crucially reveals *new* connections we hadn't made before. So-called adjunctions are a prime example, as we will see.

Making new connections in turn enables new mathematical discoveries. And it was because of the depth and richness of the resulting discoveries in e.g. modern algebra and topology that category theory first came to prominence. However, as we said, here we stick to more elementary concerns, with an emphasis on unification and conceptual clarification. This way, we can keep everything as accessible as possible.

1.3 Ascending to the categorial heights

The gadgets of basic category theory do fit together rather beautifully in multiple ways. These intricate interconnections mean, however, that there isn't a single best route into the theory. Different treatments can quite appropriately take topics in very different orders, all illuminating in their various ways.

This Gentle Introduction, though, follows perhaps the simplest plan. We begin at ground-level, first talking about *categories* – many paradigm cases are indeed structured-families-of-structures – and we develop ways of describing what happens inside a category. In this new setting, we revisit e.g. the very familiar ideas of forming new structures within a family by taking products or taking quotients, etc.

Only after some extended exploration of categories taken singly do we then move up a level to consider *functors*, maps between categories.

And then, only after we have spent a number of chapters thinking about how particular functors work (and how they interact with products, quotients and the like), do we move up another level to define operations sending one functor to another – these are the so-called *natural transformations* and *natural isomorphisms*. We then explore these notions, and the related idea of one functor being a *representation* of another, at some length before we at last start exploring the key notion of *adjunctions*.

Finally, having climbed to these heights, we hope to reflect some more about logic and sets.

In short, then, our route into the basics of category theory steadily ascends through the increasing levels of abstraction in a particularly natural way. True, this does mean that we take rather a long time to reach the *really* novel and exciting categorial ideas.¹ However, I think that this disadvantage is considerably outweighed by the real gain in secure understanding which comes from taking our gently sloping ascent towards the categorial heights. I will just have to do my best to make the views we glimpse along the way seem interesting enough!

¹Logicians already have a quite different use for 'categorial'. So when talking about categories, I much prefer the adjectival form 'categorial', even though it is the minority usage.

2 One structured family of structures

We said that category theory is a framework in which we can think systematically about structured families of mathematical structures: or at least, it is this aspect of the theory which is going to be our focus. And as we noted, a paradigm case of such a structured family comprises groups organized by the homomorphisms between them.

It will be useful in this preliminary chapter to briskly review some features of this example. We will then find ourselves tangling with some general issues about sets. We proceed quite informally (and inconclusively!).

2.1 Groups

(a) Here then is one definition of a group (you will probably notice straight away that in one respect this is not quite the usual definition, but we'll return to say something about its mildly deviant character in §2.5):

Definition 1. A *group* is some objects G equipped with a binary operation $*$ (which sends any objects x, y among G to an object $x * y$ also among G) and with a distinguished object e , the group identity, such that

- (i) for any x, y, z among G , $(x * y) * z = x * (y * z)$,
- (ii) for any x among G , $x * e = x = e * x$,
- (iii) for any x among G , there is some y also among G such that $x * y = e = y * x$. △

Elementary examples of groups will be very familiar. There are trivial cases, such as the one object group (a single object e , whatever you like, equipped with the only possible binary operation $*$ such that $e * e = e$) and the two object case (objects d, e , again whatever you like, with e the identity and $d * d = e$). Then there are, for example, additive groups of numbers (e.g. the integers equipped with addition or with addition mod n , with zero as the identity). These examples are abelian, i.e. the binary operation is commutative.

Then there are groups of functions. For a simple case, take the group of permutations of the first n naturals, with functional composition as the group

2.2 Group homomorphisms and isomorphisms

operation and the do-nothing permutation as the group identity. If $n > 2$, then this permutation group is non-abelian.

For another sort of example, take groups of geometrical transformations – for instance the group of symmetries of a regular polygon (i.e. the rotation and reflection operations which map the polygon to itself). Then there are various groups of matrices, groups of closed paths in a topological space, And so on it goes. Groups are indeed many and various!

(b) Let's fix some notation:

We will, for now, use ' $(G, *, e)$ ' simply to abbreviate 'the objects G equipped with the operation $*$ and with distinguished object e ': similarly, of course, for ' (H, \star, d) ' etc. And when convenient we will abbreviate such expressions further by ' \mathcal{G} ', ' \mathcal{H} ', etc.

If $(G, *, e)$ satisfy the conditions for forming a group, then we briskly write 'the group $(G, *, e)$ ' (or simply 'the group \mathcal{G} ') rather than 'the group consisting in $(G, *, e)$ '.

And three elementary observations for future use. First, given a group $(G, *, e)$, it is immediate from the axioms that for each group object x there is a unique y such $x * y = e = y * x$. Hence the axioms for a group implicitly define a function $inv: G \rightarrow G$ which sends an object to its unique group inverse.

Second, picking out a distinguished object e to act as the group identity can alternatively be thought of as equipping G with a function, i.e. a nullary function which takes no input and outputs e .

So, third, we could just as well have defined a group as comprising objects equipped with three functions (one binary, one unary, one nullary) satisfying certain axioms.

2.2 Group homomorphisms and isomorphisms

(a) To impose some order on the multiplicity of groups, we will be interested in the interrelations between groups, as traced out by maps which preserve group structure. Here's how to define such maps:

Definition 2. A *group homomorphism* from the group $(G, *, e)$ as source to the group $(G', *', e')$ as target is a function f defined over the objects G with values among G' such that:

(i) for every x, y among G , $f(x * y) = f x *' f y$,

(ii) $f e = e'$. △

Thought of simply as mapping objects to objects, the function we can symbolize as $f: G \rightarrow G'$ is conventionally said to be the *underlying function* of the

One structured family of structures

homomorphism. When thought of as a homomorphism between groups, we will symbolize the function as $f: (G, *, e) \rightarrow (G', *, e')$.

For the moment, we will take just three very simple examples:

- (1) Let $(G, *, e)$ form a group. Then there is a homomorphism $f: (G, *, e) \rightarrow \mathbf{1}$, where $\mathbf{1}$ is any one-object group. Just let f send every object among G to the sole object of the target group.

This trivial case reminds us that, although homomorphisms are standardly described as ‘preserving’ group structure, this has to be understood with a large pinch of salt. Homomorphisms can in fact suppress many aspects of structure (indeed nearly all aspects of structure) simply by mapping distinct objects to one and same value. Perhaps the weaker ‘respecting structure’ would be better.

- (2) Suppose ‘ Z ’ denotes the integers, while ‘ $[n]$ ’ denotes the first n natural numbers (starting from 0), and ‘ $+_n$ ’ denotes addition modulo n . Then there is a homomorphism $g: (Z, +, 0) \rightarrow ([n], +_n, 0)$. Just make g send an integer k to its remainder on division by n . The underlying function $g: Z \rightarrow [n]$ is surjective but not injective.
- (3) If ‘ Q ’ denotes the rationals, then there is a homomorphism $h: (Z, +, 0) \rightarrow (Q, +, 0)$ which sends an integer to the corresponding rational. This time, as a function from Z to Q , h is injective but not surjective.

Note that in cases (2) and (3) there are other homomorphisms between the groups (except for the trivial instance of the first case when $n = 1$).

- (b) What can be said about group homomorphisms in general, various though they also are? We have the following elementary three-part result:

Theorem 1. (1) Assuming that \mathcal{G} , i.e. $(G, *, e)$, form a group, there is an identity homomorphism $1_{\mathcal{G}}: \mathcal{G} \rightarrow \mathcal{G}$ which sends each object among G to itself.

- (2) Given two homomorphisms $f: \mathcal{G} \rightarrow \mathcal{H}$, $g: \mathcal{H} \rightarrow \mathcal{J}$, with the target of the first being the source of the second, they always compose to give a homomorphism $g \circ f: \mathcal{G} \rightarrow \mathcal{J}$.

- (3) Composition of homomorphisms is associative. In other words, if one of $j \circ (g \circ f)$ and $(j \circ g) \circ f$ is defined so is the other, and they are equal.

Proof. (i) is indeed trivial. For (ii) we, of course, simply take $g \circ f$ applied to an object x among the objects of \mathcal{G} to be $g(f(x))$ and then check that $g \circ f$ so defined satisfies the condition for being a homomorphism. For (iii), associativity of homomorphisms is inherited from the associativity of ordinary functional composition for the underlying functions. \square

2.2 Group homomorphisms and isomorphisms

(c) We have seen that the underlying function of a homomorphism may or may not be injective, and may or may not be surjective. The special case where the underlying function is both injective and surjective gets its own familiar terminology and notation:

Definition 3. A *group isomorphism* $f: \mathcal{G} \xrightarrow{\sim} \mathcal{H}$ is a homomorphism where the underlying function is a bijection between the objects of \mathcal{G} and the objects of \mathcal{H} .

A *group automorphism* is a group isomorphism whose source and target are the same.

We say that the groups $(G, *, e)$ and $(G', *, e')$ are *isomorphic* as groups iff there is a group isomorphism $f: (G, *, e) \xrightarrow{\sim} (G', *, e')$. \triangle

Again, some quick examples:

- (1) There are two automorphisms from the group $(\mathbb{Z}, +, 0)$ to itself. One is the trivial identity homomorphism; the other is the function which sends an integer j to $-j$.
- (2) There are infinitely many automorphisms from the group $(\mathbb{Q}, +, 0)$ to itself: for any non-zero rational q , the map $x \mapsto qx$ ‘stretches/compresses’ the rationals, perhaps inverting their order, while preserving additive structure.
- (3) Let \mathcal{K}_1 be the group consisting in the numbers 1, 3, 5, 7 equipped with multiplication mod 8. And let \mathcal{K}_2 be the group of symmetries of a non-equilateral rectangle whose four ‘objects’ are the operations of leaving the rectangle in place, vertical reflection, horizontal reflection and rotation through 180° , with the group operation being simply composition of geometric operations. Then \mathcal{K}_1 is isomorphic to \mathcal{K}_2 .

The easiest way to see this is by noting that both groups have the same abstract ‘multiplication table’. Thus consider the table:

$*$	1	a	b	c
1	1	a	b	c
a	a	1	c	b
b	b	c	1	a
c	c	b	a	1

Read in the obvious way, this table is correct if we take $1, a, b, c$ to be respectively the numbers 1, 3, 5, 7, and take $*$ to be multiplication mod 8. It is also correct if we take $1, a, b, c$ to be the geometric operations on a rectangle as listed and take $*$ to be composition. Matching up the two interpretations of $1, a, b, c$ gives us the desired isomorphism $f: \mathcal{K}_1 \xrightarrow{\sim} \mathcal{K}_2$.

We should next note for future reference another very easy result, which gives us an alternative characterization of isomorphisms:

Theorem 2. A *group homomorphism* $f: \mathcal{G} \rightarrow \mathcal{H}$ is an *isomorphism* iff it has a *two-sided inverse*, i.e. there is a homomorphism $f': \mathcal{H} \rightarrow \mathcal{G}$ such that $f' \circ f = 1_{\mathcal{G}}$ and $f \circ f' = 1_{\mathcal{H}}$.

One structured family of structures

Proof. If $f: (G, *, e) \rightarrow (H, \star, d)$ is a bijective homomorphism, then the underlying function $f: G \rightarrow H$ is a bijection and so has a two-sided inverse $f': H \rightarrow G$. We need to show that this gives rise to a homomorphism $f': (H, \star, d) \rightarrow (G, *, e)$. But

$$f'(x \star y) = f'(ff'x \star ff'y) = f'f(ff'x * ff'y) = f'x * f'y$$

and

$$f'd = f'fe = e$$

as required.

Conversely, if f has a two-sided inverse as a homomorphism then its underlying function must have a two-sided inverse; but it is a familiar elementary result that a function with a two-sided inverse is a bijection. \square

It is almost immediate from the characterization of isomorphisms as homomorphisms with two-sided inverses that the inverses and the compositions of isomorphisms are also isomorphisms. Hence, as we would expect,

Theorem 3. *Isomorphism between groups is an equivalence relation between groups.*

2.3 New groups from old

(a) Groups, as we have already reminded ourselves, are many! Moreover, given a group or some groups, we can construct further groups from them in various ways. We will recall just three such constructions in this section:

- (1) Suppose that \mathcal{G} , i.e. $(G, *, e)$, form a group. Suppose G' are some of the objects G , and they are closed with respect to the group operation $*$ (i.e. all $*$ -products and inverses of objects among G' are also among G' , and hence e is too). Then $(G', *, e)$ form a *subgroup* of the group \mathcal{G} .
- (2) Suppose we have two groups $(G, *, e)$ and $(G', *', e')$. Assume too that we have some way of coding any ordered pair of an object x from G and an object x' from G' (where a coding scheme comes with pairing and unpairing functions in the obvious way). Let's represent the object, whatever it is, that codes the pair x, x' as $\langle x, x' \rangle$.

Let H be those pair-coding objects (i.e. every $\langle x, x' \rangle$ for some x among G and some x' among G'). Define $d = \langle e, e' \rangle$, and put $\langle x, x' \rangle \star \langle y, y' \rangle = \langle x * y, x' *' y' \rangle$. Then (H, \star, d) is easily seen to form a group – a *product* of the groups formed by $(G, *, e)$ with $(G', *', e')$.

- (3) Suppose $(G, *, e)$ form a group, and suppose that \sim is an equivalence relation defined over the objects G . We will say that \sim respects the structure of the group if, for any objects x, y, z from G , given $x \sim y$, then $x * z \sim y * z$ and $z * x \sim z * y$ (in other words, ‘multiplying’ equivalent objects by the same object yields equivalent results).

Now suppose also that for every object x from G there is a corresponding object $[x]$ (which may or may not be among G), such that $[x] = [y]$ iff $x \sim y$; and let $[G]$ be all the objects $[x]$ for x among G . Claim: we can define a binary relation \star on the objects $[G]$ by putting $[x] \star [y] = [x * y]$.

To confirm this, we need to show that the result of \star -multiplication does not depend on how we pick out the multiplicands. Thus, for multiplication-on-the-left, we need to show that if $[x] = [x']$ then $[x] \star [y] = [x'] \star [y]$. So suppose $[x] = [x']$. Then $x \sim x'$ and hence $x * y \sim x' * y$, since by hypothesis the equivalence relation respects group structure. Whence $[x * y] = [x' * y]$, which entails $[x] \star [y] = [x'] \star [y]$, as required. Similarly of course for multiplication-on-the-right.

It is now easily checked that $([G], \star, [e])$ also form a group – a *quotient of the group $(G, *, e)$ with respect to the equivalence relation \sim* .

There are other constructions for forming new groups too; but our very restricted diet of initial examples should be enough for our current purposes.

(b) Now some comments, in particular linking our new-from-old constructions to homomorphisms.

(1') Two examples of subgroups. First, take the group \mathcal{Z} , i.e. the group $(\mathbb{Z}, +, 0)$ of integers under addition. Let f be the homomorphism $x \mapsto 2x$ from that group to itself. Then the f -image of \mathcal{Z} is the subgroup of \mathcal{Z} comprising the even integers under addition.

Second, let \mathcal{R}_n be the group of rotations of an n -sided polygon onto itself, with the obvious group operation (one rotation following another), and the do-nothing rotation as the identity. And let \mathcal{P}_n be the group of permutations of the first n numbers. If we number the vertices on an n -sided polygon in the obvious way, then a rotation of the polygon corresponds to a permutation of the first n numbers, and that induces a homomorphism $f: \mathcal{R}_n \rightarrow \mathcal{P}_n$. The f -image of \mathcal{R}_n will be the subgroup of \mathcal{P}_n comprising the cyclical permutations.

(2') These two examples illustrate a general point. A very natural way in which a subgroup of a group \mathcal{G} can arise is as the image of a group \mathcal{D} under a homomorphism $f: \mathcal{D} \rightarrow \mathcal{G}$. For suppose we have a homomorphism $f: (D, \star, d) \rightarrow (G, *, e)$. Writing ' $f[D]$ ' to denote those objects which are f -images of objects from among D , then $f[D]$ are some of the objects G , and it is then very easily checked that $(f[D], *, e)$ indeed form a group which is a subgroup of the group $(G, *, e)$.

So, every homomorphism with target \mathcal{G} corresponds to a subgroup \mathcal{H} of \mathcal{G} . Conversely, a subgroup \mathcal{H} of \mathcal{G} gives rise to a trivial homomorphism, i.e. the injection $i: \mathcal{H} \rightarrow \mathcal{G}$ which sends an object x among \mathcal{H} to the same object among \mathcal{G} . Hence we can trade in talk about subgroups of \mathcal{G} for talk about homomorphisms with target \mathcal{G} . (OK, that won't be a one-to-one

trade, as different homomorphisms can give rise to the same subgroup: but the point remains that with a bit of fiddling, the idea of a subgroup is in principle replaceable by talk about homomorphisms mapping to a group.)

- (3') Next, note that products of groups are not unique, since schemes for coding ordered pairs are not unique (a familiar fact that will exercise us later). But suppose we have two pairing schemes, respectively producing pair-codes $\langle x, x' \rangle_1$ and $\langle x, x' \rangle_2$ for given x, x' . Then the groups (H_1, \star_1, h_1) and (H_2, \star_2, h_2) – the resulting products of our original groups using the two pairing schemes – will be isomorphic. For there will be a group isomorphism $f: (H_1, \star_1, h_1) \xrightarrow{\sim} (H_2, \star_2, h_2)$ generated by the bijection $\langle x, x' \rangle_1 \mapsto \langle x, x' \rangle_2$ (we of course require different pairing schemes to behave so that there *is* such a bijection!). And we won't normally care about the difference between the two product groups. In some sense, we will treat them as being 'essentially identical' – see the next section.

For a nice simple example of a product group, start with a two-object group with objects d, e (as before, e is the identity, and $d * d = e$). Form the product of this group with itself. We've just seen that it isn't going to matter which pairing scheme we choose: so fix on one, and let the resulting product group be K_3 . This has objects $\langle e, e \rangle, \langle e, d \rangle, \langle d, e \rangle, \langle d, d \rangle$ – which we can call respectively $1, a, b, c$ for short. Then it is easily checked that the derived group 'multiplication table' for these objects is again the table as above in §2.2. In other words, K_3 is isomorphic to both K_1 and K_2 .

- (4') To repeat, what we care about in coding ordered pairs is essentially that pairs come with associated pairing and unpairing functions. A pairing function codes up two objects x, x' to give us something-or-other $\langle x, x' \rangle$; matched unpairing functions enable us to recover from $\langle x, x' \rangle$ its two 'components' x and x' . And as long as pairing and unpairing behave as we would expect, we won't really care at all about the intrinsic character of the something-or-other $\langle x, x' \rangle$ which codes for the two objects.

So, when we come to think more about product groups, we will find ourselves focusing not on the objects that form a particular product group but rather on suitable homomorphisms to and from product groups. More about this, at some length, in Chapters 7 and 8.

- (5') In constructing the product of two groups, then, we really only care about the result 'up to isomorphism'. The same goes for quotient groups.

Here's a baby example of such a group. Consider $(\mathbb{Z}, +, 0)$, the group of integers under addition. Congruence mod 8 is an equivalence relation \equiv_8 on the integers which respects additive structure. That is to say, if $x \equiv_8 y$ then $x + z \equiv_8 y + z$ and $z + x \equiv_8 z + y$.

We can now easily find objects $[x]$ such that $[x] = [y] \leftrightarrow x \equiv_8 y$. For example, one choice is simply to take $[x]$ to be the remainder on division of x by 8. With this choice, the quotient of the group $(\mathbb{Z}, +, 0)$ by \sim is just the group of integers from 0 to 7, equipped with addition modulo 8, and

with 0 as the identity.

However, we could equally well have taken other objects to act as the objects $[x], [y]$ whose identity goes with the equivalence $x \sim y$ – for another obvious example we could have taken equivalence classes built in your favourite applicable set theory with urllements. So, again, the quotient of a group by a suitable equivalence relation is not unique. Just as we can implement products in various ways, we can implement representatives of equivalent objects in various ways. As with products, different quotients of a group built by different implementations will be isomorphic. And also as with products, we won't care about, so to speak, the inner details of the implementation so long as it 'works'. Further, as we will also see, 'working' can be thought of as a matter of the quotient groups behaving in the right way as sources and targets of some homomorphisms. More about this in due course.

Here then we get our first glimpses of what will be a key categorial theme: *very* roughly, we can trade in claims about what is going on inside structures for claims about the homomorphisms or other maps between them.

2.4 'Identity up to isomorphism'

(a) We have met the groups $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ which are isomorphic to each other. They are also isomorphic to any other group whose four objects can be labelled $1, a, b, c$ in such a way that the same 'multiplication table' applies again. Call such groups *Klein four-groups*.

Klein four-groups are many. Trivially, \mathcal{K}_1 , the group consisting in the natural numbers 1, 3, 5, 7 equipped with multiplication mod 8, is distinct from \mathcal{K}_2 , the group consisting in the symmetries of some non-equilateral rectangle, for the boring reason that natural numbers just aren't symmetries of a rectangle. And neither \mathcal{K}_1 nor \mathcal{K}_2 is the same as \mathcal{K}_3 (at least on most choices of pairing schemes for constructing \mathcal{K}_3).

Still, the way in which $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ and the other Klein four-groups differ from each other, namely in the constitution of their various *objects*, is not at all relevant to their behaviour as groups, for that depends just on the *relations between the objects*. In other words, despite the differences between their objects, the groups are the same at least as far as their group-theoretic properties – the properties as determined by their shared 'multiplication table' – are concerned. For that reason, although our sample Klein groups are not strictly identical, a group theorist will say that they are 'essentially identical' or, perhaps rather better, are *identical up to isomorphism*.

(b) Apparently going further, however, it is common to talk of *the* Klein four-group – and to similarly talk about *the* one-object group, *the* product of two

One structured family of structures

given groups, *the* quotient of a group by an equivalence relation, etc., etc. What does this mean? There are three ways of parsing such talk, two simple and natural, one which initially looks more problematic. Sticking to the Klein example,

- (1) Perhaps some one Klein group has in fact been introduced as a paradigm case, a canonical exemplar, and discussion of ‘the’ Klein group refers to *that*. The generalizability of group-theoretic claims about ‘the’ Klein group to cover isomorphic groups – the rest of the Klein groups – is then taken for granted.

(You might have met the notion of a free group. Standardly, a particular concrete implementation using lists of A -elements (and potential inverses for A -elements) is specified as constituting *the* free group with generators A . Then this is taken as a canonical exemplar.)

- (2) Alternatively, no one Klein group is especially picked out for playing a privileged role, but rather claims about ‘the’ Klein group are construed from the outset as general claims about all Klein groups. So ‘The Klein group is abelian’ is to be understood as simply saying that any Klein group is abelian; similarly ‘There is a unique homomorphism from the Klein group to the one object group’ says that for any Klein group and any one object group, there is a unique homomorphism from the first to the second. And so on.
- (3) A third line supposes that – as well as all the ‘concrete’ Klein four groups built from numbers, or from symmetries of a rectangle, or from ordered pairs, etc. – there is an ‘abstract’ Klein four group. The objects of *this* group are then supposed to have no intrinsic nature – they aren’t numbers or symmetries or whatever, but are just (as it were) bare positions in the structure, with no basic features other than being related to each other as displayed in the group’s multiplication table.

The last view may for a moment look rather mysterious. But compare: it is a familiar thought, which goes back to Dedekind, that e.g. the natural numbers likewise aren’t this or that concrete ω -sequence but are, more abstractly, bare positions in an ω -sequence which have no basic features other than being related to each other to form such a sequence. And if the natural numbers can be thought of abstractly like that, why not the objects comprising an abstract Klein group?

But obviously, we don’t want to get bogged down pursuing such philosophical questions right now – we’ve got some category theory to do! For the moment, then, we won’t pause to fuss about this sort of thing: we’ll just let talk of ‘*the* Klein group’ be interpreted however you prefer (the exemplar reading, the generalizing reading, the Dedekind abstraction reading). And similarly, of course, for e.g. ‘*the* one-object group’, and so on. And later, similarly for the likes of ‘the one-object category’. In due course, however, we may well want to return to this theme.

2.5 Groups and sets

(a) The basic ideas about groups and group homomorphisms in the previous sections have no doubt been quite familiar. But now let's highlight a deviant feature of our presentation so far.

What is a group? We said: some objects G (one or more) equipped with an associative two-place operation defined over them, satisfying some familiar conditions. Standard textbook definitions, however, announce that a group is a *set* of objects Γ endowed with a binary operation etc.

What is a homomorphism from one group (say, some objects G suitably equipped) to another group (some objects G' suitably equipped)? We said it is a function $f: G \rightarrow G'$, sending an object among G to an object among G' , and satisfying some other familiar conditions. Standard textbook definitions, however, announce that a homomorphism is a function from a *set* of objects Γ into another set of objects Γ' .

So why did we initially use plural talk rather than use set talk? What additional commitments do we take on by moving to the more conventional way of introducing groups? Do we *need* to take on these additional commitments?

(b) Let's pause to distinguish three grades of commitment to sets (maybe there are finer-grained distinctions to be made, but these coarse distinctions ought to be enough for now).

- (1) Much informal set talk is simply an idiom for talking about many things at once. Instead of referring, plurally, to the X s, we refer singly to the set of X s.

And indeed, this is often a *useful* idiom. By a historical quirk, received logical symbolism likes to keep reference singular. So when it comes to shoehorning mathematical claims into Loglish (that familiar mash-up of logical notation and ordinary English that mathematicians often use for clarity), it is handy to be able to use singular talk of the set of X s, rather than try to refer plurally to the X s. But in many cases, this is indeed no more than a *façon de parler*.

At this level, to say p is a member of the set of prime numbers greater than two is just to say that p is one of the primes great that two. To say that the set of prime numbers greater than two is a subset of the set of odd numbers is just to say that the prime numbers greater than two are among the odd numbers. Similarly, talk about e.g. the intersection of the set of primes and the set of even numbers can be paraphrased away into talk about the numbers which are both prime and even. And so it goes.

Such paraphrasable set-talk is often said to be talk of *virtual* classes. It is useful but non-committal, and is straightforwardly eliminable.

- (2) Merely virtual classes are not objects in their own right which can, in particular, themselves be members of sets. So we take on a different grade of

One structured family of structures

commitment to sets when we start allowing them to be members of other sets. Suppose we now do this, in the sort of ways explained in those introductory chapters on logic and sets in many a non-set-theory mathematics text: take e.g. Munkres's standard textbook *Topology* (2000) as a typical example.

So now, if we start with e.g. the natural numbers, we can construct not just sets of numbers, but sets of sets of numbers, sets of sets of sets of numbers, etc.

Similarly, we can start with other elements at the base level, e.g. points of some kind, and build a space as a set of points, and form a set of subsets of the set of points to get a topology, etc.

Two points about these sets-for-applied-use:

- (i) In a particular application, there is a base level of non-sets – natural numbers, real numbers, points, functions, whatever.

The empty set is, of course, normally allowed. However – quite explicitly e.g. for Munkres – it can be treated as a fiction. It's there as a convenience, so we don't have to qualify e.g. the construction of the intersection of two sets A and B by the clause 'if A and B have an element in common, ...', and so on.

- (ii) The 'set-of' operation in practice only gets iterated a finite number – indeed, a very *small* finite number – of times.

Even, say, the familiar arithmetization of analysis – constructing integers from pairs of naturals, rationals from pairs of integers, reals from sets of rationals, and n -place functions as $n + 1$ -tuples of reals requires only a dozen or so levels of sets-of-sets-of-...-sets-of-naturals. (If we want to use ordinal numbers to index repeated operations in 'ordinary' mathematics, just treat the ordinals we need as we do the naturals – as axiomatically governed primitives.)

- (3) Contrast this with the grade of commitment to sets which we adopt when we buy the set-theorist's universe of sets as described in the canonical theory ZFC. Here, contra (i) above, the sets are 'pure', with the empty set – rather than being a convenient fiction – taken as the sole base-level object. Then, to recover even an ω -sequence of objects isomorphic to the natural numbers, we need, contra (ii) above, to iterate the set-building operation into transfinite. (Fraenkel's Axiom of Replacement then allows us to keep going, pulling ourselves up by our bootstraps, with an iteration step for every ordinal we can construct within the theory.)

So, to repeat our question: which grade of commitment do we need to take on while discussing families of groups as structures-of-structures?

- (c) As we've seen, we can at least make a start by just talking plurally of a group as some objects together with suitable functions; and then a structured family of

groups will be lots of these things together with homomorphisms between them. True, it soon gets very convenient to use some equivalent virtual set talk: but at the outset, this is eliminable.

However, here's a reason for ascending to the second grade of commitment to sets. Suppose we want to be able to assume that, given two groups (at least, two groups of the same kinds of objects), there is always a product of those groups. Then given an object from each of the groups, we need there to be something that will 'work' as an ordered pair of them. Given sets-for-applied-use, we can use the familiar Kuratowski definition for pair-objects; then we get what we need in a uniform way. Similarly, suppose we want to be able to assume that, for any group with a suitable equivalence relation defined over its objects, there is always a quotient group. Then we need to be able to go from the original group objects to something that 'works' as a plurality of equivalence classes of them. Obviously, sets-for-applied-use give us a uniform way of doing this. Similarly too for various other standard constructions of ordinary mathematics: sets-for-applied-use provide a setting in which the constructions can be uniformly carried out.

(d) What more, then, would we gain by going up another level of commitment to sets, and working now within pure set theory?

Take the thought that, insofar as we are interested in the structural, group-theoretic, properties of a group, we don't care about the difference between isomorphic copies of the group: any Klein group, for example, will do. Combine that with the thought that the world of pure sets is rich enough to have an isomorphic copy of any given group. Then we might as well just consider groups living in the world of sets. Up to isomorphism, that will give us all the groups.

The world of pure sets is also rich enough to be able to model any homomorphisms between groups by functions-as-sets between isomorphic copies of those groups, with everything in our model now living in the world of pure sets.

And what goes for groups goes for Boolean algebras, or topological spaces, or metric spaces, and the maps between *them*. We can again find isomorphic copies of these structures in the universe of pure sets. Since we don't really care about the difference between the copies, *we might as well henceforth concentrate just on the examples living in the world of sets.*

To repeat, this is not to *identify* groups (for example) with sets across the board – or at least, it only does so 'up to isomorphism'. The claim is that, insofar as we care about the structural properties of groups (for example), we can without loss of generality look just at the ones that *are* sets. And likewise for other mathematical structures.

Hence, by thinking just about pure sets, we get a single stage on which a motley cast of mathematical structures can "have their exits and entrances" and play their many parts. That's a nice economy, and it gives us one familiar stage setting in which to devise our drama.

2.6 An unresolved tension

We have now seen why – although we might start off thinking of a particular mathematical structure as just some objects (plural) equipped with various gadgetry – there is pressure to get entangled with the set-theoretic ideas. First, by adopting the apparatus of sets-for-applied-use, we get what it takes to construct pairs, quotients, etc. in a systematic way that can be applied across the board. Then second, by going the step further into pure set theory, we get a single unifying setting for our investigations.

Hence we might well suppose that category theory – which, as we announced, is going to be our framework for talking about structures of structures – can, essentially, be thought of as a way of talking about set-theoretic constructions, all living in the world of sets. That’s why Saunders Mac Lane in his canonical *Categories for the Working Mathematician* can say, simply, a category will be ‘any interpretation of the category axioms within set theory (1997, p. 10).

However, there *is* an alternative line of thought about category theory which apparently goes in quite the opposite direction. Mathematicians are right, the rival story goes, in their ordinary supposition that there are fundamentally different kinds of mathematical structure built up of different kinds of objects and maps between them. Moreover these different kinds of structure stand on their own feet, so to speak, *without* needing reduction to sets. Indeed, the world of pure sets is then just one big structure living in a wider democratic universe of structures. And category theory allows us to talk about the interrelations of these structures (and the place of the world of sets in the wider universe), while breaking free from set-theoretic imperialism.

There seems to be an unresolved tension here which we certainly aren’t in a position to comment on at this point. And this tension is going to show through at various points in what follows – indeed, in a later version of these notes I’m going to have to take a definite line, and tidy up some moments of arm-waving indecision. I’m afraid that, on this score, at various points we currently just muddle through ...

3 Categories defined

Let's dive straight in, and immediately give a standard definition of categories, followed by a range of initial examples. We proceed informally to begin with. The logically pernickety may immediately spot some potentially worrying issues. We soon comment on those, however.

3.1 The very idea of a category

We said that many paradigm cases of categories are families of structures with structure-preserving maps between them. But what can we say about such families at an abstract level?

One sufficiently general thought is this: if, within a family of structures including A , B , and C we have a structure-preserving map f from A to B , and another structure-preserving map g from B to C , then we should be able to *compose* these maps. That is to say, the first map f followed by the second g should also count as a structure-preserving map $g \circ f$ from A to C .

What principles will govern such composition of maps? Associativity, surely. Using a natural diagrammatic notation, if we are given maps

$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$$

it really ought not matter how we carve up the journey from A to D . It ought not matter whether we apply the map f followed by the composite g -followed-by- h , or alternatively apply the composite map f -followed-by- g and then afterwards apply h .

What else can we say at the same level of stratospheric generality about families of structures and structure-preserving maps? Very little indeed. Except that there presumably will always be the limiting case of a 'do nothing' identity map, which applied to any structure A leaves it untouched.

That apparently doesn't give us a great deal to work with. But in fact it is already enough to shape our definition of categories. We abstract from the idea of families of structures with structure-preserving maps between them, and – using more neutral terminology – we'll speak more generally of *objects* and of *arrows* between them. Then we say:

Categories defined

Definition 4. A category \mathcal{C} comprises two kinds of things:

- (1) \mathcal{C} -objects (which we will typically notate by ‘ A ’, ‘ B ’, ‘ C ’, ...).
- (2) \mathcal{C} -arrows (which we typically notate by ‘ f ’, ‘ g ’, ‘ h ’, ...).

These objects and arrows are governed by the following axioms:

Sources and targets For each arrow f , there are unique associated objects $\text{src}(f)$ and $\text{tar}(f)$, respectively the *source* and *target* of f , not necessarily distinct.

We write ‘ $f: A \rightarrow B$ ’ or ‘ $A \xrightarrow{f} B$ ’ to notate that f is an arrow with $\text{src}(f) = A$ and $\text{tar}(f) = B$.

Composition For any two arrows $f: A \rightarrow B$, $g: B \rightarrow C$, where $\text{src}(g) = \text{tar}(f)$, there exists an arrow $g \circ f: A \rightarrow C$, ‘ g following f ’, which we call the *composite* of f with g .

Identity arrows Given any object A , there is an arrow $1_A: A \rightarrow A$ called the *identity arrow* on A .

We also require the arrows to satisfy the following further axioms:

Associativity of composition. For any $f: A \rightarrow B$, $g: B \rightarrow C$, $h: C \rightarrow D$, we have $h \circ (g \circ f) = (h \circ g) \circ f$.

Identity arrows behave as identities. For any $f: A \rightarrow B$ we have $f \circ 1_A = f = 1_B \circ f$. △

Evidently, given what we have already said, the objects which are mathematical structures of a particular kind taken together with the arrows which are structure-preserving maps between them should satisfy those axioms, and hence should indeed count as forming a category.

Here are six more quick remarks on terminology and notation:

- (a) The objects and arrows of a category are very often called the category’s *data*. That’s a helpfully neutral term if you don’t read too much into it, and we will occasionally adopt this common way of speaking.
- (b) The labels ‘objects’ and ‘arrows’ for the two kinds of data are quite standard. But note that the ‘objects’ in categories needn’t be objects at all in the logician’s familiar strict sense, i.e. in the sense which contrasts objects with entities like relations or functions. There are perfectly good categories whose ‘objects’ – in the sense of their first type of data – are actually relations, and other categories where they are functions.
- (c) Borrowing familiar functional notation ‘ $f: A \rightarrow B$ ’ for arrows in categories is entirely natural given that arrows in many categories *are* (structure-preserving) functions: indeed, that is the motivating case. But again, as we’ll soon see, not all arrows in categories are functions. Which means that not all arrows are morphisms either, in the usual sense of that term. Which is why I rather prefer the colourless ‘arrow’ to the equally common

term ‘morphism’ for the second sort of data in a category. (Not that that will stop me often talking of morphisms or maps when context makes that natural!)

- (d) In keeping with the functional paradigm, the source and target of an arrow are frequently called, respectively, the ‘domain’ and ‘codomain’ of the arrow (for usually, when arrows are functions, that’s what the source and target are). But that usage has the potential to mislead when arrows aren’t functions (or aren’t functions ‘in the right direction’), which is again why I prefer our common alternative terminology.
- (e) Note the order in which we write the components of a composite arrow (some from computer science do things the other way about). Our standard notational convention is again suggested by the functional paradigm. In a category where $f: A \rightarrow B$, $g: B \rightarrow C$ are both functions, then $(g \circ f)(x) = g(f(x))$. Occasionally, to reduce clutter, we may later allow ourselves to write simply ‘ gf ’ rather than ‘ $g \circ f$ ’.
- (f) In early chapters we will explicitly indicate which object an identity arrow has as both source and target, as in ‘ 1_A ’. Eventually, again to reduce clutter, we will often allow ourselves simply write ‘ 1 ’ when context makes it clear which identity arrow is in question.

Our axioms now imply our first mini-result:

Theorem 4. *Identity arrows on a given object are unique; and the identity arrows on distinct objects are distinct.*

Proof. For the first part, suppose A has identity arrows 1_A and $1'_A$. Then applying the identity axioms for each, we immediately have $1_A = 1_A \circ 1'_A = 1'_A$.

For the second part, we simply note that $A \neq B$ entails $\text{src}(1_A) \neq \text{src}(1_B)$ which entails $1_A \neq 1_B$. \square

(As this illustrates, the most trivial of lemmas, as well as run-of-the-mill propositions, interesting corollaries, and the weightiest results, will all be labelled ‘theorems’ without distinction.)

A remark. We’ve just seen that every object in a category is associated with one and only one identity arrow; and we can in fact pick out such identity arrows by the special way they interact with all the other arrows. Hence we could in principle give a variant definition of categories which initially deals just in terms of arrows. For an account of how to do this, see Adámek et al. (2009, pp. 41–43). But I find this bit of trickery rather unhelpful. As we will see, a central theme of category theory is indeed the idea that we should probe the objects in a category by considering the arrows between them; but that’s no reason to write the objects out of the story altogether.

3.2 Monoids and pre-ordered collections

(a) We start by looking at two simple but instructive examples of categories. First an algebraic example; and we'll begin not with families of groups but – to cut the algebraic structure to the bone – families of *monoids*. Recall: a monoid is, so to speak, a group except for the requirement for inverses. So,

Definition 5. A monoid (M, \cdot, e_M) comprises some objects M , equipped with a two-place ‘multiplication’ function (defined over M , with values among M) and with a distinguished object e_M . It is required only that

- (i) ‘multiplication’ is associative: for all elements a, b, c among M , $(a \cdot b) \cdot c = a \cdot (b \cdot c)$,
- (ii) the distinguished object e_M acts as a unit, i.e. is such that for any a among M , $e_M \cdot a = a = a \cdot e_M$.

A monoid homomorphism $f: (M, \cdot, e_M) \rightarrow (N, \times, e_N)$ is then defined to be a function $f: M \rightarrow N$ between the objects of the monoids which preserves ‘products’ and units. In other words, for any a, b among M , $f(a \cdot b) = fa \times fb$, and also $fe_M = e_N$. \triangle

The function f between the underlying objects of the monoids is said to be the underlying function of the homomorphism f between monoids. (We could adopt distinguishing notation, and use e.g. ‘ \underline{f} ’ for the underlying function. But since context will usually make it clear which we are talking about, we’ll be notationally more relaxed.)

It is evident that monoid homomorphisms $f: (M, \cdot, e_M) \rightarrow (N, \times, e_N)$ and $g: (N, \times, e_N) \rightarrow (O, *, e_O)$ compose to give a homomorphism $g \circ f: (M, \cdot, e_M) \rightarrow (O, *, e_O)$. Composition of homomorphisms is associative (because composition of the underlying functions is). And the identity function on objects M is a homomorphism $f: (M, \cdot, e_M) \rightarrow (M, \cdot, e_M)$ which acts as an identity with respect to composition.

Which all adds up to give us our first official example of a category:

- (C1) **Mon** is the category whose objects are all monoids and whose arrows are the monoid homomorphisms.

(b) Next, an example involving ordered objects; and again we’ll cut structure to the bone by considering the simplest case, pre-orderings.

Definition 6. The pre-ordered collection (M, \preceq) comprise some objects M equipped with a pre-ordering \preceq , i.e. a relation such that for all a, b, c among M ,

- (i) if $a \preceq b$ and $b \preceq c$, then $a \preceq c$,
- (ii) $a \preceq a$.

3.3 Some rather sparse categories

A monotone map $f: (M, \leq) \rightarrow (N, \sqsubseteq)$ between such pre-orderings is then defined to be a function $f: M \rightarrow N$ between the underlying objects which respects order, i.e. such that for any a, b among M , if $a \leq b$, then $fa \sqsubseteq fb$. \triangle

It is again immediate that monotone maps between pre-ordered collections compose to give monotone maps, and the identity map on some objects gives rise to an identity monotone map on those-objects-equipped-with-a-pre-order. So we have our second example of a category:

- (C2) **Ord** is the category whose objects are all pre-ordered collections and whose arrows are the monotone maps between such collections.

3.3 Some rather sparse categories

(a) So far, so very unsurprising! But now note that monoids can get into the story in a second way. As we've seen, monoids as objects taken together with all the monoid homomorphisms as arrows form a (*very* large!) category **Mon**. However, any single monoid taken just by itself can also be thought of as corresponding to a category (perhaps *very* small category). Here's how:

- (C3) Take any monoid (M, \cdot, e_M) . Then define a corresponding category \mathcal{M} whose data is as follows:
- (1) \mathcal{M} 's sole object is some arbitrary entity – choose whatever you like, it *doesn't* have to be one of the objects M , and dub it ' \star ';
 - (2) An \mathcal{M} -arrow $a: \star \rightarrow \star$ is just one of the monoid's objects a , with composition of arrows $a \circ b$ defined to be the monoid product $a \cdot b$, and with the identity arrow 1_\star defined to be the monoid identity e_M .

It is trivial that the category axioms are satisfied. So we can think of any monoid as in effect being a one-object category. (Conversely, a one-object category gives rise to an associated monoid built from its arrows, and we can think of categories as, in a sense, generalized monoids.)

Note in this case, since the 'object' in the category can be anything you like, it needn't be an object in any ordinary sense (let alone be a structure). And unless the objects of the original monoid M happen to be functions, the arrows of the associated category \mathcal{M} will also not be functions or morphisms or maps in any ordinary sense. So this sort of single-monoid-as-a-category won't usually be a 'structure of structures'!

(b) Similarly, we can think of any single collection of pre-ordered objects just by itself as forming a category. Here's how.

- (C4) Take any pre-ordered objects (N, \leq) . Then define a corresponding category \mathcal{N} whose data is as follows:
- (1) \mathcal{N} 's objects are the objects N again;

Categories defined

(2) there is a (single) \mathcal{N} -arrow from A to B just in case $A \leq B$ – this arrow might as well be identified as the ordered pair (A, B) , and then we can define composition by putting $(B, C) \circ (A, B) = (A, C)$. Take the identity arrow 1_A to be (A, A) .

It is trivial that, so defined, the arrows for \mathcal{N} satisfy the identity and associativity axioms, so we do indeed have another category here – and again, not one comprising structures and structure-preserving maps. (Conversely, any category with objects O and where there is at most one arrow between objects can be regarded as a pre-ordered set (O, \leq) , where for A, B among O , $A \leq B$ just in case there is an arrow from A to B in the category).

It is therefore natural to call a category with at most one arrow between objects a *pre-order category*. And we can think of the unrestricted notion of a category as a generalization of the case of preordered collections.

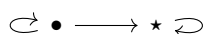
(c) Monoids-as-categories and pre-ordered-objects-as-categories can give us very small categories with few objects and/or arrows. And here are some more sparse categories.

(C5) For any collection of objects M , there is a *discrete category* on those objects. This is the category whose objects are just the members of M , and which has as few arrows as possible, i.e. just the identity arrow for each object in M .

(C6) For convenience, we can allow the empty category, with zero objects and zero arrows. Otherwise, the smallest discrete category is 1 which has exactly one object and one arrow (the identity arrow on that object). Let's picture it in all its glory!



(C7) And having mentioned the one-object category 1 , here's another very small category, this time with two objects, the necessary identity arrows, and one further arrow between them. We can picture it like this:



Call this category 2 . We can think of as arising from the von Neumann ordinal 2 , i.e. the set $\{\emptyset, \{\emptyset\}\}$; take the ordinal's members as objects of the category, and let there be an arrow between objects when the source is a subset of the target. Other von Neumann ordinals, finite and infinite, similarly give rise to other categories.

But hold on! Should we in fact talk about *the* category 1 (or the category 2 , etc.)? Won't different choices of object make for different one-object categories?

Well, yes and no! We can have, in our mathematical universe, different cases of single objects equipped with an identity arrow – *but they will be indiscernible from within category theory*. So as far as category theory is concerned, they are all ‘essentially the same’ (in the same spirit as e.g. different Klein four-group are ‘essentially the same’).

3.4 More categories

Let’s continue our list of examples, first generalizing from the cases of **Mon** and **Ord**, and then adding some geometric and other categories.

The category of monoids is just the first of a family of similar cases, where the objects are algebraic structures – comprising objects equipped with some functions (including zero-place functions picking out certain distinguished objects) – and the arrows are the homomorphisms preserving the relevant amount of structure. Thus, we also have:

- (C8) **Grp**, the category of groups. The objects are groups – i.e. monoids where every object has an inverse. The arrows are group homomorphisms.
- (C9) **Ab** is the category whose objects are abelian groups, and whose objects are group homomorphisms again.
- (C10) **Rng** is, the category of rings, whose objects are predictably enough all rings and whose objects are ring homomorphisms.
- (C11) And **Bool** is the category of Boolean algebras and structure-preserving maps between them.

And so it goes!

We similarly have further categories of ordered objects. Enrich the notion of a pre-order, take as structures objects-equipped-with-the-richer-order, take as arrows order-preserving functions, and we get another category. For example

- (C12) **Pos** is the category of partially-ordered collections (where a partial order is a pre-order which is anti-symmetric), and the arrows are order-preserving maps again.
- (C13) **Tot** is the category of totally-ordered collections (where a total order is partial order where any two objects stands in the order relation, one way round or the other). The arrows are as you would now expect!

Now for another paradigm type of category, namely geometric categories (as central to the development of the theory as the cases of algebraic categories like **Mon** and **Grp** or order categories).

- (C14) **Top** is the category with
 - objects: all the topological spaces,
 - arrows: the continuous maps between spaces.

Categories defined

(C15) **Met** is also a category: this has

objects: metric spaces, which we can take to be a set of points S equipped with a real metric d ,

arrows: the non-expansive maps, where – in an obvious notation – $f: (S, d) \rightarrow (T, e)$ is non-expansive iff $d(x, y) \geq e(f(x), f(y))$.

(C16) **Vect_k** is a category with

objects: vector spaces over the field k (each such space is a set of vectors, equipped with vector addition and multiplication by scalars in the field k),

arrows: linear maps between the spaces.

Finally in this section, let's have a logical example.

(C17) Suppose T is a formal theory (the details don't matter for our example, so long as you can chain proofs in a standard sort of way). Then there is a category **Proof_T** with

objects: sentences φ, ψ, \dots of the formal language of T ,

arrows: there is an arrow $d: \varphi \rightarrow \psi$ iff $T, \varphi \vdash \psi$, i.e. there is a formal proof in the formal theory T of the conclusion ψ given φ as premiss.

3.5 The category of sets

(a) As in the previous chapter, we have again been using plural talk – for example, we took a monoid to be some objects (plural) equipped with a binary relation. But of course as we said before, it is standard to think of a structure like a monoid as being, officially, a *set* of objects equipped with that relation. If we are only interested in distinguishing monoids, say, up to isomorphism, then these set structures will give us enough to be getting on with!

So, this means that for many examples of categories for which we can start thinking of their objects as paradigmatically being *sets*-equipped-with-widgets, and the arrows between such objects will then be suitable *set-functions between these carrier sets*. In the extremal case, the sets will come equipped with *no* additional structure. And then we get the following category! –

(C18) **Set** is the category with

objects: all sets.

arrows: given sets X, Y , every (total) set-function $f: X \rightarrow Y$ is an arrow.

There's an identity function on any set. Set-functions $f: A \rightarrow B, g: B \rightarrow C$ (where the source of g is the target of f) always compose. And so the axioms for being a category are evidently satisfied.

Three initial remarks:

- (i) Note that the arrows in **Set**, like any arrows, must come with determinate targets/codomains. But the standard way of treating functions set-theoretically is simply to identify a function f with its *graph* \hat{f} , i.e. with the set of pairs (x, y) such that $f(x) = y$. This definition is lop-sided in that it fixes the function's source/domain, the set of first elements in the pairs, but it doesn't determine the function's target. (For a quite trivial example, consider the **Set**-arrows $z: \mathbb{N} \rightarrow \mathbb{N}$ and $z': \mathbb{N} \rightarrow \{0\}$ where both functions send every number to zero. Same graphs, but functions with different targets and correspondingly different properties – the second is surjective, the first isn't.)

Perhaps set theorists themselves ought really to identify a set-function $f: A \rightarrow B$ with a triple (A, \hat{f}, B) . But be that as it may, that's how category theorists can officially regard arrows $f: A \rightarrow B$ in **Set**.

- (ii) We should perhaps remind ourselves why there *is* an identity arrow for \emptyset in **Set**. Vacuously, for *any* target set Y , there is exactly one set-function $f: \emptyset \rightarrow Y$, i.e. the one whose graph is the empty set. Hence in particular there is a function $1_{\emptyset}: \emptyset \rightarrow \emptyset$.

Note that in **Set**, the empty set is in fact the *only* set such that there is exactly one arrow from it to any other set. This gives us a nice first example of how we can characterize a significant object in a category not by its internal constitution, so to speak, but by what arrows it has to and from other objects. Much more on this sort of point later.

- (iii) The function $id_A: A \rightarrow A$ defined by $id_A(x) = x$ evidently serves in the category **Set** as the (unique) identity arrow 1_A .

We can't say that, however, in pure category-speak. Still, we can do something that comes to the same. Looking ahead, note first that we can define singletons in **Set** by relying on the observation that there is exactly one arrow from any object *to* a singleton. So now choose a singleton, it won't matter which one. Call your chosen singleton '1'. And consider the possible arrows (i.e. set-functions) from 1 to A .

We can represent the arrow from 1 to A which sends the element of the singleton 1 to $x \in A$ as $\vec{x}: 1 \rightarrow A$ (the over-arrow here is simply a helpful reminder that we are indeed notating an arrow). Then there is evidently a one-one correspondence between these arrows \vec{x} and the elements $x \in A$. So talk of such arrows \vec{x} is available as a category-speak surrogate for talking about elements x of A . Hence now, instead of saying $id_A(x) = x$ for all elements x of A , we can say that for any arrow $\vec{x}: 1 \rightarrow A$ we have $1_A \circ \vec{x} = \vec{x}$.

Again, more on this sort of thing in due course: but it gives us another glimpse ahead of how we might trade in talk of sets-and-their-elements for categorial talk of sets-and-arrows-between-them.

(b) So far, so straightforward. But there is a more substantive issue about this standard example of a category that we can't just pass by in silence.

For we can ask: exactly what category do we have in mind here when talking about **Set**? – we haven't been explicit. For a start, what *kind* of sets are its objects? Are these sets-for-applied-use (built up from urelements), in the sense of §2.5. Or are these the pure sets as governed by the axioms of ZFC? Or indeed do we accept stronger set-existence axioms (large cardinal axioms, for example). Or what about taking a differently structured universe of sets better described by a rival set theory like Quine's NF (or NFU, the version with urelements again)?

The answers could matter later for various purposes. But we cannot pause over them now or we'll never get started! So the conventional dodge is just to take your favoured conception of the universe of sets and work with that (or perhaps, if you think that the universe is indeterminately open-ended, consider levels of the set universe up to some suitable 'inaccessible' rank to get enough sets for all the ordinary maths you want to do): its objects and functions should assuredly satisfy at least the modest requirements for constituting a category. Therefore, for the moment, you can just interpret our talk of sets and the category **Set** in your preferred way assuming that isn't too wildly idiosyncratic!

(c) Note that familiar size considerations now kick in. The category of sets has all sets as its objects. There is no set of all sets however – such a collection is, in a familiar way, 'too big' to be a set. So the category of sets is itself too big to be a set or to be modelled as a set. *That's why our initial definition of a category did not say e.g. that a category always comprises a set of objects.*

Similarly for e.g. the category of monoids. Even throwing away the isomorphic copies outside the world of sets, there are too many monoids-as-sets left for there to be a set of them. In other words, the category of monoids too will have more than a set's worth of objects. Similarly again for many other categories. We will be returning though to issues of size.

3.6 Yet more examples

Let's finish our initial list of examples of categories. And now we can go more briskly:

(C19) There is a category **FinSet** whose objects are the hereditarily finite sets (i.e. sets with at most finitely many members, these members in turn having at most finitely many members, which in turn ...), and whose arrows are the set-functions between such objects.

(C20) **Pfn** is the category of sets and *partial* functions. Here, the objects are all the sets again, but an arrow $f: A \rightarrow B$ is a function not necessarily everywhere defined on A (one way to think of such an arrow is as a total function $f: A' \rightarrow B$ where $A' \subseteq A$). Given arrows-qua-partial-

functions $f: A \rightarrow B$, $g: B \rightarrow C$, their composition $g \circ f: A \rightarrow C$ is defined in the obvious way, though you need to check that this indeed makes composition associative.

(C21) Set_\star is the category (of ‘pointed sets’) with

objects: all the non-empty sets, with each set A having a distinguished member \star_A (or equivalently, think of each A as equipped with a zero-place function picking out some $\star_A \in A$).

arrows: all the total functions $f: A \rightarrow B$ which map the distinguished member \star_A to the distinguished member \star_B , for any objects A, B .

As we’ll show later, Pfn and Set_\star are in a good sense equivalent categories (it is worth pausing to think why we should expect that).

(C22) The category Rel again has naked sets as objects, but this time an arrow $A \rightarrow B$ in Rel is (not a function but) any relation R between A and B . We can take this officially to be a triple (A, \hat{R}, B) , where the graph $\hat{R} \subset A \times B$ is the set of pairs (a, b) such that aRb .

The identity arrow on A is then the diagonal relation with the graph $\{(a, a) \mid a \in A\}$. And $S \circ R: A \rightarrow C$, the composition of arrows $R: A \rightarrow B$ and $S: B \rightarrow C$, is defined by requiring $a S \circ R c$ if and only if $\exists b(aRb \wedge bSc)$. It is easily checked that composition is associative.

So here we have another example where the arrows in a category are *not* functions.

And that will do for the moment as an introductory list. There is no shortage of categories, then!

Indeed we might well begin to wonder whether it is just *too* easy to be a category. If such very different sorts of structures as e.g. a particular small monoid on the one hand and the whole universe of topological spaces on the other hand equally count as categories, how much mileage can there be general theorizing about categories and their interrelations?

Well, that’s exactly what we hope to find out over the coming chapters.

3.7 Diagrams

We can graphically represent categories (objects related by arrows) in a very natural way – we’ve already seen a couple of trivial mini-examples. And in particular, we can represent facts about the equality of arrows using so-called commutative diagrams. We’ll soon be using diagrams a great deal: so we’d better say something about them straight away.

Talk of diagrams is in fact used by category theorists in three related ways. In §17.1 we will give a sharp formal characterization of one notion of diagram.

Categories defined

For the moment, we can be more informal and work with two looser but more immediately intuitive notions:

Definition 7. A *representational diagram* is a ‘graph’ with nodes representing objects from a given category \mathcal{C} , and drawn arrows between nodes representing arrows of \mathcal{C} . Nodes and drawn arrows are usually labelled.

Two nodes in a diagram can be joined by zero, one or more drawn arrows. A drawn arrow labelled ‘ f ’ from the node labeled ‘ A ’ to the node labeled ‘ B ’ of course represents the arrow $f: A \rightarrow B$ of \mathcal{C} . There can also be arrows looping from a node to itself, representing the identity arrow on an object or some other ‘endomorphism’ (i.e. other arrow whose source and target is the same). \triangle

Definition 8. A *diagram in a category* \mathcal{C} is what is represented by a representational diagram – i.e. is some \mathcal{C} -objects and \mathcal{C} -arrows between them. \triangle

I’m being a little picky in distinguishing the two ideas here, the diagram-as-picture, and the diagram-as-what-is-pictured. But having made the distinction, we will rarely need to fuss about it, and can let context determine a sensible reading of informal claims about diagrams.

An important point is that diagrams (in either sense) needn’t be *full*. That is to say, a diagram-as-a-picture need only show *some* of the objects and arrows in a category; and a diagram-as-what-is-pictured need only be a fragment of the category in question.

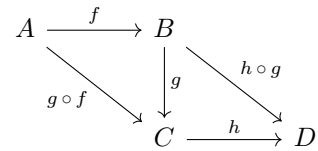
Now, within a drawn diagram, we may be able to follow a directed path through more than two nodes, walking along the connecting drawn arrows (from source to target, of course). So a path in a representational diagram from node A to node E (for example) might look like this

$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D \xrightarrow{j} E$$

And we will call the represented composite arrow $j \circ h \circ g \circ f$ the *composite along the path*. (We know that the composite must exist, and also that because of the associativity of composition we needn’t worry about bracketing here. Indeed, henceforth we freely insert or omit brackets, doing whatever promotes local clarity. And for convenience, we’ll allow ‘composite’ to cover the one-arrow case.) Then we say:

Definition 9. A category diagram *commutes* if for any two directed paths along edges in the diagram from a node X to the node Y , the composite arrow along the first path is equal to the composite arrow along the second path. \triangle

Hence, for example, the associativity law can be represented by saying that the following diagram commutes:



Each triangle commutes by definition of composition; and the commutativity axiom amounts then to the claim that we can paste such triangles together to get a larger commutative diagram.

But note: to say a given diagram commutes is just a vivid way of saying that certain identities hold between composites – it is the identities that matter. And note too that merely drawing a diagram with different routes from e.g. A to D in the relevant category doesn't always mean that we have a *commutative* diagram – the identity of the composites along the paths in each case has to be argued for!

4 Categories beget categories

We have already seen that categories are very plentiful. But we certainly aren't done yet in giving examples of categories. And in this chapter we describe a number of general methods for constructing new categories from old, methods which can then be applied and re-applied to our existing examples to get many more. (We'll meet further construction methods later, but these first ones will be enough to be going on with.)

4.1 Duality

An easy but particularly important way of getting one category from another is to reverse all the arrows. More carefully:

Definition 10. Given a category \mathcal{C} , then its *opposite* or *dual* \mathcal{C}^{op} is the category such that

- (1) The objects of \mathcal{C}^{op} are just the objects of \mathcal{C} again.
- (2) If f is an arrow of \mathcal{C} with source A and target B , then f is also an arrow of \mathcal{C}^{op} but now it is assigned source B and target A .
- (3) Identity arrows remain the same, i.e. $1_A^{op} = 1_A$.
- (4) Composition-in- \mathcal{C}^{op} is defined in terms of composition-in- \mathcal{C} by putting $f \circ^{op} g = g \circ f$. △

It is trivial to check that this definition is in good order and that \mathcal{C}^{op} is indeed a category. And it is trivial to check that $(\mathcal{C}^{op})^{op}$ is \mathcal{C} . So *every* category is the opposite of some category.

Do be careful here, however. Take for example \mathbf{Set}^{op} . An arrow $f: A \rightarrow B$ in \mathbf{Set}^{op} is the same thing as an arrow $f: B \rightarrow A$ in \mathbf{Set} , which is of course a set-function from B to A . But this means that $f: A \rightarrow B$ in \mathbf{Set}^{op} typically *won't* be a function from *its* source to its target – it's an arrow in that direction but usually only a function in the opposite one! (This is one of those cases where talking of 'domains' and 'codomains' instead of 'sources' and 'targets' could initially encourage confusion, since the 'domain' of an arrow in \mathbf{Set}^{op} is its codomain as a function.)

4.2 Subcategories, product and quotient categories

Set^{op} is in fact a very different sort of category to Set ; and indeed, in general, taking the opposite category gives us something essentially new. But not always. Consider the category Rel^{op} , for example, and just remember that every relation comes as one of a pair with its converse or opposite.

Take \mathcal{L} to be the elementary pure language of categories. This will be a two-sorted first-order language with identity, with one sort of variable for objects, $A, B, C \dots$, and another sort for arrows f, g, h, \dots . It has built-in function-expressions ‘ src ’ and ‘ tar ’ (denoting two operations taking arrows to objects), a built-in relation ‘ \dots is the identity arrow for \dots ’, and a two place function-expression ‘ $\dots \circ \dots$ ’ which expresses the function which takes two composable arrows to another arrow.

Definition 11. Suppose φ is a wff of \mathcal{L} . Then its *dual* φ^{op} is the wff you get by (i) swapping ‘ src ’ and ‘ tar ’ and (ii) reversing the order of composition, so ‘ $f \circ g$ ’ becomes ‘ $g \circ f$ ’, etc. △

Now, the claim that \mathcal{C}^{op} is a category just reflects the fact that the duals of the axioms for a category are also axioms. And *that* observation gives us the following *duality principle*:

Theorem 5. *Suppose φ is an \mathcal{L} -sentence (a wff with no free variables) – so φ is a general claim about objects/arrows in an arbitrary category. Then if the axioms of category theory entail φ , they also entail the dual claim φ^{op} .*

Since we are dealing with a first-order theory, syntactic and semantic entailment come to the same, and we can prove the theorem either way:

Syntactic proof. If there’s a first-order proof of φ from the axioms of category theory, then by taking the duals of every wff in the proof we’ll get a proof of φ^{op} from the duals of the axioms of category theory. But those duals of axioms are themselves axioms, so we have a proof of φ^{op} from the axioms of category theory. □

Semantic proof. If φ always holds, i.e. holds in every category \mathcal{C} , then φ^{op} will hold in every \mathcal{C}^{op} – but the \mathcal{C}^{op} s comprise every category again, so φ^{op} also holds in every category. □

The duality principle is very simple but also a hugely labour-saving result; we’ll see this time and time again, starting in the next chapter.

4.2 Subcategories, product and quotient categories

Three familiar ways of getting new widgets from old are by taking subwidgets, forming products of widgets, and quotienting by an equivalence relation. We can do all these with categories.

(a) The simplest way of getting a new category is by slimming down an old one:

Definition 12. Given a category \mathcal{C} , if \mathcal{S} consists of the data

- (1) objects: some or all of the \mathcal{C} -objects,
- (2) arrows: some or all of the \mathcal{C} -arrows,

subject to the conditions

- (3) for each \mathcal{S} -object C , the \mathcal{C} -arrow 1_C is also an \mathcal{S} -arrow,
- (4) for any \mathcal{S} -arrows $f: C \rightarrow D$, $g: D \rightarrow E$, the \mathcal{C} -arrow $g \circ f: C \rightarrow E$ is also an \mathcal{S} -arrow,

then, with composition of arrows in \mathcal{S} defined as in the original category \mathcal{C} , \mathcal{S} is a *subcategory* of \mathcal{C} . △

Plainly, the conditions in the definition – containing identity arrows for the remaining objects and being closed under composition – are there to ensure that the slimmed-down \mathcal{S} is indeed still a category.

Some cases where we prune an existing category will leave us with unnatural constructions of no particular interest. Other cases can be more significant, and indeed we have already met some examples:

- (1) Set is a subcategory of Pfn,
- (2) FinSet is a subcategory of Set,
- (3) Ab is a subcategory of Grp,
- (4) The discrete category on the objects of \mathcal{C} is a subcategory of \mathcal{C} for any category.

So, we can shed objects and/or arrows in moving from a category to a subcategory. In examples (1) and (4) we keep all the objects but shed some or all of the non-identity arrows. But cases (2) and (3) are ones where we drop some objects while keeping all the arrows between those objects retained in the subcategory, and there is a standard label for such cases:

Definition 13. If \mathcal{S} is a subcategory of \mathcal{C} where, for all \mathcal{S} -objects A and B , the \mathcal{S} -arrows from A to B are all the \mathcal{C} -arrows from A to B , then \mathcal{S} is said to be a *full subcategory* of \mathcal{C} . △

We'll meet more cases of full subcategories later.

(b) It is also easy to form products of categories:

Definition 14. If \mathcal{C} and \mathcal{D} are categories, then the product category $\mathcal{C} \times \mathcal{D}$ is such that:

- (1) Its objects are pairs (C, D) where C is a \mathcal{C} -object and D is a \mathcal{D} -object;

4.3 Arrow categories and slice categories

- (2) Its arrows $(f, g): (C, D) \rightarrow (C', D')$ are pairs (f, g) where $f: C \rightarrow C'$ is a \mathcal{C} -arrow and $g: D \rightarrow D'$ is a \mathcal{D} -arrow.
- (3) For each pair (C, D) we define the identity arrow on this object by putting $1_{(C,D)} = (1_C, 1_D)$;
- (4) Composition is defined componentwise in the obvious way: $(f, g) \circ (f', g') = (f \circ_{\mathcal{C}} f', g \circ_{\mathcal{D}} g')$. \triangle

It is trivial to check that this is a category.

- (c) For quotients, we first say:

Definition 15. The relation \sim is a *congruence* on the arrows of the category \mathcal{C} iff it is an equivalence relation which respects composition. That is to say, $f \sim g$ is an equivalence such that (i) if $f \sim g$, then $src(f) = src(g)$ and $tar(f) = tar(g)$, and (ii) if $f \sim g$, then $f \circ h \sim g \circ h$ and $k \circ f \sim k \circ g$ whenever the composites are defined. \triangle

Then things again go as you would expect:

Definition 16. Suppose \mathcal{C} is a category, and suppose \sim is a congruence on its arrows. Then \mathcal{C}/\sim is the category whose objects are the same as those of \mathcal{C} and whose arrows are the \sim -equivalence classes (with such a class having its source and target as an arrow inherited from the arrows in the class). \triangle

We've defined the notion of congruence so that it becomes trivial to check that \mathcal{C}/\sim is indeed a category (assuming that our ambient set theory indeed allows us to form the required equivalence classes).

For a natural example, take the category **Top**; and consider the congruence \sim which holds between two of its arrows, i.e. two continuous maps between spaces, when one map can be continuously deformed into the other, i.e. there is a so-called homotopy between the maps. Then **Top**/ \sim is the important homotopy category **hTop**.

4.3 Arrow categories and slice categories

- (a) For the moment, for future reference, we will mention just two more ways of deriving a new category from an old one. First:

Definition 17. Given a category \mathcal{C} , the derived *arrow category* $\mathcal{C}^{\rightarrow}$ has the following data:

- (1) $\mathcal{C}^{\rightarrow}$'s objects, its first sort of data, are simply the *arrows* of \mathcal{C} ,
- (2) Given $\mathcal{C}^{\rightarrow}$ -objects f_1, f_2 (i.e. \mathcal{C} -arrows $f_1: X_1 \rightarrow Y_1, f_2: X_2 \rightarrow Y_2$), a $\mathcal{C}^{\rightarrow}$ -arrow $f_1 \rightarrow f_2$ is a pair (j, k) of \mathcal{C} -arrows such that the following diagram commutes:

$$\begin{array}{ccc}
 X_1 & \xrightarrow{j} & X_2 \\
 \downarrow f_1 & & \downarrow f_2 \\
 Y_1 & \xrightarrow{k} & Y_2
 \end{array}$$

The identity arrow on $f: X \rightarrow Y$ is defined to be the pair $(1_X, 1_Y)$. And composition of arrows $(j, k): f_1 \rightarrow f_2$ and $(j', k'): f_2 \rightarrow f_3$ is then defined in the obvious way to be $(j' \circ j, k' \circ k): f_1 \rightarrow f_3$ (just think of pasting together two of those commuting squares). \triangle

It is straightforward to check that this definition does indeed characterize a category.

There are moderately fancy examples of arrow categories which do arise tolerably naturally e.g. in topology, but we won't delay over them now. We mention such categories here mainly to reinforce the point that what makes given data count as objects rather than arrows in a category is not a matter of intrinsic nature but of the role they play.

(b) Suppose next that \mathcal{C} is a category, and I a particular \mathcal{C} -object. We next define a new category from \mathcal{C} , the so-called 'slice' category \mathcal{C}/I , where each of the new category's objects involves pairing up one of \mathcal{C} 's objects A with a \mathcal{C} -arrow $f: A \rightarrow I$.

Now, if \mathcal{C}/I 's objects are pairs (A, f) , what can be a \mathcal{C}/I -arrow from (A, f) to (B, g) ? Well, we'll surely need a \mathcal{C} -arrow j which sends A to B . However, not any old arrow $j: A \rightarrow B$ will do: we'll need j to interact appropriately with the arrows f and g .

This leads to the following definition (and to keep things clear but brief, let's continue to use ' \mathcal{C} -arrow' to refer to the old arrows in \mathcal{C} , and reserve plain 'arrow' for the new arrows to be found in \mathcal{C}/I):

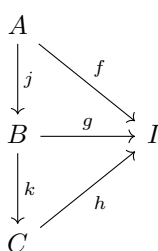
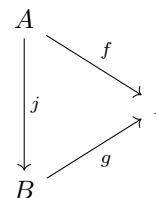
Definition 18. Let \mathcal{C} be a category, and I be a \mathcal{C} -object. Then the category \mathcal{C}/I , the *slice category over I* , has the following data:

- (1) The objects are pairs (A, f) where A is an object in \mathcal{C} , and $f: A \rightarrow I$ is a \mathcal{C} -arrow.
- (2) An arrow from (A, f) to (B, g) is a \mathcal{C} -arrow $j: A \rightarrow B$ such that $g \circ j = f$ in \mathcal{C} .
- (3) The identity arrow on (A, f) is the \mathcal{C} -arrow $1_A: A \rightarrow A$.
- (4) Given arrows $j: (A, f) \rightarrow (B, g)$ and $k: (B, g) \rightarrow (C, h)$, their composition $k \circ j: (A, f) \rightarrow (C, h)$ is the \mathcal{C} -arrow $k \circ j: A \rightarrow C$. \triangle

Of course, we need to check that these data do indeed together satisfy the axioms for constituting a category. So let's do that.

4.3 Arrow categories and slice categories

Take the \mathcal{C} -arrows $f: A \rightarrow I$, $g: B \rightarrow I$. There are corresponding objects (A, f) , (B, g) of \mathcal{C}/I . And the arrows of \mathcal{C}/I from (A, f) to (B, g) will be the \mathcal{C} -arrows like $j: A \rightarrow B$ which make our first diagram commute. (Note an important point: the source and target of j as an arrow in \mathcal{C} are respectively A, B . But the source and target of j as an arrow in the slice category \mathcal{C}/I are respectively (A, f) and (B, g) .)



We now need to confirm that our definition of $k \circ j$ for composing \mathcal{C}/I -arrows works. We are given that $j: A \rightarrow B$ is such that $g \circ j = f$, and likewise that $k: B \rightarrow C$ is such that $h \circ k = g$. So putting things together we get our second commutative diagram. Or in equations, we have $(h \circ k) \circ j = f$ in \mathcal{C} , and therefore $h \circ (k \circ j) = f$. So $(k \circ j)$ does indeed count as an arrow in \mathcal{C}/I from f to h , as we require.

The remaining checks to confirm \mathcal{C}/I satisfies the axioms for being a category are then trivial.

(c) There's a dual notion we can define here, namely the idea of a *co-slice category* I/\mathcal{C} (or the slice category *under* I). This category has as objects pairs (A, f) where this time the arrow goes in the opposite direction, i.e. we have $f: I \rightarrow A$. Then the rest of the definition is as you would predict given our explanation of duality: just go through the definition a slice category reversing arrows and the order of composition. (Check that this works!)

(d) Here are two quick examples of slice and co-slice categories, one of each kind (unlike arrow categories, naturally arising examples are easy to come by):

- (1) Pick a singleton set '1'. We have mentioned before the idea that we can think of any element x of X as an arrow $\vec{x}: 1 \rightarrow X$.

So now think about the co-slice category $1/\mathbf{Set}$. Its objects are the pairs (X, \vec{x}) . We can think of such a pair (X, \vec{x}) as a set with a selected distinguished element x ; in other words, it's a pointed set. And then the arrows $1/\mathbf{Set}$ from some (X, \vec{x}) to (Y, \vec{y}) are all the maps $f: X \rightarrow Y$ in \mathbf{Set} such that $f \circ \vec{x} = \vec{y}$: so we can think of such maps as the maps which preserve basepoints.

Hence $1/\mathbf{Set}$ is (or at least, in some strong sense to be later explained, comes to the same as) the category \mathbf{Set}_* of pointed sets.

- (2) Second, take an n -membered index set $I_n = \{c_1, c_2, c_3, \dots, c_n\}$. Think of the members of I_n as 'colours'. Then a pair (S, c) , where c is a morphism $S \rightarrow I_n$, can therefore be thought of as a set whose members are coloured from that palette of n colours.

Hence we can think of \mathbf{FinSet}/I_n as the category of n -coloured finite sets, exactly the sort of thing that combinatorialists are interested in.

Categories beget categories

More generally, we can think of a slice category \mathbf{Set}/I as a category of ‘indexed’ sets, with I providing the indices.

(e) Defining the objects of a slice category \mathcal{C}/I to be pairs (A, f) where the arrow f has source A and target I involves, you might well think, a certain inelegant redundancy. After all, the first element of the pair is required to be the source of the second: so we wouldn’t lose anything if we defined the object data of \mathcal{C}/I more economically, just to be \mathcal{C} -arrows f with target I .

True. And it is indeed at least as common officially to define slice categories that way. Obviously nothing hangs on this, and we’ll in future treat the objects in slice categories either way, as locally convenient.

5 Kinds of arrows

This chapter characterizes a number of kinds of arrows in terms of how they interact with other arrows in the relevant category. This will give us some elementary but characteristic examples of categorial, arrow-theoretic, (re)definitions of familiar notions.

5.1 Monomorphisms, epimorphisms

(a) Take a set-function $f: A \rightarrow B$ living as an arrow in **Set**: how could we say that it is injective, i.e. one-one, using just category-speak about arrows?

We noted that we can think of elements x of f 's domain A as arrows $\vec{x}: 1 \rightarrow A$ (where 1 is some singleton). Injectiveness then comes to this: $f \circ \vec{x} = f \circ \vec{y}$ implies $\vec{x} = \vec{y}$, for any element-arrows \vec{x}, \vec{y} . Hence if a function is more generally 'left-cancellable' in **Set** – meaning that, for any g, h , $f \circ g = f \circ h$ implies $g = h$ – then it certainly has to be an injection.

Conversely, if f is injective as a set-function, then for all x , $f(g(x)) = f(h(x))$ implies $g(x) = h(x)$ – which is to say that if $f \circ g = f \circ h$ then $g = h$, i.e. f is left-cancellable.

So that motivates introducing a notion with the following definition (the terminology comes from abstract algebra):

Definition 19. An arrow $f: C \rightarrow D$ in the category \mathcal{C} is a *monomorphism* (is *monic*) if and only if it is left-cancellable, i.e. for every 'parallel' pair of maps $g: B \rightarrow C$ and $h: B \rightarrow C$, if $f \circ g = f \circ h$ then $g = h$. \triangle

We have just proved

Theorem 6. *The monomorphisms in Set are exactly the injective functions.*

And the same applies in many, but not all, other categories where arrows are functions. For example, we have:

Theorem 7. *The monomorphisms in Grp are exactly the injective group homomorphisms.*

Proof. We can easily show as before that the injective group homomorphisms are monomorphisms in **Grp**.

Kinds of arrows

For the other direction, suppose that $f: C \rightarrow D$ is a group homomorphism between the groups (C, \cdot, e_C) and (D, \star, e_D) but is *not* an injection.

We must then have $f(c) = f(c')$ for some $c, c' \in C$ where $c \neq c'$. Note that

$$f(c^{-1} \cdot c') = f(c^{-1}) \star f(c') = f(c^{-1}) \star f(c) = f(c^{-1} \cdot c) = f(e_C) = e_D.$$

So $c^{-1} \cdot c'$ is an element in $K \subseteq C$, the kernel of f (i.e. K is the set of elements that f sends to the unit of (D, \star, e_D)). Since $c \neq c'$, we have $c^{-1} \cdot c' \neq e_C$, and hence K has more than one element.

Now define $g: K \rightarrow C$ to be the obvious inclusion map (which send an element of K to the same element of C), while $h: K \rightarrow C$ sends everything to e_C . Since K has more than one element, $g \neq h$. But obviously, $f \circ g = f \circ h$ (both send everything in K to e_D). So the non-injective f isn't left-cancellable.

Hence, contraposing, if f is monic in \mathbf{Grp} it is injective. \square

(b) Next, here is a companion definition:

Definition 20. An arrow $f: C \rightarrow D$ in the category \mathcal{C} is an *epimorphism* (is *epic*) if and only if it is right-cancellable, i.e. for every pair of maps $g: D \rightarrow E$ and $h: D \rightarrow E$, if $g \circ f = h \circ f$ then $g = h$. \triangle

Evidently, the notion of an epimorphism is dual to that of a monomorphism. Hence f is right-cancellable and so epic in \mathcal{C} if and only if it is left-cancellable and hence monic in \mathcal{C}^{op} . And, again predictably, just as monomorphisms in the category \mathbf{Set} are injective functions, we have:

Theorem 8. *The epimorphisms in \mathbf{Set} are exactly the surjective functions.*

Proof. Suppose $f: C \rightarrow D$ is surjective. And consider two functions $g, h: D \rightarrow E$ where $g \neq h$. Then for some $d \in D$, $g(d) \neq h(d)$. But by surjectivity, $d = f(c)$ for some $c \in C$. So $g(f(c)) \neq h(f(c))$, whence $g \circ f \neq h \circ f$. So contraposing, the surjectivity of f in \mathbf{Set} implies that if $g \circ f = h \circ f$, then $g = h$, i.e. f is epic.

Conversely, suppose $f: C \rightarrow D$ is not surjective, so $f[C] \neq D$. Consider two functions $g: D \rightarrow E$ and $h: D \rightarrow E$ which agree on $f[C] \subset D$ but disagree on the rest of D . Then $g \neq h$, even though by hypothesis $g \circ f$ and $h \circ f$ will agree everywhere on C , so f is not epic. Contraposing, if f is epic in \mathbf{Set} , it is surjective. \square

A similar result holds in many other categories, but in §5.3, Ex. (2), we'll encounter a case where we have an epic function which is *not* surjective.

As the very gentlest of exercises, let's add for the record a mini-theorem:

Theorem 9. (1) *Identity arrows are always monic. Dually, they are always epic too.*

(2) *If f, g are monic, so is $f \circ g$. If f, g are epic, so is $f \circ g$.*

(3) *If $f \circ g$ is monic, so is g . If $f \circ g$ is epic, so is f .*

Proof. (1) is trivial.

For (2), we need to show that if $(f \circ g) \circ j = (f \circ g) \circ k$, then $j = k$. So suppose the antecedent. By associativity, $f \circ (g \circ j) = f \circ (g \circ k)$. Whence, assuming f is monic, $g \circ j = g \circ k$. Whence, assuming g is monic, $j = k$. Which establishes that if f and g are monic, so is $(f \circ g)$.

Interchanging f and g , if f and g are monic, so is $(g \circ f)$: applying the duality principle it follows that f and g are epic, so is $(f \circ g)$.

For (3) assume $f \circ g$ is monic. Suppose $g \circ j = g \circ k$. Then $f \circ (g \circ j) = f \circ (g \circ k)$, and hence $(f \circ g) \circ j = (f \circ g) \circ k$, so $j = k$. Therefore if $g \circ j = g \circ k$ then $j = k$; i.e. g is monic. Dually again for epics. \square

(c) We should note a common convention of using special arrows in representational diagrams, a convention which we will follow occasionally but not religiously:

$$f: C \hookrightarrow D \text{ or } C \xrightarrow{f} D \text{ represents a monomorphism } f,$$

$$f: C \twoheadrightarrow D \text{ or } C \xrightarrow{f} D \text{ represents an epimorphism.}$$

As a useful mnemonic (well, it works for me!), just think of the alphabetic proximity of *ML* and of *PR*: a *monomorphism* is *left* cancellable and its representing arrow has an extra fletch on the *left*; while an *epimorphism* is *right* cancellable and its representing arrow has an extra head on the *right*.

5.2 Inverses

(a) We define some more types of arrow:

Definition 21. Given an arrow $f: C \rightarrow D$ in the category \mathcal{C} ,

- (1) $g: D \rightarrow C$ is a *right inverse* of f iff $f \circ g = 1_D$.
- (2) $g: D \rightarrow C$ is a *left inverse* of f iff $g \circ f = 1_C$.
- (3) $g: D \rightarrow C$ is an *inverse* of f iff it is both a right inverse and a left inverse of f . \triangle

Three remarks. First, on the use of ‘left’ and ‘right’. Note that if we represent the situation in (1) like this

$$\begin{array}{ccccc}
 D & \xrightarrow{g} & C & \xrightarrow{f} & D \\
 & & \searrow & \nearrow & \\
 & & & & 1_D
 \end{array}$$

then f ’s right inverse g appears on the left! It is just a matter of convention that we standardly describe handedness by reference to the representation ‘ $f \circ g = 1_D$ ’ rather than by reference to our diagram. (Similarly, of course, in defining left-cancellability, etc.)

Kinds of arrows

Second, note that $g \circ f = 1_C$ in \mathcal{C} iff $f \circ^{op} g = 1_C$ in \mathcal{C}^{op} . So a left inverse in \mathcal{C} is a right inverse in \mathcal{C}^{op} . And vice versa. The ideas of a right inverse and left inverse are therefore, exactly as you would expect, dual to each other; and the idea of an inverse is dual to itself.

Third, if f has a right inverse g , then it is itself a left inverse (of g , of course!). Dually, if f has a left inverse, then it is a right inverse.

It is obvious that an arrow f need not have a left inverse: just consider, for example, those arrows in **Set** which are many-one functions. An arrow f can also have more than one left inverse: for a miniature example in **Set** again, consider $f: \{0, 1\} \rightarrow \{0, 1, 2\}$ where $f(0) = 0, f(1) = 1$. Then the map $g: \{0, 1, 2\} \rightarrow \{0, 1\}$ is a left inverse so long as $g(0) = 0, g(1) = 1$, which leaves us with two choices for $g(2)$, and hence we have two left inverses.

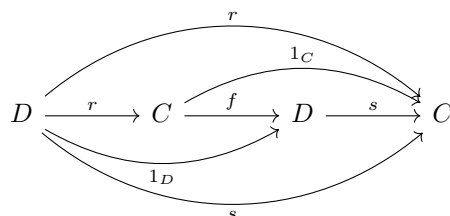
By the duality principle, an arrow can also have zero or many right inverses. However,

Theorem 10. *If an arrow has both a right inverse and a left inverse, then these are the same and are the arrow's unique inverse.*

Proof. Suppose $f: C \rightarrow D$ has right inverse $r: D \rightarrow C$ and left inverse $s: D \rightarrow C$. Then

$$r = 1_C \circ r = (s \circ f) \circ r = s \circ (f \circ r) = s \circ 1_D = s.$$

Or, to put it diagrammatically, the following commutes:



Hence $r = s$ and r is an inverse.

Suppose now that f has inverses r and s . Then in particular, r will be a right inverse and s a left inverse for f , so as before $r = s$. Therefore inverses are unique. \square

(b) By way of an aside, let's remark that just as we can consider a particular monoid as a category, in the same way we can consider a particular group as a category. Take a group (G, \cdot, e) and define \mathcal{G} to be the corresponding category whose sole object is whatever you like, and whose arrows are the elements g of G , with e the identity arrow. Composition of arrows in \mathcal{G} is defined as group-multiplication of elements in G . And since every element in the group has an inverse, it follows immediately that every arrow in the corresponding category has an inverse. So in sum, a group-as-a-category is a category with one object and whose every arrow has an inverse. (And generalizing, a category with perhaps more than one object but whose arrows all still have inverses is called a *groupoid*.)

(c) Now, how does talk of an arrow as a right inverse/left inverse hook up to talk of an arrow as monic/epic?

Theorem 11. (1) *In general, not every monomorphism is a right inverse; and dually, not every epimorphism is a left inverse.*
 (2) *But every right inverse is monic, and every left inverse is epic.*

Proof. (1) can be shown by a toy example. Take the category $\mathbf{2}$ which we met back in §3.6, Ex. (C7) – i.e. take as a category the two-object category which has just one non-identity arrow. That non-identity arrow is trivially monic and epic, but it lacks both a left and a right inverse.

For (2), suppose f is a right inverse for e , which means that $e \circ f = 1$ (suppressing unnecessary labellings of domains and codomains). Now suppose $f \circ g = f \circ h$. Then $e \circ f \circ g = e \circ f \circ h$, and hence $1 \circ g = 1 \circ h$, i.e. $g = h$, so indeed f is monic. Similarly for the dual. \square

So monics need not in general be right inverses nor epics left inverses. But how do things pan out in the particular case of the category \mathbf{Set} ? Here's the answer:

Theorem 12. *In \mathbf{Set} , every monomorphism is a right inverse apart from arrows of the form $\emptyset \rightarrow D$. Also in \mathbf{Set} , the proposition that every epimorphism is a left inverse is (a version of) the Axiom of Choice.*

Proof. Suppose $f: C \rightarrow D$ in \mathbf{Set} is monic. It is therefore one-to-one between C and $f[C]$, so consider a function $g: D \rightarrow C$ that reverses f on $f[C]$ and maps everything in $D - f[C]$ to some particular object in C . Such a g is always possible to find in \mathbf{Set} unless C is the empty set. So $g \circ f = 1_C$, and hence f is a right inverse.

Now suppose $f: C \rightarrow D$ in \mathbf{Set} is epic, and hence a surjection. Assuming the Axiom of Choice, there will be a function $g: D \rightarrow C$ which maps each $d \in D$ to some chosen one of the elements c such that $f(c) = d$ (but note that this time, in the general case, we do have to make an infinite number of choices, picking out one element among the pre-images of d for every $d \in D$: that's why Choice is involved). Given such a function g , $f \circ g = 1_D$, so f is a left inverse.

Conversely, suppose we have a partition of C into disjoint subsets indexed by (exactly) the elements of D . Let $f: C \rightarrow D$ be the function which sends an object in C to the index of the partition it belongs to. f is surjective, hence epic. Suppose f is also a left inverse, so for some $g: D \rightarrow C$, $f \circ g = 1_D$. Then g is evidently a choice function, picking out one member of each partition. So the claim that every epic is a left inverse gives us (one version of) the Axiom of Choice. \square

(d) There is an oversupply of other jargon hereabouts, also in pretty common use. We should note the alternatives for the record.

Assume we have a pair of arrows in opposite directions, $f: C \rightarrow D$, and $g: D \rightarrow C$.

Definition 22. If $g \circ f = 1_C$, then f is also called a *section* of g , and g is a retraction of f . (In this usage, f is a section iff it *has* a retraction, etc.) \triangle

Definition 23. If f has a left inverse, then f is a *split monomorphism*; if g has a right inverse, then g is a *split epimorphism*. (In this usage, we can say e.g. that the claim that every epimorphism splits in **Set** is the categorial version of the Axiom of Choice.) \triangle

Note that Theorem 11 tells us that right inverses are monic, so a split monomorphism is indeed properly called a monomorphism. Dually, a split epimorphism is an epimorphism.

5.3 Isomorphisms

(a) Before we ever encounter category theory, we are familiar with the notion of an isomorphism between structured sets (between groups, between topological spaces, whatever): it's a bijection between the sets which preserves all the structure. In the extremal case, in the category **Set** of sets with no additional structure, the bijections are the arrows which are both monic and epic. Can we generalize from this case and define the isomorphisms of any category to be arrows which are monic and epic there?

No. Isomorphisms properly so called need to have inverses. But being monic and epic doesn't always imply having an inverse. We can use again the toy case of 2, or here's a generalized version of the same idea:

- (1) Take the category \mathcal{S} corresponding to the pre-ordered set (S, \leq) . Then there is at most one arrow between any given objects of \mathcal{S} . But if $f \circ g = f \circ h$, then g and h must share the same object as domain and same object as codomain, hence $g = h$, so f is monic. Similarly f must be epic. But no arrows other than identities have inverses.

The arrows in that example aren't functions, however. So here's a revealing case where the arrows *are* functions but where being monic and epic still doesn't imply having an inverse:

- (2) Consider the category **Mon** of monoids. Among its objects are $\mathcal{N} = (\mathbb{N}, +, 0)$ and $\mathcal{Z} = (\mathbb{Z}, +, 0)$ – i.e. the monoid of natural numbers equipped with addition and the monoid of positive and negative integers equipped with addition.

Let $i: \mathcal{N} \rightarrow \mathcal{Z}$ be the map which sends a natural number to the corresponding positive integer. This map obviously does not have an inverse in **Mon**. We can show, however, that it is both monic and epic.

First, suppose $\mathcal{M} = (M, \cdot, 1_M)$ is some monoid and we have two arrows $g, h: \mathcal{M} \rightarrow \mathcal{N}$, where $g \neq h$. There is then some element $m \in M$ such that the natural numbers $g(m)$ and $h(m)$ are different, which means that the

corresponding integers $i(g(m))$ and $i(h(m))$ are different, so $i \circ g \neq i \circ h$. Contraposing, this means i is monic in the category.

Second, again take a monoid \mathcal{M} and this time consider any two monoid homomorphisms $g, h: \mathbb{Z} \rightarrow \mathcal{M}$ such that $g \circ i = h \circ i$. Then g and h must agree on all integers from zero up. But then note

$$\begin{aligned} g(-1) &= g(-1) \cdot 1_M = g(-1) \cdot h(0) = g(-1) \cdot h(1 + -1) \\ &= g(-1) \cdot h(1) \cdot h(-1) = g(-1) \cdot g(1) \cdot h(-1) \\ &= g(-1 + 1) \cdot h(-1) = g(0) \cdot h(-1) = 1_M \cdot h(-1) = h(-1). \end{aligned}$$

But if $g(-1) = h(-1)$, then

$$g(-2) = g(-1 + -1) = g(-1) \cdot g(-1) = h(-1) \cdot h(-1) = h(-1 + -1) = h(-2),$$

and the argument iterates, so we have $g(j) = h(j)$ for all $j \in \mathbb{Z}$, positive and negative. Hence $g = h$ and i is right-cancellable, i.e. epic.

So in sum: we can't define an isomorphism as an epic monic if isomorphisms are to have the essential feature of invertibility.

(b) What to do? Build in invertibility from the start, and say:

Definition 24. An *isomorphism* (in category \mathcal{C}) is an arrow which has an inverse. We conventionally represent isomorphisms by decorated arrows, thus: $\xrightarrow{\sim}$. \triangle

From what we have already seen, we know or can immediately check that

Theorem 13. (1) *Identity arrows are isomorphisms.*

(2) *An isomorphism $f: C \xrightarrow{\sim} D$ has a unique inverse which we can call $f^{-1}: D \xrightarrow{\sim} C$, such that $f^{-1} \circ f = 1_C$, $f \circ f^{-1} = 1_D$, $(f^{-1})^{-1} = f$, and f^{-1} is also an isomorphism.*

(3) *If f and g are isomorphisms, then $g \circ f$ is an isomorphism if it exists, whose inverse will be $f^{-1} \circ g^{-1}$.*

Let's immediately give some simple examples of isomorphisms in different categories:

- (1) In **Set**, the isomorphisms are the bijective set-functions.
- (2) In **Grp**, the isomorphisms are the bijective group homomorphisms.
- (3) In **Vect_k**, the isomorphisms are invertible linear maps.
- (4) In a group treated as a category, every arrow is an isomorphism.
- (5) But as we noted, in a pre-order category, the only isomorphisms are the identity arrows.

Kinds of arrows

(c) Isomorphisms are monic and epic by Theorem 11. And we now know that arrows which are monic and epic need not be isomorphisms. However, we do have this:

Theorem 14. *If f is both monic and split epic (or both epic and split monic), then f is an isomorphism.*

Proof. If f is a split epimorphism, it has a right inverse, i.e. there is a g such that $f \circ g = 1$. Then $(f \circ g) \circ f = f$, whence $f \circ (g \circ f) = f \circ 1$. Hence, given that f is also mono, $g \circ f = 1$. So g is both a left and right inverse for f , i.e. f has an inverse. Dually for the other half of the theorem. \square

We will also mention another easy result in the vicinity:

Theorem 15. *If f and g are both monic arrows with the same target, and each factors through the other, i.e. there are i, j such that $f = g \circ i$ and $g = f \circ j$, then the factors i and j are isomorphisms and inverse to each other.*

In other words, if each of the triangles in the following diagram commutes, then so does the whole diagram:

$$\begin{array}{ccc}
 X & \begin{array}{c} \xrightarrow{i} \\ \xleftarrow{j} \end{array} & Y \\
 & \begin{array}{c} \searrow f \\ \swarrow g \end{array} & \\
 & & Z
 \end{array}$$

Proof. We have $f \circ 1_X = f = g \circ i = f \circ j \circ i$. Hence, since f is monic, $j \circ i = 1_X$. Similarly, $i \circ j = 1_Y$. So i and j are each other's two-sided inverse, and both are isomorphisms. \square

(d) Finally, we should mention a bit of standard terminology:

Definition 25. A category \mathcal{C} is *balanced* iff every arrow which is both monic and epic is in fact an isomorphism.

Then we have seen that some categories like **Set** are balanced, while others like **Mon** are not. **Top** is another example of an unbalanced category.

5.4 Isomorphic objects

(a) Finally in this chapter, we introduce another key notion:

Definition 26. If there is an isomorphism $f: C \xrightarrow{\sim} D$ in \mathcal{C} then the objects C, D are said to be *isomorphic* in \mathcal{C} , and we write $C \cong D$. \triangle

From the ingredients of Theorem 13, we immediately get the desirable result that

Theorem 16. *Isomorphism between objects in a category \mathcal{C} is an equivalence relation.*

An isomorphism between objects in a category also induces a bijection between the arrows to (or from) those objects:

Theorem 17. *If $C \cong D$ in \mathcal{C} , then there is a one-one correspondence between arrows $X \rightarrow C$ and $X \rightarrow D$ for all objects X in \mathcal{C} , and likewise a one-one correspondence between arrows $C \rightarrow X$ and $D \rightarrow X$.*

Proof. If $C \cong D$ then there is an isomorphism $j: C \xrightarrow{\sim} D$. Consider the map which sends an arrow $f: X \rightarrow C$ to $\hat{f} = j \circ f: X \rightarrow D$. This map $f \mapsto \hat{f}$ is injective (for $\hat{f} = \hat{g}$ entails $j^{-1} \circ \hat{f} = j^{-1} \circ \hat{g}$ and hence $f = g$). It is also surjective (for any $g: X \rightarrow D$, put $f = j^{-1} \circ g$ then $\hat{f} = g$). Similarly for the other part. \square

(b) We might wonder how far the notion of isomorphism between objects actually captures the idea of two objects amounting to the same as far as their ambient category is concerned.

We mentioned before the example where we have, living in \mathbf{Grp} , lots of instances of a Klein four-group which are group-theoretically indiscernible by virtue of being isomorphic (indeed, between any two instances, there is a unique isomorphism). And yes, we then cheerfully talk about *the* Klein four-group.

There is a real question, however, about just what this way of talking amounts to, when we seemingly identify isomorphic objects. Some claim that category theory itself throws a lot of light on this very issue (see e.g. Mazur 2008). And certainly, category theory typically doesn't care about distinguishing isomorphic objects in a category. But note that it would, for example, initially strike us as odd to say that, just because all the instances of singleton sets are isomorphic (indeed, between any two instances, there is a unique isomorphism), we can always happily talk about *the* singleton. There are contexts where any singleton will do, as for example when we associate elements x of a set X with arrows $\bar{x}: 1 \rightarrow X$. But in other contexts, the pairwise distinctness of singletons is important, e.g. when we treat $\{\emptyset\}, \{\{\emptyset\}\}, \{\{\{\emptyset\}\}\}, \{\{\{\{\emptyset\}\}\}\}, \dots$ as a sequence of distinct sets in one possible construction (Zermelo's) for the natural numbers.

But we can't delay to explore this issue further at the moment: we are just flagging up that there are questions we'll at some point want to discuss around and about the idea of isomorphism-as-sameness.

6 Initial and terminal objects

Any introduction to the theory of categories is going to start with the definition of a category, a catalogue of examples, an explanation of duality (and perhaps of other ways of getting new categories from old), and a categorial definition of isomorphisms and other kinds of arrow. But now the possible onward paths begin to fork. We could take the steep ascent and start talking straight away about functors, i.e. maps between categories, and then quickly climb up again to discuss transformations between these functors. This can be very illuminating; but it can also make things unnecessarily tough for the beginner. So instead we will set out by taking a more pedestrian route through the foothills for the moment, and over the following chapters think a lot more about what happens *inside* a category, before we begin to consider relations *between* categories in Chapter 15.

Now, when we defined an isomorphism, we characterized a type of arrow not by (so to speak) its internal workings – not by how it operated on its source domain – but by reference to its interaction with another arrow, its inverse. This is entirely typical of a category-theoretic (re)definition of a familiar notion: we look for similarly external, relational, characterizations of arrows and/or structured objects.

Here is Awodey, offering some similarly arm-waving

... remarks about category-theoretical definitions. By this I mean characterizations of properties of objects and arrows in a category in terms of other objects and arrows only, that is, in the language of category theory. Such definitions may be said to be abstract, structural, operational, relational, or external (as opposed to internal). The idea is that objects and arrows are determined by the role they play in the category via their relations to other objects and arrows, that is, by their position in a structure and not by what they ‘are’ or ‘are made of’ in some absolute sense. (Awodey, 2006, p. 25)

We proceed, then, to give some further examples of external category-theoretic definitions of a range of familiar notions. A prime exhibit will be the illuminating treatment of products, starting in the next chapter. In this chapter, however, we warm up by considering a particularly simple pair of cases.

6.1 Initial and terminal objects, definitions and examples

Definition 27. The object I is an *initial* object of the category \mathcal{C} iff, for every \mathcal{C} -object X , there is a unique arrow $I \rightarrow X$.

Dually, the object T is a *terminal* object of \mathcal{C} iff, for every \mathcal{C} -object X , there is a unique arrow $X \rightarrow T$.¹ \triangle

It is common to use the likes of ‘ $! : I \rightarrow X$ ’ or ‘ $! : X \rightarrow T$ ’ for the unique arrow from an initial object or to a terminal object. If we want explicitly to indicate e.g. the source of such a unique arrow to a terminal object, we can write $!_X$.

Some examples:

- (1) In the poset (\mathbb{N}, \leq) thought of as a category, zero is trivially the unique initial object and there is no terminal object. The poset (\mathbb{Z}, \leq) has neither initial nor terminal objects.
More generally, a poset- (S, \leq) -treated-as-a-category has an initial object iff the poset has a minimum, an object which \leq -precedes all the others. Dually for terminal objects/maxima.
- (2) In **Set**, the empty set is an initial object (cf. the comment in §3.5).
And any singleton set $\{\star\}$ is a terminal object. (For if X has members, there’s a unique **Set**-arrow which sends all the members to \star ; while if X is empty, then there’s a unique **Set**-arrow to any set, including $\{\star\}$).
- (3) In **Set** $_{\star}$ – the category of pointed sets, non-empty sets equipped with a distinguished member – each singleton is both initial *and* terminal. (A one-membered pointed set’s only member has to be the distinguished member. Arrows in **Set** $_{\star}$ are functions which map distinguished elements to distinguished element. Hence there can be one and only one arrow from a singleton pointed set to some other pointed set.)
- (4) In **Poset**, the empty poset is initial, and any singleton equipped with the only possible order relation on it (the identity relation!) is terminal.
- (5) In **Rel**, the category of sets and relations, the empty set is both the sole initial and sole terminal object.
- (6) In **Top**, the empty set (considered as a trivial topological space) is the initial object. Any one-point singleton space is a terminal object.
- (7) In **Grp**, the trivial one-element group is an initial object (a group has to have at least one object, the identity; now recall that a group homomorphism sends identity elements to identity elements; so there is one and only one homomorphism from the trivial group to any given group G). The same one-element group is also terminal.

¹Warning: some call terminal objects *final*; and then that frees up ‘terminal’ to mean *initial* or *final*.

Initial and terminal objects

- (8) In the category **Bool**, the trivial one-object algebra is terminal. While the two-object algebra on $\{0, 1\}$ familiar from propositional logic is initial – for a homomorphism of Boolean algebras from $\{0, 1\}$ to B must send 0 to the bottom object of B and 1 to the top object, and there’s a unique map that does that.
- (9) Recall: in the slice category \mathcal{C}/X an object (defined the more economical way) is a \mathcal{C} -arrow like $f: A \rightarrow X$, and an arrow from $f: A \rightarrow X$ to $g: B \rightarrow X$ is a \mathcal{C} -arrow $j: A \rightarrow B$ such that $g \circ j = f$ in \mathcal{C} . Consider the \mathcal{C}/X -object $1_X: X \rightarrow X$. A \mathcal{C}/X arrow from $f: A \rightarrow X$ to 1_X is a \mathcal{C} -arrow $j: A \rightarrow X$ such that $1_X \circ j = f$, i.e. such that $j = f$ – which exists and is unique! So 1_X is terminal in \mathcal{C}/X .

Such various cases show that a category may have zero, one or many initial objects, and (independently of that) may have zero, one or many terminal objects. Further, an object can be both initial and terminal.

There is, incidentally, a standard bit of jargon for the last case:

Definition 28. An object O in the category \mathcal{C} is a *null object* of the category \mathcal{C} iff it is both initial and terminal. \triangle

6.2 Uniqueness up to unique isomorphism

A category \mathcal{C} , to repeat, may have no initial objects, or only one, or have many. However, we do have the following key result:

Theorem 18. *Initial objects, when they exist, are ‘unique up to unique isomorphism’: i.e. if the \mathcal{C} -objects I and J are both initial in the category \mathcal{C} , then there is a unique isomorphism $f: I \xrightarrow{\sim} J$ in \mathcal{C} . Dually for terminal objects.*

Further, if I is initial and $I \cong J$, then J is also initial. Dually for terminal objects.

Proof. Suppose I and J are both initial objects in \mathcal{C} . By definition there must be unique \mathcal{C} -arrows $f: I \rightarrow J$, and $g: J \rightarrow I$. Then $g \circ f$ is an arrow from I to itself. Another arrow from I to itself is the identity arrow 1_I . But since I is initial, there can only be one arrow from I to itself, so $g \circ f = 1_I$. Likewise $f \circ g = 1_J$. Hence the unique arrow f has a two-sided inverse and is an isomorphism. (Note this pattern of argument: we’ll be using it a lot!)

Now suppose I is initial and $I \cong J$, so that there is an isomorphism $i: I \rightarrow J$. Then for any X , there is a unique arrow $f: I \rightarrow X$, and hence there is an arrow $f \circ i^{-1}: J \rightarrow X$. Assume we also have $g: J \rightarrow X$. Then $g \circ i: I \rightarrow X$, and so $g \circ i = f$, hence $g = f \circ i^{-1}$. In sum, for any X there is a unique arrow from J to X , thus J is also initial.

Duals of these two arguments deliver, of course, the dual results. \square

It is standard to introduce notation for an arbitrary initial and terminal objects (since categorically, we usually won't care about distinctions among instances):

Definition 29. We use '0' to denote an initial object of \mathcal{C} (assuming one exists), and likewise '1' to denote a terminal object. \triangle

Note that in **Set**, 0 is \emptyset , the only initial object – and \emptyset is also the von Neumann ordinal 0. While the von Neumann ordinal 1 is $\{\emptyset\}$, i.e. a singleton, i.e. a terminal object 1. Which perhaps excuses the recycling of the notation.

By the way, null objects (objects which are both initial and terminal) are often alternatively called 'zero' objects. But that perhaps doesn't sit happily with using '0' for an initial object: for 0 (in the sense of an initial object) typically isn't a zero (in the sense of null) object. Hence our preference for 'null'.

6.3 Elements and generalized elements

(a) Consider the category **Set** again. As we have remarked before, arrows $\vec{x}: 1 \rightarrow X$ correlate one-to-one with elements $x \in X$: so in **Set** we can think of talk of such arrows $\vec{x}: 1 \rightarrow X$ as the categorial version of talking of members of X . We now carry this idea over to other categories more generally:

Definition 30. In a category \mathcal{C} with a terminal object 1, an *element* or *point* of the \mathcal{C} -object X is an arrow $f: 1 \rightarrow X$.² \triangle

We immediately see, however, that in categories \mathcal{C} other than **Set**, these 'elements' $1 \rightarrow X$ won't always line up nicely with the elements of X in the intuitive sense. In **Grp**, for example, a homomorphism from 1 (the one-element group) to a group X has to send the only group element of 1 to the identity element e of X : so there is only one possible homomorphism $\vec{e}: 1 \rightarrow X$, independently of how many elements there are in the group X .

We can put this last observation in more categorial terms. First, some standard terminology:

Definition 31. Suppose the category \mathcal{C} has a terminal object. And suppose that for any objects X, Y in \mathcal{C} , and parallel arrows $f, g: X \rightarrow Y$, $f = g$ if for all $\vec{x}: 1 \rightarrow X$, $f \circ \vec{x} = g \circ \vec{x}$. Then \mathcal{C} is said to be *well-pointed*. \triangle

Then **Set** is, in this sense, well-pointed. There are enough elements-as-arrows to ensure that parallel arrows with domain X which act identically on all relevant elements of X are indeed identical. By contrast, we have just noted that **Grp** is not well-pointed. Take any two group homomorphisms $f, g: X \rightarrow Y$ where $f \neq g$: for all possible $\vec{e}: 1 \rightarrow X$, both $f \circ \vec{e}$ and $g \circ \vec{e}$ must send the sole member of 1 to the identity element of the group Y , so are equal.

²Other standard terminology for such an element is 'global element', picking up from a paradigm example in topology – but we won't fuss about that.

Initial and terminal objects

(b) Our definition of well-pointedness invokes a choice of the terminal object 1 in terms of which we define elements $\vec{x}: 1 \rightarrow X$. But whether a category is well-pointed doesn't actually depend on that choice:

Theorem 19. *Take two terminal objects 1 and $1'$ and define two different types of elements of X in \mathcal{C} as arrows $1 \rightarrow X$ and $1' \rightarrow X$. \mathcal{C} is well-pointed with respect to elements of the first kind iff it is well-pointed with respect to elements of the second kind.*

Proof. We need only prove one direction. Since 1 and $1'$ are terminal, there is a unique isomorphism $i: 1' \rightarrow 1$, and we can set up a one-one correspondence between elements $\vec{x}: 1 \rightarrow X$ and $\vec{x}': 1' \rightarrow X$ by putting $\vec{x}' = \vec{x} \circ i$.

Assume \mathcal{C} is well-pointed with respect to elements of the first kind. Then, for all $f, g: X \rightarrow Y$, if $f \circ \vec{x}' = g \circ \vec{x}'$, then $f \circ \vec{x} = f \circ \vec{x}' \circ i^{-1} = g \circ \vec{x}' \circ i^{-1} = g \circ \vec{x}$, and therefore $f = g$.

That proves well-pointedness with respect to the second sort of element. \square

(c) We have just seen that, even when arrows in a category are functions, acting the same way on elements (in the sense of Defn. 30) need not imply being the same arrow. Can we generalize the notion of an element so that acting the same way on generalized elements *does* imply being the same arrow?

Well, suppose we say:

Definition 32. A *generalized element* (of shape S) of the object X in \mathcal{C} is an arrow $e: S \rightarrow X$. \triangle

Generalized elements give us more ways of interacting with the data of a category than the original point elements. And now we indeed have

Theorem 20. *Parallel arrows in a category \mathcal{C} are identical if and only if they act identically on all generalized elements.*

Proof. If $f, g: X \rightarrow Y$ act identically on *all* generalized elements, they act identically on $1_X: X \rightarrow X$: therefore $f \circ 1_X = g \circ 1_X$, and so $f = g$. The converse is trivial. \square

(d) A final remark. Note that

Theorem 21. *Point elements $\vec{x}: 1 \rightarrow X$ in a category are monic.*

Proof. Suppose $\vec{x} \circ f = \vec{x} \circ g$; then, for the compositions to be defined and equal, both f and g must be morphisms $Y \rightarrow 1$, for the same Y . Hence $f = g$ since 1 is terminal. \square

Obviously, in most categories, not all *generalized* elements $e: S \rightarrow X$ will be monic. The special monic case will, however, turn out to be significant: see §12.1.

7 Products introduced

Our discussion of the notions of initial and terminal objects provides an introduction to a number of categorial themes which will now keep recurring in rather more exciting contexts, starting in this chapter where we introduce our next main topic, products.

We are familiar with constructing products for all kinds of widgets. The paradigm case, of course, is in **Set** where we take sets X and Y and form their Cartesian product, the set of ordered pairs of their elements. But what *are* ordered pairs? We'll start by considering this basic question as our route in to a categorial treatment of products.

7.1 Real pairs, virtual pairs

A word of caution first. We have fallen into a familiar modern practice of using a single notation for talking about pairs in two different senses. I didn't want to pause distractingly to remark on this before: but we should now draw an important distinction relevant to our current concerns.

- (1) We have, as is standard, used parentheses as in (x, y) or (f, g) to refer to ordered pairs, where an ordered pair is to be thought of as a *single* object.

Here, the parentheses do essential work, expressing constructors taking two given items and outputting a pair-object. In other words, the expression $(\dots, _)$, with its two slots waiting to be filled, here serves as a two-place function expression, a handy formal substitute for the expression 'the ordered pair whose first member is \dots and whose second member is $_$ '.

- (2) But we have also used parentheses in contexts where we can take them as providing no more than helpful punctuation. For example, when talking informally of the pre-ordered set (S, \leq) , we are talking about a pair only in the sense of talking of *two* things: we are referring to the set S and to the ordering \leq defined over S , and we are not – or at least, not straight off – referring to some further pair-object.

For example, if we talk of a function f as 'a monotone map between the posets (S, \leq) and (T, \sqsubseteq) ', this can be unpacked into talk of a set-function

Products introduced

$f: S \rightarrow T$ which is such that $x \leq y$ implies $f(x) \sqsubseteq f(y)$ – so here, the superficial appearance of reference to a pair-object can be translated away. Likewise for other elementary contexts where we talk of posets.

Now here's a nice question: in which contexts does an apparent reference to pairs really commit us to real pair-objects as a single entities; and when can the apparent reference be translated away, so we are at best only countenancing merely virtual pair-objects?

For example, perhaps at some point we do need to treat a poset as if it is a single pair-object over and above the relevant set and order-relation. But when? It might be suggested that when we start talking about the category *Poset* whose objects are posets, then *that* commits us to thinking of the likes of ' (S, \leq) ' as referring to single objects. But we must be careful here not just to rely on a pun on 'object'. After all, if the objects in a category (in the sense of the first kind of data for the category) can already be as diverse as objects (in the logical sense), relations, functions, arrows from other categories, etc., why shouldn't they also be plural, with each *Poset*-object' being in fact two items, a set and an order relation?

Fortunately we don't need to tangle with such questions of logical grammar yet. For the moment, we flag that there are non-trivial issues lurking here, and merely say this. It may be that at some point we do need to start the likes of ' (S, \leq) ', ' $(M, \cdot, 1_M)$ ', etc., as referring to single objects, over and above the relevant sets and relations/functions, i.e. treat them as denoting real rather than virtual pairs or triples, etc. But the stronger interpretation need not be understood as built into our notation from the very start. *The principle should always be: read notations such as ' (S, \leq) ', ' $(M, \cdot, 1_M)$ ', etc., as noncommittally as possible.*

In this chapter, however, we are going to be concerned with cases where we indeed want to deal with ordered-pairs-as-single-objects. But what objects are they?

7.2 Pairing schemes

(a) Suppose for a moment that we are working in a theory of arithmetic and we need to start considering ordered pairs of natural numbers. Perhaps we want to go on to use such pairs in constructing integers or rationals.

Well, we can easily handle such pairs of natural numbers as single objects, and without taking on any new commitments, by using *code-numbers*. For example, if we want a bijective coding between pairs of naturals and all the numbers, we could adopt the scheme of coding the ordered pair (m, n) by the single number $\langle m, n \rangle = \{(m+n)^2 + 3m + n\}/2$. Or, if we don't insist on every number coding a pair, we could adopt the simpler policy of using $\langle m, n \rangle =_{\text{def}} 2^m 3^n$, which allows simpler decoding functions for extracting m and n from $\langle m, n \rangle$. Relative to a given coding scheme, we can call such code-numbers $\langle m, n \rangle$ *pair-numbers*. Or,

by a slight abuse of terminology, we can call them simply *pairs*, and we can refer to m as the first element of the pair, and n as the second element.

Why should this way of handling ordered pairs of natural numbers be regarded as somehow inferior to other, albeit more familiar, coding devices such as explicitly set-theoretic ones? Well, it might be said that (i) a single pair-number is really neither ordered nor a twosome; (ii) while a number m is a member of (or is one of) the pair of m with n , a number can't be a genuine member of a pair-number $\langle m, n \rangle$; and in any case (iii) such a coding scheme is pretty arbitrary (e.g. we could equally well have used $3^m 5^n$ as a code for the pair m, n).

Which is all true. But we can lay *exactly* analogous complaints against e.g. the familiar Kuratowski definition of ordered pairs that we all know and love. This treats the ordered pair of m with n as the set $\langle m, n \rangle_K = \{\{m\}, \{m, n\}\}$. But (i) that set is not intrinsically ordered (after all, it is a *set*!), nor is it always two-membered (consider the case where $m = n$). (ii) Even when it is a twosome, its members are not the members of the pair: in standard set theories, m cannot be a member of $\{\{m\}, \{m, n\}\}$. And (iii) the construction again involves pretty arbitrary choices: thus $\{\{n\}, \{m, n\}\}$ or $\{\{\{m\}\}, \{\{m, n\}\}\}$ etc., etc., would have done just as well. On these counts, at any rate, coding pairs of numbers by using pair-numbers involves no worse a trick than coding them using Kuratowski's standard gadget.

There is indeed a rather neat symmetry between the adoption of pair numbers as representing ordered pairs of numbers and another very familiar procedure adopted by the enthusiast for working in standard ZFC. For remember that standard ZFC knows only about pure sets. So to get natural numbers into the story at all – and hence to get Kuratowski pair-sets of natural numbers – the enthusiast for sets has to choose some convenient sequence of sets to implement the numbers (or to ‘stand proxy’ for numbers, ‘simulate’ them, ‘play the role’ of numbers, or even ‘define’ them – whatever your favourite way of describing the situation is). But someone who, for her purposes, has opted to play the game this way, treats pure sets as basic and is dealing with natural numbers by selecting some convenient sets to implement them, is hardly in a position to complain about someone else who, for his purposes, goes in the opposite direction and treats numbers as basic, and deals with ordered pairs of numbers by choosing some convenient code-numbers to implement *them*. Both theorists are in the implementation game.

It might be retorted that the Kuratowski trick at least has the virtue of being an all-purpose device, available not just when you want to talk about pairs of *numbers*, while e.g. the powers-of-primes coding is of much more limited use. Again true. Similarly you can use sledgehammers to crack all sorts of things, while nutcrackers are only useful for dealing with nuts. But that's not particularly to the point if it happens to be nuts you currently want to crack, efficiently and with light-weight resources. If we want to implement pairs of numbers without ontological inflation – say in pursuing the project of ‘reverse mathematics’ (with its eventual aim of exposing the minimum commitments required for e.g.

Products introduced

doing classical analysis) – then pair-numbers are exactly the kind of thing we need.

(b) Pair-numbers and Kuratowski pairs belong to two different schemes for pairing up numbers, each of which works (though a particular surrounding context might lead us to prefer one to the other). So what does it take to have such a workable scheme for pairing numbers with numbers, or more generally to have a scheme for pairing up X s with Y s?

Evidently, we need some objects O to serve as ordered pairs, a pairing function that sends a given x from the X s and a given y from the Y s to a particular pair-object o , and (of course!) a couple of functions which allow us to recover x and y from o . And the point suggested by the case of rival pairing schemes for numbers is that maybe we shouldn't care too much about the 'internal' nature of the objects O , so long as we associate them with suitable pairing and unpairing functions which fit together in the right way (for example, pairing and then unpairing gets us back to where we started).

Which motivates the following general definition (where we now use the informal set idiom because of its familiarity, though we could recast this by continuing to use plural talk of the X s rather talk of the set X , etc.):

Definition 33. Suppose X , Y and O are sets of objects (these can be the same or different). Let $pr: X, Y \rightarrow O$ be a two-place function, while $\pi_1: O \rightarrow X$, and $\pi_2: O \rightarrow Y$, are one-place functions. Then $[O, pr, \pi_1, \pi_2]$ form a pairing scheme for X with Y iff

- (a) $(\forall x \in X)(\forall y \in Y)(\pi_1(pr(x, y)) = x \wedge \pi_2(pr(x, y)) = y)$,
- (b) $(\forall o \in O) pr(\pi_1 o, \pi_2 o) = o$.

The members of O will be said to be the *pair-objects* of the pairing scheme, with pr the associated *pairing function*, while π_1 and π_2 are unpairing or *projection* functions. △

Evidently, if O is the set of naturals of the form $2^m 3^n$ and $pr(m, n) = 2^m 3^n$, with $\pi_1 o$ ($\pi_2 o$) returning the exponent of 2 (3) in the factorization of o , then $[O, pr, \pi_1, \pi_2]$ form a pairing scheme for \mathbb{N} with \mathbb{N} . And if O' is the set of Kuratowski pairs $\{\langle m, n \rangle_K \mid m, n \in \mathbb{N}\}$, with $pr'(m, n) = \langle m, n \rangle_K$, and π_1 (π_2) taking a pair $\langle m, n \rangle_K$ and returning its first (second) element, then $[O', pr', \pi'_1, \pi'_2]$ form another pairing scheme for \mathbb{N} with \mathbb{N} .

By the way, in accord with our maxim in §7.1, don't over-interpret the square brackets in the definition: they need be read as no more than punctuation. After all, we are in the business of characterizing ordered-pairs-as-single-objects; so we certainly don't want to presuppose e.g. that we already know about ordered-quadruples-as-single-objects!

Two simple facts about pairing schemes:

Theorem 22. *If $[O, pr, \pi_1, \pi_2]$ is a pairing scheme, then (i) different pairs of objects are sent by pr to different pair-objects, and (ii) pr , π_1 and π_2 are all surjective.*

Proof. For (i) suppose $pr(x, y) = pr(x', y')$. Then by condition (a) on pairing schemes, $x = \pi_1(pr(x, y)) = \pi_1(pr(x', y')) = x'$, and likewise $y = y'$.

For (ii) it is immediate that pr is surjective by (b). The projection function π_1 is surjective because, given $x \in X$, we can take any $y \in Y$ and put $o = pr(x, y)$, and then by (a), $x = \pi_1 o$. Similarly for π_2 . \square

As we'd also expect, a given pairing function fixes the two corresponding projection functions, and vice versa, in the following sense:

Theorem 23. (1) *If $[O, pr, \pi_1, \pi_2]$ and $[O, pr, \pi'_1, \pi'_2]$ are both pairing schemes for X with Y , then $\pi_1 = \pi'_1$ and $\pi_2 = \pi'_2$.*

(2) *If $[O, pr, \pi_1, \pi_2]$ and $[O, pr', \pi_1, \pi_2]$ are both pairing schemes for X with Y , then $pr = pr'$.*

Proof. For (1), take any $o \in O$. There is some (unique) x, y such that $o = pr(x, y)$. Hence, applying (a) to both schemes, $\pi_1 o = x = \pi'_1 o$. Hence $\pi_1 = \pi'_1$, and similarly $\pi_2 = \pi'_2$.

For (2), take any $x \in X$, $y \in Y$, and let $pr(x, y) = o$, so $\pi_1 o = x$ and $\pi_2 o = y$. Then by (b) applied to the second scheme, $pr'(\pi_1 o, \pi_2 o) = o$. Whence $pr'(x, y) = pr(x, y)$. \square

Further, there is a sense in which all schemes for pairing X with Y are equivalent up to isomorphism. More carefully,

Theorem 24. *If $[O, pr, \pi_1, \pi_2]$ and $[O', pr', \pi'_1, \pi'_2]$ are both schemes for pairing X with Y , then there is a unique bijection $f: O \rightarrow O'$ such that for all $x \in X, y \in Y$, $pr'(x, y) = f(pr(x, y))$.*

Putting it another way, there is a unique bijection f such that, if we pair x with y using pr (in the first scheme), use f to send the resulting pair-object o to o' , and then retrieve elements using π'_1 and π'_2 (from the second scheme), we get back to the original x and y .

Proof. Define $f: O \rightarrow O'$ by putting $f(o) = pr'(\pi_1 o, \pi_2 o)$. Then it is immediate that $f(pr(x, y)) = pr'(x, y)$.

To show that f is injective, suppose $f(o) = f(o')$, for $o, o' \in O$. Then we have $pr'(\pi_1 o, \pi_2 o) = pr'(\pi_1 o', \pi_2 o')$. Apply π'_1 to each side and then use principle (a), and it follows that $\pi_1 o = \pi_1 o'$. And likewise $\pi_2 o = \pi_2 o'$. Therefore $pr(\pi_1 o, \pi_2 o) = pr(\pi_1 o', \pi_2 o')$. Whence by condition (b), $o = o'$.

To show that f is surjective, take any $o' \in O'$. Then put $o = pr(\pi'_1 o', \pi'_2 o')$. By the definition of f , $f(o) = pr'(\pi_1 o, \pi_2 o)$; plugging the definition of o twice into the right hand side and simplifying using rules (a) and (b) confirms that $f(o) = o'$.

Products introduced

So f is a bijection with the right properties. And since every $o \in O$ is $pr(x, y)$ for some x, y , the requirement that $f(pr(x, y)) = pr'(x, y)$ fixes f uniquely. \square

(c) Here's another simple theorem, to motivate the final definition in this section:

Theorem 25. *Suppose X, Y, O are sets of objects, and the functions $\pi_1: O \rightarrow X$, $\pi_2: O \rightarrow Y$ are such that there is a unique two-place function $pr: X, Y \rightarrow O$ satisfying the condition (a):*

$$(\forall x \in X)(\forall y \in Y)(\pi_1(pr(x, y)) = x \wedge \pi_2(pr(x, y)) = y).$$

Then $[O, pr, \pi_1, \pi_2]$ also satisfies (b) and so forms a pairing scheme.

Proof. We argue that the uniqueness of pr ensures that the function pr is surjective, and then that its surjectivity implies that condition (b) from Defn. 33 holds as well as the given condition (a).

Suppose pr is not surjective. Then for some $o \in O$, there is no $x \in X, y \in Y$ such that $pr(x, y) = o$. So $pr(\pi_1 o, \pi_2 o) = o' \neq o$. Consider then the function pr' which agrees with pr on all inputs except that $pr'(\pi_1 o, \pi_2 o) = o$. For all cases other than $x = \pi_1 o, y = \pi_2 o$ we still have $\pi_1(pr'(x, y)) = x \wedge \pi_2(pr'(x, y)) = y$, and by construction for the remaining case $\pi_1(pr'(\pi_1 o, \pi_2 o)) = \pi_1 o \wedge \pi_2(pr'(\pi_1 o, \pi_2 o)) = \pi_2 o$. So condition (a) holds for pr' , where $pr' \neq pr$. Contraposing, if pr uniquely satisfies the condition (a), it is surjective.

Because pr is surjective, every $o \in O$ is $pr(x, y)$ for some x, y . But then by (a) $\pi_1 o = x \wedge \pi_2 o = y$, and hence $pr(\pi_1 o, \pi_2 o) = pr(x, y) = o$. Which proves (b). \square

Pairing up X with Y through a pairing scheme, then, gives us a set-of-pair-objects O : so we can think of O as a serving as product of X with Y (relative to that scheme). But we don't want to identify the resulting product simply with the set O , for it depends crucially on the rest of the pairing scheme that O can play the right role. Our last theorem, however, makes the following an appropriate definition:

Definition 34. If X, Y are sets, then $[O, \pi_1, \pi_2]$ form a product of X with Y , where O is a set, and $\pi_1: O \rightarrow X, \pi_2: O \rightarrow Y$ are functions, so long as there is a unique two-place function $pr: X, Y \rightarrow O$ such that $(\forall x \in X)(\forall y \in Y)(\pi_1(pr(x, y)) = x \wedge \pi_2(pr(x, y)) = y)$. \triangle

7.3 Binary products, categorially

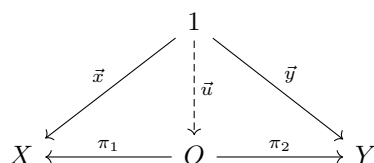
(a) We have characterized pairing schemes and the resulting products they create in terms of a set of objects O being the source and target of some appropriate morphisms satisfying the principles in Defns. 33 and 34. Which all looks highly categorial, very much in the spirit of the preamble to the previous chapter.

7.3 Binary products, categorially

But note that one crucial ingredient of our story so far, namely the pairing function $pr: X, Y \rightarrow O$, is a *binary* function (taking two objects as input, of course, not a single pair-object). And we can't just transport this over to a categorial setting. For an arrow in a category is always unary, with just one of the category's objects as its source. So how can we turn our very natural story about pairing schemes into a properly categorial account of products?

(b) Suppose for a moment that we are working in a well-pointed category like **Set**, where 'elements' in the sense of Defn. 30 do behave sufficiently like how elements intuitively should behave. In this case, instead of talking informally of two elements x of X and y of Y , we can talk of two arrows $\vec{x}: 1 \rightarrow X$ and $\vec{y}: 1 \rightarrow Y$.

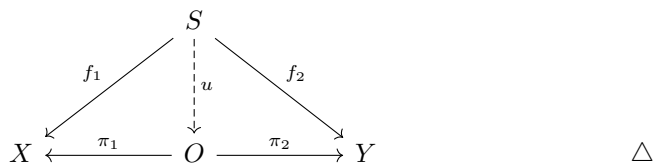
Now, suppose that there is an object O and two arrows, $\pi_1: O \rightarrow X$ and $\pi_2: O \rightarrow Y$ such that for every \vec{x} and \vec{y} there is a *unique* arrow $\vec{u}: 1 \rightarrow O$ such the following commutes:



Our arrow \vec{u} serves to pick out an element in O to serve as the product-object $pr(x, y)$. And the requirement that, uniquely, $\pi_1 \circ \vec{u} = \vec{x} \wedge \pi_2 \circ \vec{u} = \vec{y}$ is an instance of the condition in Defn. 34, now re-written in terms of elements-as-arrows. Which therefore gives us a categorial way of saying that $[O, \pi_1, \pi_2]$ form a product of X with Y .

So far, so good. But this will only give us what we want in well-pointed categories with 'enough' elements-as-arrows; for think what would happen if we were working e.g. in the category **Grp**. However, we know a potential way of generalizing claims to non-well-pointed categories: just replace talk about point elements with talk of generalized elements. Which motivates, at last, the following key definition:

Definition 35. In any category \mathcal{C} , a (*binary*) *product* $[O, \pi_1, \pi_2]$ for the objects X with Y is an object O together with 'projection' arrows $\pi_1: O \rightarrow X, \pi_2: O \rightarrow Y$, such that for any object S and arrows $f_1: S \rightarrow X$ and $f_2: S \rightarrow Y$ there is always a unique 'mediating' arrow $u: S \rightarrow O$ such that the following diagram commutes:



Products introduced

Note, by the way, that we are falling into the following common convention: in a category diagram, we use a dashed arrow \dashrightarrow to indicate an arrow which is uniquely fixed by the requirement that the diagram commutes.

(c) True, you can just stare at Defn. 35 if it is presented without ceremony, and ‘see’ that it is the sort of thing we need in a categorial context. But it has been worth taking the slow route, and so finding that it does arise entirely naturally from general considerations about what we want from a pairing scheme.

Let’s now have some examples of products in categories.

- (1) In **Set**, as you would certainly hope, the usual Cartesian product treated a the set $X \times Y$ of Kuratowski pairs $\langle x, y \rangle$ of elements from X and Y , together with the obvious projection functions $\langle x, y \rangle \xrightarrow{\pi_1} x$ and $\langle x, y \rangle \xrightarrow{\pi_2} y$, form a binary product.

Let’s just confirm this. Suppose we are given any set S and functions $f_1: S \rightarrow X$ and $f_2: S \rightarrow Y$. Then if, for $s \in S$, we put $u(s) = \langle f_1(s), f_2(s) \rangle$, the diagram evidently commutes. Now trivially, for any pair $p \in X \times Y$, $p = \langle \pi_1 p, \pi_2 p \rangle$. Hence if $u': S \rightarrow X \times Y$ is another candidate for completing the diagram, $u(s) = \langle f_1(s), f_2(s) \rangle = \langle \pi_1 u'(s), \pi_2 u'(s) \rangle = u'(s)$. So u is unique.

Motivated by this paradigm case, we will henceforth often use the notation $X \times Y$ for the object O in a binary product $[O, \pi_1, \pi_2]$ for X with Y .

Continuing our examples:

- (2) In group theory, we construct the direct product of two groups $\mathcal{G} = (G, \cdot, e_G)$ and $\mathcal{H} = (H, \odot, e_H)$ as follows. Take group elements to pairs in $G \times H$, the usual Cartesian product of the underlying sets; and then define the new group operation \times component-wise, i.e. put $\langle g, h \rangle \times \langle g', h' \rangle = \langle g \cdot g', h \odot h' \rangle$. It is immediate that the direct product of \mathcal{G} and \mathcal{H} , equipped with the obvious two projection functions which send $\langle g, h \rangle$ to g and to h respectively, is a categorial product of these groups in **Grp**.
- (3) Similarly a product of topological spaces defined in the usual way, equipped with the trivial projection functions recovering the original spaces, is a categorial product of topological spaces in **Top**.
- (4) Here’s a new example of a category, call it **Prop _{\mathcal{L}}** – its objects are propositions, wffs of a given first-order language \mathcal{L} , and there is a unique arrow from X to Y iff $X \models Y$, i.e. iff X semantically entails Y . The reflexivity and transitivity of semantic entailment means we get the identity and composition laws which ensure that this is a category.

In this case, the obvious categorial product of X with Y will be their logical product, i.e. the conjunction $X \wedge Y$, taken together with the obvious projections $X \wedge Y \rightarrow X$, $X \wedge Y \rightarrow Y$.

So far, then, so good: intuitive cases of products and categorial products are lining up nicely. One more example to be going on with:

- (5) Take a poset (P, \leq) considered as a category (so there is an arrow $p \rightarrow q$ iff $p \leq q$). Then a product of p and q would be an object c such that $c \leq p, c \leq q$ and such that for any object d with arrows from it to p and q , i.e. any d such that $d \leq p, d \leq q$, there is a unique arrow from d to c , i.e. $d \leq c$. That means the categorial product of p and q must be their ‘meet’ or greatest lower bound (equipped with the obvious two arrows).

A simple moral from the last example: since pairs of objects in posets need not in general have greatest lower bounds, this shows that a category in general need not have products (other than some trivial ones, as we shall see).

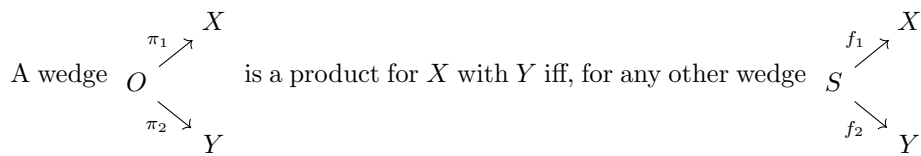
- (d) We noted at the beginning of this section that arrows in categories are unary. We don’t have true binary maps of the type $f: X, Y \rightarrow Z$ which we appealed to in the preceding section. We now know how to get round this issue, at least in a category with appropriate products. We can use instead arrows $f: X \times Y \rightarrow Z$.

But we won’t say more about this device now, but wait until we start putting it to real work later, beginning in Chapter 13.

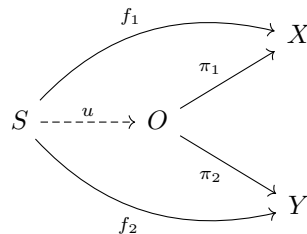
7.4 Products as terminal objects

Here’s a slightly different way of putting things. Let’s say

Definition 36. A wedge to X and Y (in category \mathcal{C}) is an object S and a pair of arrows $f_1: S \rightarrow X, f_2: S \rightarrow Y$. \triangle



to X and Y , there exists a unique morphism u such that the following diagram commutes:



Products introduced

We can say f_1 ‘factors’ as $\pi_1 \circ u$ and f_2 as $\pi_2 \circ u$, and hence the whole wedge from S into X and Y (*uniquely*) factors through the product via the mediating arrow u .

This definition of a product, now using the notion of wedges, can in turn be reframed as follows. First, we say:

Definition 37. Given a category \mathcal{C} and \mathcal{C} -objects X, Y , then the derived wedge category $\mathcal{C}_{W(XY)}$ has the following data. Its object-data are all the wedges $[O, f_1, f_2]$ to X, Y .¹ And an arrow from $[O, f_1, f_2]$ to $[O', f'_1, f'_2]$ is a \mathcal{C} -arrow $g: O \rightarrow O'$ such that the two resulting triangles commute: i.e. $f_1 = f'_1 \circ g$, $f_2 = f'_2 \circ g$. The identity arrow on $[O, f_1, f_2]$ is 1_O , and the composition of arrows in $\mathcal{C}_{W(XY)}$ is the same as their composition as arrows of \mathcal{C} . \triangle

It is easily confirmed that $\mathcal{C}_{W(XY)}$ is indeed a category.

With our new notion of the derived category $\mathcal{C}_{W(XY)}$ to hand, then the previous definition of a product is elementarily equivalent to

Definition 38. A product of X with Y in \mathcal{C} is a terminal object of the derived category $\mathcal{C}_{W(XY)}$. \triangle

7.5 Uniqueness up to unique isomorphism

As noted, products need not exist for arbitrary objects X and Y in a given category \mathcal{C} ; and when they exist, they need not be strictly unique. However, when they do exist, they *are* ‘unique up to unique isomorphism’ (compare Theorem 24). That is to say,

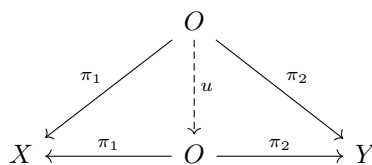
Theorem 26. *If both $[O, \pi_1, \pi_2]$ and $[O', \pi'_1, \pi'_2]$ are products for X with Y in the category \mathcal{C} , then there is a unique isomorphism $f: O \xrightarrow{\sim} O'$ commuting with the projection arrows (i.e. such that $\pi'_1 \circ f = \pi_1$ and $\pi'_2 \circ f = \pi_2$).*

Note the statement of the theorem carefully. It is *not* being baldly claimed that there is a unique isomorphism between any objects O and O' which are parts of products for some given X, Y . That’s false. For a very simple example, in **Set**, take the standard product object $X \times X$ comprising Kuratowski pairs: there are evidently two isomorphisms between it and itself, given by the maps $\langle x, x' \rangle \mapsto \langle x, x' \rangle$, and $\langle x, x' \rangle \mapsto \langle x', x \rangle$. The claim is, to repeat, that there is a unique isomorphism between any two product objects for X with Y *which commutes with their associated projection arrows*.

Plodding proof from basic principles. Since $[O, \pi_1, \pi_2]$ is a product, every wedge factors uniquely through it, including itself. In other words, there is a unique u such that this diagram commutes:

¹Does regarding $[O, f_1, f_2]$ as comprising an item of data in the category automatically mean treating it as single object in the logician’s sense, a real triple, meaning something over and above its components? Again, why so?

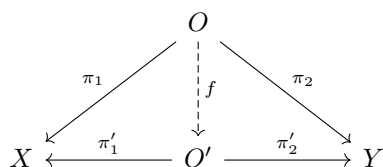
7.5 Uniqueness up to unique isomorphism



But evidently putting 1_O for the central arrow trivially makes the diagram commute. So by the uniqueness requirement we know that

- (i) Given an arrow $u: O \rightarrow O$, if $\pi_1 \circ u = \pi_1$ and $\pi_2 \circ u = \pi_2$, then $u = 1_O$.

Now, since $[O', \pi'_1, \pi'_2]$ is a product, $[O, \pi_1, \pi_2]$ has to uniquely factor through it:



In other words, there is a unique $f: O \rightarrow O'$ commuting with the projection arrows, i.e. such that

- (ii) $\pi'_1 \circ f = \pi_1$ and $\pi'_2 \circ f = \pi_2$.

And since $[O, \pi_1, \pi_2]$ is also a product, the other wedge has to uniquely factor through it. That is to say, there is a unique $g: O' \rightarrow O$ such that

- (iii) $\pi_1 \circ g = \pi'_1$ and $\pi_2 \circ g = \pi'_2$.

Whence,

- (iv) $\pi_1 \circ g \circ f = \pi'_1 \circ f = \pi_1$ and $\pi_2 \circ g \circ f = \pi_2$.

From which it follows – given our initial observation (i) – that

- (v) $g \circ f = 1_O$

The situation with the wedges is symmetric so we also have

- (vi) $f \circ g = 1_{O'}$

Hence f has a two-sided inverse, i.e. is an isomorphism. □

However, you'll recognize the key proof idea here is akin to the one we used in proving that initial/terminal objects are unique up to unique isomorphism. And we indeed can just appeal to that earlier result:

Proof using the alternative definition of products. $[O, \pi_1, \pi_2]$ and $[O', \pi'_1, \pi'_2]$ are both terminal objects in the wedge category $\mathcal{C}_{W(XY)}$. So by Theorem 18 there is a unique $\mathcal{C}_{W(XY)}$ -isomorphism f between them. But, by definition, this has to be a \mathcal{C} -arrow $f: O \rightarrow O'$ commuting with the projection arrows. And it is immediate that an isomorphism in $\mathcal{C}_{W(XY)}$ is also an isomorphism in \mathcal{C} . □

7.6 ‘Universal mapping properties’

Let’s pause for a moment. We have defined a binary product for X with Y categorially as a special sort of wedge to X and Y .

Now, that doesn’t fix a product absolutely; but we have now seen that products will be ‘unique up to unique isomorphism’. And what makes a wedge a product for X with Y is that it has a certain universal property – i.e. *any* other wedge to X and Y factors uniquely through a product wedge via a unique arrow.

Since arrows are typically functions or maps, we can therefore say that products are defined by a *universal mapping property*. We’ve already met other examples of universal mapping properties: terminal and initial objects are defined by how any other object has a unique map/arrow to or from them. We will meet lots more examples.

It is perhaps too soon, however, to attempt a formal definition of what it is to be defined by a universal mapping property. So for the moment take the notion as an informal gesture towards a common pattern of definition which we will recognize when we come across it.

7.7 Coproducts

(a) We are going now to discuss the duals of products. But first, we should note a common terminological device:

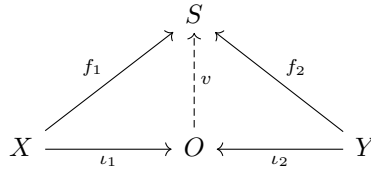
Definition 39. Duals of categorially defined widgets are very often called *co-widgets*. Thus a *co-widget* of the category \mathcal{C} is a widget of \mathcal{C}^{op} . \triangle

For example, we have met co-slice categories, the duals of slice categories. We could (and a few do) call initial objects ‘co-terminal’. Likewise we could (and a few do) call sections ‘co-retractions’. True, there is a limit to this sort of thing – no one, as far as I know, talks e.g. of ‘co-monomorphisms’ (instead of ‘epimorphisms’). But still, the general convention is used very widely.

In particular, it is absolutely standard to talk of the duals of products as ‘co-products’ – though in this case, as in some others, the hyphen is usually dropped.

(b) The definition of a coproduct is immediately obtained, then, by reversing all the arrows in our definition of products. Thus:

Definition 40. In any category \mathcal{C} , a (binary) *coproduct* $[O, \iota_1, \iota_2]$ for the objects X with Y is an object O together with two ‘injection’ arrows $\iota_1: X \rightarrow O, \iota_2: Y \rightarrow O$, such that for any object S and arrows $f_1: X \rightarrow S$ and $f_2: Y \rightarrow S$ there is always a unique ‘mediating’ arrow $v: O \rightarrow S$ such that the following diagram commutes:



The object O in a coproduct for X with Y can be notated ' $X \oplus Y$ ' or ' $X \amalg Y$ '. \triangle

Note, however, that the 'injections' in this sense need not be injective or even monic.

Let's say that objects and arrows arranged as $X \xrightarrow{\iota_1} O \xleftarrow{\iota_2} Y$ form a *corner* (or we could say 'co-wedge'!) from X and Y with vertex O . Then a coproduct of X with Y can be thought of as a corner from X and Y which factors through any other corner from X and Y via a unique map between the vertices of the corners.

We could now go on to define a category of corners from X and Y on the model of a category of wedges to X and Y , and then redefine a coproduct of X with Y as an initial object of this category. It is a useful reality check to work through the details.

(c) Let's have some examples of coproducts. Start with easy cases:

- (1) In **Set**, disjoint unions are instances of coproducts.

Given sets X and Y , let $X \oplus Y$ be the set with members $\langle x, 0 \rangle$ for $x \in X$ and $\langle y, 1 \rangle$ for $y \in Y$. And let the injection arrow $\iota_1: X \rightarrow X \oplus Y$ be the function $x \mapsto \langle x, 0 \rangle$, and similarly let $\iota_2: Y \rightarrow X \oplus Y$ be the function $y \mapsto \langle y, 1 \rangle$. Then $[X \oplus Y, \iota_1, \iota_2]$ is a coproduct for X with Y .

To show this, take any object S and arrows $f_1: X \rightarrow S$ and $f_2: Y \rightarrow S$, and then define the function $v: X \oplus Y \rightarrow S$ as sending an element $\langle x, 0 \rangle$ to $f_1(x)$ and an element $\langle y, 1 \rangle$ to $f_2(y)$.

By construction, this will make both triangles commute in the diagram in the definition above.

Moreover, if v' is another candidate for completing the diagram, then $v'(\langle x, 0 \rangle) = v' \circ \iota_1(x) = f_1(x) = v(\langle x, 0 \rangle)$, and likewise $v'(\langle y, 1 \rangle) = v(\langle y, 1 \rangle)$, whence $v' = v$, which gives us the necessary uniqueness.

- (2) In **Prop_L** (which we met in §7.3) the disjunction $X \vee Y$ (with the obvious injections $X \rightarrow X \vee Y$, $Y \rightarrow X \vee Y$) is a coproduct of X with Y .
- (3) Take a poset (P, \leq) considered as a category (so there is an arrow $p \rightarrow q$ iff $p \leq q$). Then a coproduct of p and q would be an object c such that $p \leq c, q \leq c$ and such that for any object d such that $p \leq d, q \leq d$ there is a unique arrow from c to d , i.e. $c \leq d$. Which means that the coproduct of p and q , if it exists, must be their least upper bound (equipped with the obvious two arrows).

Products introduced

(d) In some cases, however, the story about coproducts gets more complicated. We'll mention a couple of examples: but the details here aren't going to matter, so by all means skip:

- (4) In the category **Grp**, coproducts are (isomorphic to) the so-called 'free products' of groups.

Take the groups $G = (G, \cdot)$, $H = (H, \odot)$.² If necessary, now doctor the groups to equate their identity elements while ensuring the sets G and H are otherwise disjoint. Form all the finite 'reduced words' $G \star H$ you get by concatenating elements from $G \cup H$, and then multiplying out neighbouring G -elements by \cdot and neighbouring H -elements by \odot as far as you can. Equip $G \star H$ with the operation \diamond of concatenation-of-words-followed-by-reduction. Then $G \star H = (G \star H, \diamond)$ is a group – the free product of the two groups G and H – and there are obvious 'injection' group homomorphisms $\iota_1: G \rightarrow G \star H$, $\iota_2: H \rightarrow G \star H$.

Claim: $[G \star H, \iota_1, \iota_2]$ is a coproduct for the groups G and H . That is to say, for any group $K = (K, *)$ and morphisms $f_1: G \rightarrow K$, $f_2: H \rightarrow K$, there is a unique v such that this commutes:

$$\begin{array}{ccccc}
 & & K & & \\
 & \nearrow f_1 & \uparrow v & \nwarrow f_2 & \\
 G & \xrightarrow{\iota_1} & G \star H & \xleftarrow{\iota_2} & H
 \end{array}$$

Put $v: G \star H \rightarrow K$ to be the morphism that sends a word $g_1 h_1 g_2 h_2 \cdots g_r$ ($g_i \in G, h_i \in H$) to $f_1(g_1) * f_2(h_1) * f_1(g_2) * f_2(h_2) * \cdots * f_1(g_r)$. By construction, $v \circ \iota_1 = f_1$, $v \circ \iota_2 = f_2$. So that makes the diagram commute.

Let v' be any other candidate group homomorphism to make the diagram commute. Then, to take a simple example, consider $gh \in G \star H$. Then $v'(gh) = v'(g) * v'(h) = v'(\iota_1(g)) * v'(\iota_2(h)) = f_1(g) * f_2(h) = v(\iota_1(g)) * v(\iota_2(h)) = v(\iota_1(g) * \iota_2(h)) = v(gh)$. Similarly $v'(hg) = v(hg)$. So by induction over the length of words w we can go on to show quite generally $v'(w) = v(w)$. Hence, as required, v is unique.

- (5) So what about coproducts in **Ab**, the category of abelian groups? Since the free product of two abelian groups need not be abelian, the same construction won't work again as it stands.

OK: hit the construction with the extra requirement that words in $G \star H$ be treated as the same if one can be shuffled into the other (in effect, further reduce $G \star H$ by quotienting by the obvious equivalence relation). But that

²If you are feeling picky, you might prefer to continue writing e.g. $\mathcal{G} = (G, \cdot)$, thus more carefully signalling when you are talking about the group and when you are referring to its carrier set. Fine. Be my guest. But the conventional overloading of notation makes for less visual clutter and context always disambiguates.

means that we can take a word other than the identity, bring all the G -elements to the front, followed by all the H elements: now multiply out the G -elements and the H -elements and we are left with two-element word gh . So we can equivalently treat the members of our further reduced $G \star H$ as ordered pairs (g, h) belonging to $G \times H$. Equip these with the group operation \times defined component-wise as before (in §7.3): this gives us an abelian group $G \times H$ if G and H are abelian. Take the obvious injections, $g \xrightarrow{\iota_1} (g, 1)$ and $h \xrightarrow{\iota_2} (1, h)$. Then we claim $[G \times H, \iota_1, \iota_2]$ is a coproduct for the abelian groups G and H .

Take any abelian group $K = (K, *)$ and morphisms $f_1: G \rightarrow K$, $f_2: H \rightarrow K$. Put $v: G \times H \rightarrow K$ to be the morphism that sends (g, h) to $f_1(g) * f_2(h)$. This evidently makes the coproduct diagram (with $G \times H$ for $G \star H$) commute. And a similar argument to before shows that it is unique.

So, in the case of abelian groups, the *same* objects can serve as both products and coproducts, when equipped with appropriate projections and injections respectively.

8 Products explored

We continue to explore binary products, before going on to discuss products of more than two objects. Of course, everything in this chapter dualizes: but we can leave it as an exercise to supply all the further dual results about coproducts.

8.1 More properties of binary products

(a) We check that binary products, as defined, have various properties, including some obviously desirable ones:

Theorem 27. *In a category which has a terminal object 1,*

(1) *Products $1 \times X$ and $X \times 1$ exist, and $1 \times X \cong X \cong X \times 1$.*

In a category where the relevant products exist,

(2) $X \times Y \cong Y \times X$,

(3) $X \times (Y \times Z) \cong (X \times Y) \times Z$.

Proof for (1). We prove half the result. Note the wedge (V) $1 \xleftarrow{!_X} X \xrightarrow{1_X} X$ exists for some unique arrow $!_X$ since 1 is terminal. Take any other wedge to 1 and X , namely $1 \xleftarrow{!_Y} Y \xrightarrow{f} X$. Then the following diagram always trivially commutes:

$$\begin{array}{ccccc}
 & & Y & & \\
 & \swarrow & \downarrow f & \searrow f & \\
 1 & \xleftarrow{!_X} & X & \xrightarrow{1_X} & X
 \end{array}$$

(the triangle on the left commutes because there can only be one arrow from Y to 1 which forces $!_X \circ f = !_Y$). And obviously f is the only vertical, i.e. mediating, arrow which makes this commute. Hence $[X, !_X, 1_X]$ satisfies the conditions for being a product of 1 with X . So, by Theorem 26, given any product $[1 \times X, \pi_1, \pi_2]$, we have $1 \times X \cong X$. \square

Laborious proof for (2). Given products $[X \times Y, \pi_1, \pi_2]$ and $[Y \times X, \pi'_1, \pi'_2]$, then consider the following diagram:

$$\begin{array}{ccccc}
 X & \xleftarrow{\pi_1} & X \times Y & \xrightarrow{\pi_2} & Y \\
 \downarrow 1_X & & \downarrow o & & \downarrow 1_Y \\
 X & \xleftarrow{\pi'_2} & Y \times X & \xrightarrow{\pi'_1} & Y \\
 \downarrow 1_X & & \downarrow o' & & \downarrow 1_Y \\
 X & \xleftarrow{\pi_1} & X \times Y & \xrightarrow{\pi_2} & Y
 \end{array}$$

Here the wedge $Y \xleftarrow{1_Y \circ \pi_2} X \times Y \xrightarrow{1_X \circ \pi_1} X$ factors uniquely through the product $Y \times X$ via o . Similarly for o' . Hence, putting things together and absorbing the identities, the wedge $X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y$ factors uniquely through itself via $o' \circ o$. But of course that wedge factors through itself via $1_{X \times Y}$, so $o' \circ o = 1_{X \times Y}$. Similarly $o \circ o' = 1_{Y \times X}$. Therefore o and o' are inverse to each other, so isomorphisms, and hence $X \times Y \cong Y \times X$. \square

Snappy proof for (2). If $[X \times Y, \pi_1, \pi_2]$ is a product of X with Y , then $[X \times Y, \pi_2, \pi_1]$ will obviously serve as a product of Y with X . Hence, by Theorem 26 again, there is an isomorphism between the object in that product and the object $Y \times X$ of any other product of Y with X . \square

Proof for (3) postponed. It is a just-about-useful reality check to prove this by appeal to our initial definition of a product, using brute force. You are invited to try! But we give a slicker proof in §8.5. \square

(b) Do we similarly have $0 \times X \cong 0$ in categories with an initial object and the relevant product? Not always:

Theorem 28. *There are categories where the product $0 \times X$ or $X \times 0$ always exists but is not generally isomorphic to 0 .*

Proof. Take a category which has a null object, so here we can set $0 = 1$. Then since every product $1 \times X$ exists, so does $0 \times X$. Now suppose $0 \times X \cong 0$. Then we would have $X \cong 1 \times X = 0 \times X \cong 0$.

Take then a category like \mathbf{Grp} which has a null object (and so all products $0 \times X$ exist), but which also has other non-isomorphic objects, so we don't always have $X \cong 0$. It follows that in \mathbf{Grp} it can't always be the case that $0 \times X \cong 0$. \square

8.2 And two more results

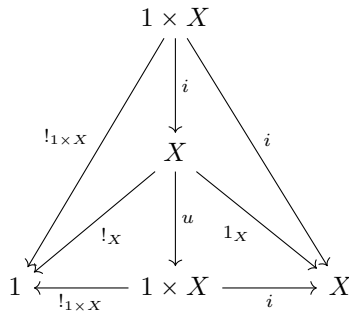
We pause to note two fiddly results, which you are very welcome to skip for now and return to when you need them. First:

Products explored

Theorem 29. *If $1 \xleftarrow{!_{1 \times X}} 1 \times X \xrightarrow{i} X$ is a product, then i is an isomorphism. Similarly for the mirror image result.*

We've shown that there is an isomorphism between $1 \times X$ and X ; but there could also be other arrows between them. So it takes another argument to show that in any product wedge (W) $1 \xleftarrow{!_{1 \times X}} 1 \times X \xrightarrow{i} X$, i has to be an isomorphism.

Proof. Consider, then, the following diagram:



This commutes. The wedge (V) $1 \xleftarrow{!_X} X \xrightarrow{!_X} X$ must factor through the product (W) via a unique mediating arrow u , and then $i \circ u = !_X$.

Similarly (W) factors through (V) as shown. But putting the triangles together means that (W) factors through (W) via the (unique) mediating arrow $u \circ i$. But since (W) also factors through itself via $!_{1 \times X}$, it follows that $u \circ i = !_{1 \times X}$.

Having inverses on both sides, i is therefore an isomorphism. □

And now second, again for future use, we should remark on a non-theorem. Suppose we have a pair of parallel composite arrows built up using the same projection arrow like this: $X \times Y \xrightarrow{\pi_1} X \xrightarrow[f]{g} X'$. In **Set**, the projection arrow here just ‘throws away’ the second component of pairs living in $X \times Y$, and all the real action happens on X , so if $f \circ \pi_1 = g \circ \pi_1$, we should also have $f = g$. Generalizing, we might then suppose that, in any category, projection arrows in products are always right-cancellable, i.e. are epic.

This is wrong. Here's a brute-force counterexample. Consider the mini category with just four objects together with the following diagrammed arrows (labelled suggestively but noncommittally), plus all identity arrows, and the necessary two composites:

$$X' \xleftarrow[f]{g} X \xleftarrow{\pi_1} V \xrightarrow{\pi_2} Y$$

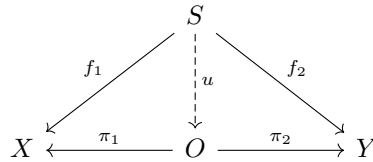
If that is all the data we have to go on, we can consistently stipulate that in this mini-category $f \neq g$ but $f \circ \pi_1 = g \circ \pi_1$.

Now, there is only one wedge of the form $X \longleftarrow ? \longrightarrow Y$, so trivially all wedges of that shape uniquely factor through it. In other words, the wedge $X \xleftarrow{\pi_1} V \xrightarrow{\pi_2} Y$ is trivially a product and π_1 is indeed a projection arrow. But by construction it isn't epic.

8.3 More on mediating arrows

We introduce some natural notation for mediating arrows in products, and then gather together a handful of further simple results.

Definition 41. Suppose $[O, \pi_1, \pi_2]$ is a binary product for the objects X with Y , and suppose the wedge $X \xleftarrow{f_1} S \xrightarrow{f_2} Y$ factors through it via the unique mediating arrow $u: S \rightarrow O$ so the following diagram commutes:



Then the unique mediating arrow u will be represented by $\langle f_1, f_2 \rangle$. △

We should check that our product-style notation $\langle f_1, f_2 \rangle$ for mediating arrows here doesn't mislead. But indeed we have:

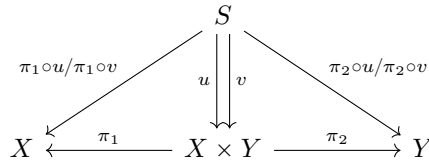
Theorem 30. *If $\langle f_1, f_2 \rangle = \langle g_1, g_2 \rangle$, then $f_1 = g_1$ and $f_2 = g_2$.*

Proof. Being equal, $\langle f_1, f_2 \rangle$ and $\langle g_1, g_2 \rangle$ must share as target the object in some product $[X \times Y, \pi_1, \pi_2]$. We therefore have $f_i = \pi_i \circ \langle f_1, f_2 \rangle = \pi_i \circ \langle g_1, g_2 \rangle = g_i$. □

We also have:

Theorem 31. *Given a product $[X \times Y, \pi_1, \pi_2]$ and arrows $S \xrightarrow[u]{v} X \times Y$, then, if $\pi_1 \circ u = \pi_1 \circ v$ and $\pi_2 \circ u = \pi_2 \circ v$, it follows that $u = v$.*

Proof. We have in fact already seen this result for the special case where v is the identity arrow. Another diagram shows all we need to prove the general case:



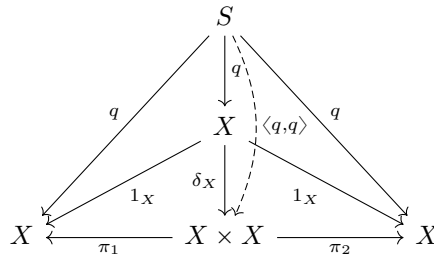
The same wedge $X \leftarrow S \rightarrow Y$ factors through $X \times Y$ both via u and v hence, by uniqueness of mediating arrows, $u = v$. □

Definition 42. Suppose we are working in a category with the relevant products. Then the wedge $X \xleftarrow{1_X} X \xrightarrow{1_X} X$ must factor uniquely through the product $X \times X$ via an arrow $\delta_X: X \rightarrow X \times X$. That unique arrow δ_X , i.e. $\langle 1_X, 1_X \rangle$, is the diagonal morphism on X . \triangle

In **Set**, thinking of $X \times X$ in the usual way, δ_X sends an element $x \in X$ to $\langle x, x \rangle$ (imagine elements $\langle x, x \rangle$ lying down the diagonal of a two-dimensional array of pairs $\langle x, y \rangle$: hence the label ‘diagonal’ and the notation δ).

Theorem 32. Given an arrow $q: S \rightarrow X$, $\delta_X \circ q = \langle q, q \rangle$.

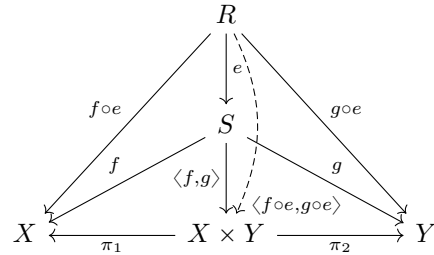
Proof. Consider the following diagram:



The inner triangles commute, hence $\delta_X \circ q$ is a mediating arrow factoring the wedge $X \xleftarrow{q} S \xrightarrow{q} X$ through the product $X \times X$. But by definition, the unique mediating arrow which does that is $\langle q, q \rangle$. \square

Theorem 33. Assuming $\langle f, g \rangle$ and e compose, $\langle f, g \rangle \circ e = \langle f \circ e, g \circ e \rangle$.

Proof. Another, rather similar, diagram gives the proof:



Again the inner triangles commute, hence $\langle f, g \rangle \circ e$ is a mediating arrow factoring the wedge with apex R through the product $X \times Y$. But by definition, the unique mediating arrow is $\langle f \circ e, g \circ e \rangle$. \square

Theorem 34. Given parallel arrows $S \begin{matrix} \xrightarrow{f_1} \\ \xrightarrow{f_2} \end{matrix} X$, with $f_1 \neq f_2$, there are (at least) four distinct arrows $S \rightarrow X \times X$.

8.4 Maps between two products

Proof. By definition of the product, for each pair of indices $i, j \in \{1, 2\}$ there is a unique map $\langle f_i, f_j \rangle$ which makes the product diagram commute,

$$\begin{array}{ccccc}
 & & S & & \\
 & \swarrow f_i & \vdots \langle f_i, f_j \rangle & \searrow f_j & \\
 X & \xleftarrow{\pi_1} & X \times X & \xrightarrow{\pi_2} & X
 \end{array}$$

It is immediate from Theorem 30 that if $\langle f_i, f_j \rangle = \langle f_k, f_l \rangle$, then $i = k, j = l$. So each of the four different pairs of indices tally different arrows $\langle f_i, f_j \rangle$. \square

8.4 Maps between two products

(a) Suppose we have two arrows $f: X \rightarrow X', g: Y \rightarrow Y'$. Then we might want to characterize an arrow between products, $f \times g: X \times Y \rightarrow X' \times Y'$, which works component-wise – i.e., putting it informally, the idea is that $f \times g$ sends the product of elements x and y to the product of $f(x)$ and $g(y)$.

In more categorical terms, we require $f \times g$ to be such that the following diagram commutes:

$$\begin{array}{ccccc}
 X & \xleftarrow{\pi_1} & X \times Y & \xrightarrow{\pi_2} & Y \\
 \downarrow f & & \downarrow f \times g & & \downarrow g \\
 X' & \xleftarrow{\pi'_1} & X' \times Y' & \xrightarrow{\pi'_2} & Y'
 \end{array}$$

Note, however, that the vertical arrow is then a mediating arrow from the wedge $X' \xleftarrow{f \circ \pi_1} X \times Y \xrightarrow{g \circ \pi_2} Y'$ through the product $X' \times Y'$. Therefore $f \times g$ is indeed fixed uniquely by the requirement that that diagram commutes, and must equal $\langle f \circ \pi_1, g \circ \pi_2 \rangle$. This warrants the following definition as in good order:

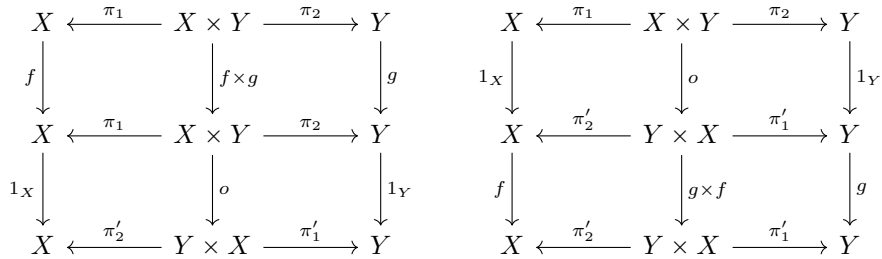
Definition 43. Given the arrows $f: X \rightarrow X', g: Y \rightarrow Y'$, and the products $[X \times Y, \pi_1, \pi_2]$ and $[X' \times Y', \pi'_1, \pi'_2]$, then $f \times g: X \times Y \rightarrow X' \times Y'$ is the unique arrow such that $\pi'_1 \circ f \times g = f \circ \pi_1$ and $\pi'_2 \circ f \times g = g \circ \pi_2$. \triangle

(b) By way of reality checks, let's prove a pair of theorems which should look obvious if you have been following the various definitions.

Theorem 35. Suppose we have arrows $f: X \rightarrow X$ and $g: Y \rightarrow Y$, and an order-swapping isomorphism $o: X \times Y \rightarrow Y \times X$. Then $o \circ (f \times g) = (g \times f) \circ o$.

Proof. Suppose we have products $[X \times Y, \pi_1, \pi_2]$ and $[Y \times X, \pi'_1, \pi'_2]$, and an isomorphism $o: X \times Y \rightarrow Y \times X$, as in the proof of Theorem 27 (2). And now consider the following pair of diagrams:

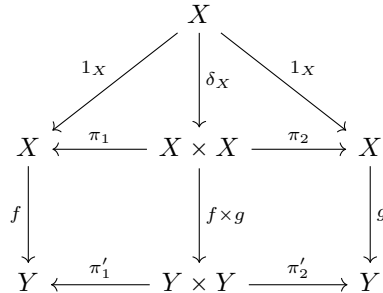
Products explored



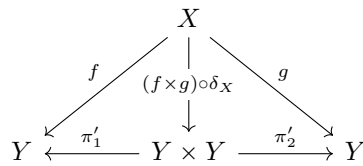
(Careful with the directions of the projection arrows!). Both diagrams commute, revealing that the same wedge factors through the bottom product via both $o \circ (f \times g)$ and $(g \times f) \circ o$. Those arrows must therefore be equal by the uniqueness of mediating arrows. \square

Theorem 36. *Suppose we have parallel arrows $f, g: X \rightarrow Y$ in a category with binary products. Then the arrow $\langle f, g \rangle$ is equal to the composite $(f \times g) \circ \delta_X$.*

Proof. The idea is that it should not matter whether we apply f and g separately to an element of X and take the product, or take the product of that element with itself and apply f and g componentwise. So take the diagram



This commutes by the definitions of δ_X and $f \times g$. Hence the following also commutes:



Which makes $(f \times g) \circ \delta_X$ the mediating arrow in a product diagram, so by uniqueness and the definition of $\langle f, g \rangle$, we have $(f \times g) \circ \delta_X = \langle f, g \rangle$. \square

(c) Here's a special case: sometimes we have an arrow $f: X \rightarrow X'$ and we want to define an arrow from $X \times Y$ to $X' \times Y$ which applies f to the first component of a product and leaves the second alone. Then $f \times 1_Y$ will do the trick.

8.5 Finite products more generally

It is tempting to suppose that if we have parallel maps $f, g: X \rightarrow X'$ and $f \times 1_Y = g \times 1_Y$, then $f = g$. But this actually fails in some categories – for example, in the toy category we met in §8.2, whose only arrows are as diagrammed

$$X' \begin{array}{c} \xleftarrow{f} \\ \xrightarrow{g} \end{array} X \xleftarrow{\pi_1} V \xrightarrow{\pi_2} Y$$

together with the necessary identities and composites, and where by stipulation $f \neq g$ but $f \circ \pi_1 = g \circ \pi_1$ (and hence $f \times 1_Y = g \times 1_Y$).

(d) Later, we will also need the following (rather predictable) general result:

Theorem 37. *Assume that there are arrows*

$$\begin{array}{ccccc} X & \xrightarrow{f} & X' & \xrightarrow{j} & X'' \\ Y & \xrightarrow{g} & Y' & \xrightarrow{k} & Y'' \end{array}$$

Assume there are products $[X \times Y, \pi_1, \pi_2]$, $[X' \times Y', \pi'_1, \pi'_2]$ and $[X'' \times Y'', \pi''_1, \pi''_2]$. Then $(j \times k) \circ (f \times g) = (j \circ f) \times (k \circ g)$.

Proof. By the defining property of arrow products applied to the three different products we get,

$$\pi''_1 \circ (j \times k) \circ (f \times g) = j \circ \pi'_1 \circ (f \times g) = j \circ f \circ \pi_1 = \pi''_1 \circ (j \circ f) \times (k \circ g).$$

Similarly

$$\pi''_2 \circ (j \times k) \circ (f \times g) = \pi''_2 \circ (j \circ f) \times (k \circ g)$$

The theorem then immediately follows by Theorem 31. □

8.5 Finite products more generally

(a) So far we have talked of binary products. But we can generalize in obvious ways. For example,

Definition 44. In any category \mathcal{C} , a *ternary product* $[O, \pi_1, \pi_2, \pi_3]$ for the objects X_1, X_2, X_3 is an object O together with projection arrows $\pi_i: O \rightarrow X_i$ (for $i = 1, 2, 3$) such that for any object S and arrows $f_i: S \rightarrow X_i$ there is always a unique arrow $u: S \rightarrow O$ such that $f_i = \pi_i \circ u$. △

And then, exactly as we would expect, using just the same proof ideas as in the binary case, we can prove

Theorem 38. *If both the ternary products $[O, \pi_1, \pi_2, \pi_3]$ and $[O', \pi'_1, \pi'_2, \pi'_3]$ exist for X_1, X_2, X_3 in the category \mathcal{C} , then there is a unique isomorphism $f: O \xrightarrow{\sim} O'$ commuting with the projection arrows.*

Products explored

We can safely leave filling in the details as an exercise.

We now note that if \mathcal{C} has binary products for all pairs of objects, then it automatically has ternary products too, for

Theorem 39. $(X_1 \times X_2) \times X_3$ together with the obvious projection arrows forms a ternary product of X_1, X_2, X_3 .

Proof. Assume $[X_1 \times X_2, \pi_1, \pi_2]$ is a product of X_1 with X_2 , and also that $[(X_1 \times X_2) \times X_3, \rho_1, \rho_2]$ is a product of $X_1 \times X_2$ with X_3 .

Take any object S and arrows $f_i: S \rightarrow X_i$. By our first assumption, (a) there is a unique $u: S \rightarrow X_1 \times X_2$ such that $f_1 = \pi_1 \circ u$, $f_2 = \pi_2 \circ u$. And by our second assumption, (b) there is then a unique $v: S \rightarrow (X_1 \times X_2) \times X_3$ such that $u = \rho_1 \circ v$, $f_3 = \rho_2 \circ v$.

Therefore $f_1 = \pi_1 \circ \rho_1 \circ v$, $f_2 = \pi_2 \circ \rho_1 \circ v$, $f_3 = \rho_2 \circ v$

Now consider $[(X_1 \times X_2) \times X_3, \pi_1 \circ \rho_1, \pi_2 \circ \rho_1, \rho_2]$. This, we claim, is indeed a ternary product of X_1, X_2, X_3 . We've just proved that the cone with vertex S and arrows $f_i: S \rightarrow X_i$ factors through the product via the arrow v . It remains to confirm v 's uniqueness in this new role.

Suppose we have $w: S \rightarrow (X_1 \times X_2) \times X_3$ where $f_1 = \pi_1 \circ \rho_1 \circ w$, $f_2 = \pi_2 \circ \rho_1 \circ w$, $f_3 = \rho_2 \circ w$. Then $\rho_1 \circ w: S \rightarrow X_1 \times X_2$ is such that $f_1 = \pi_1 \circ (\rho_1 \circ w)$, $f_2 = \pi_2 \circ (\rho_1 \circ w)$. Hence by (a), $u = \rho_1 \circ w$. But now invoking (b), that together with $f_3 = \rho_2 \circ w$ entails $w = v$. \square

Of course, an exactly similar argument will show that the product $X_1 \times (X_2 \times X_3)$ together with the obvious projection arrows will serve as another ternary product of X_1, X_2, X_3 . Hence we are now at last in a position to neatly prove

Theorem 27. (3) $X \times (Y \times Z) \cong (X \times Y) \times Z$.

Proof. Both $(X_1 \times X_2) \times X_3$ and $X_1 \times (X_2 \times X_3)$ (with their projection arrows) are ternary products of X_1, X_2, X_3 . So Theorem 38 entails that $X_1 \times (X_2 \times X_3) \cong (X_1 \times X_2) \times X_3$. \square

(b) What goes for ternary products goes for n -ary products defined in a way exactly analogous to Defn. 44. If \mathcal{C} has binary products for all pairs of objects it will have quaternary products such as $((X_1 \times X_2) \times X_3) \times X_4$, quinary products, and n -ary products more generally, for any finite $n \geq 2$.

To round things out, how do things go for the nullary and unary cases?

Following the same pattern of definition, a *nullary* product in \mathcal{C} would be an object O together with *no* projection arrows, such that for any object S there is a unique arrow $u: S \rightarrow O$. Which is just to say that a nullary product is a terminal object of the category.

And a unary product of X would be an object O and a single projection arrow $\pi_1: O \rightarrow X$ such that for any object S and arrow $f: S \rightarrow X$ there is a unique arrow $u: S \rightarrow O$ such that $\pi_1 \circ u = f$. Putting $O = X$ and $\pi_1 = 1_X$ evidently fits the bill. So the basic case of a unary product of X is not quite X itself, but

rather X equipped with its identity arrow (and like any product, this is unique up to unique isomorphism). Trivially, unary products for all objects exist in all categories.

In sum, suppose we say

Definition 45. A category \mathcal{C} has all binary products iff for all \mathcal{C} -objects X and Y , there exists a binary product of X with Y in \mathcal{C} .

\mathcal{C} has all finite products iff \mathcal{C} has n -ary products for any n objects, for all $n \geq 0$. \triangle

Then our preceding remarks establish

Theorem 40. A category \mathcal{C} has all finite products iff \mathcal{C} has a terminal object and has all binary products.

8.6 Infinite products

We can now generalize still further in an obvious way, going beyond finite products to infinite cases.

Definition 46. Suppose that we are dealing with \mathcal{C} -objects X_j indexed by items j in some suite of indices J (not now assumed finite). Then the product of the X_j , if it exists in \mathcal{C} , is an object O together with a projection arrow $\pi_j: O \rightarrow X_j$ for each index j . It is required that for any object S and family of arrows $f_j: S \rightarrow X_j$ (one for each index), there is always a unique arrow $u: S \rightarrow O$ such that $f_j = \pi_j \circ u$. \triangle

For the same reasons as before, such a generalized product will be unique up to unique isomorphism.

Now, we are in fact only going to be really interested in cases where the suite of indices J can be treated as a set in standard set theory. In other words, we are really only going to be interested in cases where we take products of set-many objects. Ignoring the over-sized cases, we then say:

Definition 47. A category \mathcal{C} has all small products iff for any \mathcal{C} -objects X_j , for $j \in J$ where J is some index set, these objects have a product. We notate the object in the product of such X_j for $j \in J$ by $\prod_{j \in J} X_j$. \triangle

Here, ‘small’ is a joke. It doesn’t mean small by any normal standards – it just indicates that we are taking products over collections of objects that are not too many to form a set. We’ll be returning to such issues of size in Chapter 16.

9 Equalizers

Terminal and initial objects, products and coproducts, are defined by universal mapping properties. In this chapter, we look at another dual pair of constructs defined by such mapping properties, so-called equalizers and co-equalizers.

9.1 Equalizers

It was useful, when defining products, to introduce the idea of a ‘wedge’ (Defn. 36) for a certain configuration of objects and arrows in a category. Here’s a similar definition that is going to be useful in defining the equalizers:

Definition 48. A *fork* (from S through X to Y) consists of arrows $k: S \rightarrow X$ with $f: X \rightarrow Y$ and $g: X \rightarrow Y$, such that $f \circ k = g \circ k$. \triangle

So a fork is a commuting diagram $S \xrightarrow{k} X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$, with the composite arrows from S to Y being equal.

Now, as we saw, a product wedge from O to X and Y is a limiting case: it’s a wedge such that any other wedge from S to X and Y uniquely factors through it. Likewise, an equalizing fork from E through parallel arrows $f, g: X \rightarrow Y$ is another limiting case: it’s a fork such that any other fork from an object S through f, g uniquely factors through it. In other, clearer, words:

Definition 49. Let \mathcal{C} be a category and $f, g: X \rightarrow Y$ be a pair of parallel arrows in \mathcal{C} . Then the object E and arrow $e: E \rightarrow X$ form an *equalizer* in \mathcal{C} for those arrows iff $f \circ e = g \circ e$ (so $E \xrightarrow{e} X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$ is indeed a fork), and for any

fork $S \xrightarrow{k} X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$ there is a unique mediating arrow $u: S \rightarrow E$ such that the following diagram commutes:

$$\begin{array}{ccccc}
 S & & & & \\
 \downarrow u & \searrow k & & & \\
 E & \xrightarrow{e} & X & \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} & Y
 \end{array}$$

\triangle

We now note that, just as with products (see Defn. 37), we can give an alternative definition which defines equalizers in terms of a terminal object in a suitable category. First we say

Definition 50. Given a category \mathcal{C} and parallel arrows $f, g: X \rightarrow Y$, then the derived category of forks $\mathcal{C}_{F(fg)}$ has as objects all forks $S \xrightarrow{k} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$.

And an arrow from $S \xrightarrow{k} \dots$ to $S' \xrightarrow{k'} \dots$ in $\mathcal{C}_{F(fg)}$ is a \mathcal{C} -arrow $g: S \rightarrow S'$ such that the resulting triangle commutes: i.e. such that $k = k' \circ g$.

The identity arrow in $\mathcal{C}_{F(fg)}$ on the fork $S \xrightarrow{k} \dots$ is the identity arrow 1_S in \mathcal{C} ; and the composition of arrows in $\mathcal{C}_{F(fg)}$ is defined as the composition of the arrows as they feature in \mathcal{C} . \triangle

It is again easily checked that this indeed defines a category. Our definition of an equalizer then comes to this:

Definition 51. An equalizer of $f, g: X \rightarrow Y$ in \mathcal{C} is some $[E, e]$ (E a \mathcal{C} -object, e a \mathcal{C} -arrow $E \rightarrow X$) such that the resulting fork $E \xrightarrow{e} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$ is terminal in $\mathcal{C}_{F(fg)}$. \triangle

Here, then, are a few examples of equalizers:

- (1) Suppose in **Set** we have parallel arrows $X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$. Then let $E \subseteq X$ be the set such that $x \in E$ iff $fx = gx$, and let $e: E \rightarrow X$ be the simple inclusion map. By construction, $f \circ e = g \circ e$. So $E \xrightarrow{e} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$ is a fork. We show that $[E, e]$ is in fact an equalizer for f and g .

Suppose $S \xrightarrow{k} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$ is any other fork through f, g , which requires $f(k(s)) = g(k(s))$ for each $s \in S$ and hence $k[S] \subseteq E \subseteq X$. Defining the mediating arrow $u: S \rightarrow E$ to agree with $k: S \rightarrow X$ on all inputs will make the diagram for equalizers commute. And this is the unique possibility: for the diagram to commute we need $k = e \circ u$, and the inclusion e doesn't affect the values of the function (only its codomain), k and u must indeed agree on all inputs.

- (2) Equalizers in categories whose objects are sets-with-structure behave similarly. Take as an example the category **Mon**. Given a pair of monoid homomorphisms $(X, \cdot, 1_X) \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} (Y, *, 1_Y)$, take the subset E of X on which the functions agree. Evidently E must contain the identity element of X (since f and g agree on this element: being homomorphisms, both must send 1_X to the 1_Y). And suppose $e, e' \in E$: then $f(e \cdot e') = f(e) * f(e') = g(e) * g(e') = g(e \cdot e')$, which means that E is closed under products of members.

Equalizers

So take E together with the monoid operation from $(X, \cdot, 1_X)$ restricted to members of E . Then $(E, \cdot, 1_X)$ is a monoid – for the shared identity element still behaves as an identity, E is closed under the operation, and the operation is still associative. And if we take $(E, \cdot, 1_X)$ and equip it with the injection homomorphism into $(X, \cdot, 1_X)$, this will evidently give us an equalizer for f and g .

- (3) Similarly, take **Top**. What is the equalizer for a pair of continuous maps

$X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$? Well, take the subset of (the underlying set of) X on which the functions agree, and give it the subspace topology. This topological space equipped with the injection into X is then the desired equalizer. (This works because of the way that the subspace topology is defined – we won't go into details).

- (4) A special case. Suppose we are in **Grp** and have a group homomorphism, $f: X \rightarrow Y$. There is also another trivial homomorphism $o: X \rightarrow Y$ which sends any element of the group X to the identity element in Y , i.e. is the composite $X \rightarrow 1 \rightarrow Y$ of the only possible homomorphisms. Now consider what would constitute an equalizer for f and o .

Suppose K is the kernel of f , i.e. the subgroup of X whose objects are the elements which f sends to the identity element of Y , and let $i: K \rightarrow X$ be the inclusion map. Then $K \xrightarrow{i} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{o} \end{array} Y$ is a fork since $f \circ i = o \circ i$.

Let $S \xrightarrow{k} X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{o} \end{array} Y$ be another fork. Now, $o \circ k$ sends every element of S to the unit of Y . Since $f \circ k = o \circ k$, k must send any element of S to some element in the kernel K . So let $k': S \rightarrow K$ agree with $k: S \rightarrow X$ on all arguments.

Then the following commutes:

$$\begin{array}{ccccc}
 S & & & & \\
 \downarrow k' & \searrow k & & & \\
 & & X & \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{o} \end{array} & Y \\
 & \nearrow i & & & \\
 K & & & &
 \end{array}$$

And evidently k' is the only possible homomorphism to make the diagram commute.

So the equalizer of f and o is f 's kernel K equipped with the inclusion map into the domain of X . Or putting it the other way about, we can define kernels of group homomorphisms categorially in terms of equalizers.

- (5) Finally we remark that the equalizer of a pair of maps $X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$ where in fact $f = g$ is simply $[X, 1_X]$.

Consider then a poset (P, \leq) considered as a category whose objects are the members of P and where there is a unique arrow $X \rightarrow Y$ (for $X, Y \in P$)

iff $X \leq Y$. So the only cases of parallel arrows from X to Y are cases of equal arrows which then, as remarked, have equalizers. So in sum, a poset category has all possible equalizers.

9.2 Uniqueness again

Just as products are unique up to unique isomorphism, equalizers are too. That is to say,

Theorem 41. *If both the equalizers $[E, e]$ and $[E', e']$ exist for $X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$, then there is a unique isomorphism $j: E \xrightarrow{\sim} E'$ commuting with the equalizing arrows, i.e. such that $e = e' \circ j$.*

Plodding proof from first principles. We can use an argument that goes along exactly the same lines as the one we used to prove the uniqueness of products and equalizers. This is of course no accident, given the similarity of the definitions via a unique mapping property.

Assume $[E, e]$ equalizes f and g , and suppose $e \circ h = e$. Then observe that the following diagram will commute

$$\begin{array}{ccc}
 E & \xrightarrow{e} & X \\
 h \downarrow & & \xrightarrow[f]{g} Y \\
 E & \xrightarrow{e} & X
 \end{array}$$

Now obviously, $h = 1_E$ makes that diagram commute. But by hypothesis there is a unique arrow $E \rightarrow E$ which makes the diagram commute. So we can conclude that if $e \circ h = e$, then $h = 1_E$.

Now suppose $[E', e']$ is also an equalizer for f and g . Then $[E, e]$ must factor uniquely through it. That is to say, there is a (unique) mediating $j: E \rightarrow E'$ such that $e' \circ j = e$. And since $[E', e']$ must factor uniquely through $[E, e]$ there is a unique k such that $e \circ k = e'$. So $e \circ k \circ j = e$, and hence by our initial conclusion, $k \circ j = 1_E$.

A similar proof shows that $j \circ k = 1_{E'}$. Which makes the unique j an isomorphism. □

Proof using the alternative definition of equalizers. $[E, e]$ and $[E', e']$ are both terminal objects in the fork category $\mathcal{C}_{F(fg)}$. So by Theorem 18 there is a unique $\mathcal{C}_{F(fg)}$ -isomorphism j between them. But, by definition, this has to be a \mathcal{C} -arrow $j: E \xrightarrow{\sim} E'$ commuting with the equalizing arrows. And j is easily seen to be an isomorphism in \mathcal{C} too. □

Let's add two further general results about equalizers. First:

Theorem 42. *If $[E, e]$ constitute an equalizer, then e is a monomorphism.*

Equalizers

Proof. Assume $[E, e]$ equalizes $X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$, and suppose $e \circ j = e \circ k$, where for some D , $D \begin{smallmatrix} \xrightarrow{j} \\ \xrightarrow{k} \end{smallmatrix} E$. Then the following diagram commutes,

$$\begin{array}{ccccc}
 D & & & & \\
 \downarrow j & \searrow e \circ j = e \circ k & & \xrightarrow{f} & \\
 \downarrow k & & X & \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} & Y \\
 E & \nearrow e & & & \\
 \end{array}$$

So $D \begin{smallmatrix} \xrightarrow{e \circ j / e \circ k} \\ \xrightarrow{f} \end{smallmatrix} X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$ is a fork factoring through the equalizer. But by the definition of an equalizer, it has to factor uniquely, and hence $j = k$. In sum, e is left-cancellable in the equation $e \circ j = e \circ k$; i.e. e is monic. \square

Second, in an obvious shorthand,

Theorem 43. *In any category, an epic equalizer is an isomorphism*

Proof. Assume again that $[E, e]$ equalizes $X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y$, so that $f \circ e = g \circ e$. So if e is epic, it follows that $f = g$. Then consider the following diagram

$$\begin{array}{ccccc}
 X & & & & \\
 \downarrow u & \searrow 1_X & & \xrightarrow{f} & \\
 \downarrow e & & X & \xrightarrow{g} & Y \\
 E & \nearrow e & & & \\
 \end{array}$$

Because e equalizes, we know there is a unique u such that (i) $e \circ u = 1_X$.

But then also $e \circ u \circ e = 1_X \circ e = e = e \circ 1_E$. Hence, since equalizers are mono by the last theorem, (ii) $u \circ e = 1_E$.

Taken together, (i) and (ii) tell us that e has an inverse. Therefore e is an isomorphism. \square

9.3 Co-equalizers

(a) Dualizing, we get the notion of a co-equalizer. First we say:

Definition 52. A *co-fork* (from X through Y to S) consists of parallel arrows $f: X \rightarrow Y$, $g: X \rightarrow Y$ and an arrow $k: Y \rightarrow S$, such that $k \circ f = k \circ g$. \triangle

(Plain ‘fork’ is often used for the dual too: but the ugly ‘co-fork’ keeps things clear.) Diagrammatically, a co-fork looks like this: $X \begin{smallmatrix} \xrightarrow{f} \\ \xrightarrow{g} \end{smallmatrix} Y \xrightarrow{k} S$, with the composite arrows from X to S being equal. Then, as you would expect:

Definition 53. Let \mathcal{C} be a category and $f: X \rightarrow Y$ and $g: X \rightarrow Y$ be a pair of parallel arrows in \mathcal{C} . The object C and arrow $c: Y \rightarrow C$ form a *co-equalizer* in \mathcal{C} for those arrows iff $c \circ f = c \circ g$, and for any co-fork from X through Y to S there is a unique arrow $u: C \rightarrow S$ such the following diagram commutes:

$$\begin{array}{ccccc}
 & & & & S \\
 & & & & \uparrow \\
 X & \xrightarrow{f} & Y & \xrightarrow{k} & \\
 & \xrightarrow{g} & & \searrow c & \\
 & & & & C
 \end{array}
 \quad \triangle$$

(b) We need not pause to spell out the dual arguments that co-equalizers are unique up to a unique isomorphism or that co-equalizers are epic. Instead, we turn immediately to consider one central example by asking: what do co-equalizers look like in **Set**?

Suppose we are given parallel arrows $f, g: X \rightarrow Y$ in **Set**. These arrows induce a relation $R_{f,g}$ (or R for short) on the members of Y , where yRy' holds when there is an $x \in X$ such that $f(x) = y \wedge g(x) = y'$. Now, given a co-fork

$$X \xrightarrow[f]{g} Y \xrightarrow{k} S,$$

then yRy' implies $k(y) = k(y')$. And trivially, having equal k -values is an equivalence relation \equiv_k on members of Y .

So, in sum, we've shown that given a co-fork via $k: Y \rightarrow S$ from the parallel arrows $f, g: X \rightarrow Y$, there is a corresponding equivalence relation \equiv_k on Y such that if $yR_{f,g}y'$ then $y \equiv_k y'$.

Now what's the limiting case of such an equivalence relation? It will have to be R^\sim , the smallest equivalence relation containing $R_{f,g}$. So we'll expect that the limiting case of a cofork will comprise an arrow $c: Y \rightarrow C$ such that $\equiv_c = R^\sim$. In other words, we want c to be such that $c(y) = c(y')$ iff $yR^\sim y'$.

Which motivates the following:

Theorem 44. *Given functions $f, g: X \rightarrow Y$ in **Set**, let R^\sim be the smallest equivalence relation containing R - where yRy' iff $(\exists x \in X)(f(x) = y \wedge g(x) = y')$.*

Let C be Y/R^\sim , i.e. the set of R^\sim -equivalence classes of Y ; and let c map $y \in Y$ to the R^\sim -equivalence class containing y . Then $[C, c]$, so defined, is a co-equalizer for f and g .

Proof. We just have to do some routine checking. First we show $c \circ f = c \circ g$. But the left-hand side sends $x \in X$ to the R^\sim -equivalence class containing $f(x)$ and the right-hand side sends x to the R^\sim -equivalence class containing $g(x)$. However, $f(x)$ and $g(x)$ are by definition R -related, and hence are R^\sim -related: so by construction they belong to the same R^\sim -equivalence class. Hence

$$X \xrightarrow[f]{g} Y \xrightarrow{c} C$$

is indeed a co-fork.

Now suppose there is another co-fork $X \xrightarrow[f]{g} Y \xrightarrow{k} S$. We need to show the co-fork ending with c factors through this via a unique mediating arrow u .

Equalizers

By assumption, $k \circ f = k \circ g$. And we first outline a proof that if $yR^{\sim}y'$ then $k(y) = k(y')$.

Start with R defined as before, and let R' be its reflexive closure. Obviously we'll still have that if $yR'y'$ then $k(y) = k(y')$. Now consider R'' the symmetric closure of R' : again, we'll still have that if $yR''y'$ then $k(y) = k(y')$. Now note that if $yR''y'$ and $y'R''y''$, then $k(y) = k(y'')$. So if we take the transitive closure of R'' , we'll still have a relation which, when it holds between some y and y'' , implies that $k(y) = k(y'')$. But the transitive closure of R'' is R^{\sim} .

We have shown, then, that k is constant on members of a R^{\sim} -equivalence class, and so we can well-define a function $u: C \rightarrow S$ which sends an equivalence class to the value of k on a member of that class. This u is the desired mediating arrow which makes the diagram defining a co-equalizer commute. Moreover, since c is surjective and C only contains R^{\sim} -equivalence classes, u is the only function for which $u \circ c = k$. \square

In a slogan then: *in Set, quotienting by an equivalence relation is (up to unique isomorphism) the same as taking an associated co-equalizer.* In many other categories co-equalizers behave similarly, corresponding to 'naturally occurring' quotienting constructions. But we won't go into more detail here.

10 Limits and colimits defined

A terminal object is defined essentially in terms of how all the other objects in the category relate to it (by each sending it a unique arrow). A product wedge is defined in terms of how all the other wedges in a certain family relate to it (each factoring through it via a unique arrow). An equalizing fork is defined in terms of how all the other forks in a certain family relate to it (each factoring through it via a unique arrow). In an informal sense, terminal objects, products, and equalizers are limiting cases, defined in closely analogous ways using universal mapping properties. Likewise for their duals.

In this chapter, we now formally capture what's common to terminal objects, products and equalizers by defining a general class of *limits*, and confirming that terminal objects, products and equalizers are indeed examples. We also define a dual class of *co-limits*, which has initial objects, coproducts and co-equalizers as examples.

We then give a new pair of examples, one for each general class, the so-called pullbacks and pushouts.

10.1 Cones over diagrams

(a) We start by defining the notion of a cone over a diagram; then in the next section we can use this to define the key notion of a limit cone.

Way back in Defn. 8, we loosely characterized a diagram D in a category \mathcal{C} as being what is represented by a representational diagram – i.e. as simply consisting in a bunch of objects with, possibly, some arrows between some of them. We now need some more systematic scheme for labelling the objects in a diagram. So henceforth we'll assume that the objects in D can be labelled by terms like ' D_j ' where ' j ' is an index from some suite of indices J . For convenience, we'll allow double counting, permitting the case where $D_j = D_k$ for different indices. We allow the limiting cases of diagrams where there are no arrows, and even the empty case where there are no objects either. So:

Definition 8* A (labelled) *diagram in a category* \mathcal{C} is some (or no) objects D_j for indices j in the suite of indices J , and some (or no) \mathcal{C} -arrows between these objects. \triangle

Limits and colimits defined

(We eventually, in §17.1, give a tauter definition of diagrams, but this will do to be getting on with.)

Definition 54. Let D be a diagram in category \mathcal{C} . Then a *cone over D* comprises a \mathcal{C} -object C , the *vertex* or *apex* the cone, together with \mathcal{C} -arrows $c_j: C \rightarrow D_j$ (often called the *legs* of the cone), one for each object D_j in D , such that whenever there is an arrow $d: D_k \rightarrow D_l$ in D , $c_l = d \circ c_k$, i.e. the following diagram commutes:

$$\begin{array}{ccc} & C & \\ c_k \swarrow & & \searrow c_l \\ D_k & \xrightarrow{d} & D_l \end{array}$$

We use ‘ $[C, c_j]$ ’ as our notation for such a cone. △

Think of it diagrammatically(!) like this: arrange the objects in the diagram D in a plane, along with whatever arrows there are between them in D . Now sit the object C above the plane, with a quiverful of arrows from C zinging down, one to each object D_j in the plane. Those arrows form the ‘legs’ of a skeletal cone. And the key requirement is that any triangles thus formed with C at the apex must commute.

We should note, by way of aside, that some authors prefer to say more austere-ly that a cone is not a vertex-object-with-a-family-of-arrows-from-that-vertex but simply a family of arrows from the vertex. Since we can read off the vertex of a cone as the common source of all its arrows, it is very largely a matter of convenience whether we speak austere-ly or explicitly mention the vertex. But for the moment, we’ll take the less austere line.

(b) For later use, but also to help check understanding now, here is another definition and then two theorems:

Definition 55. The (reflexive, transitive) *closure* of a diagram D in a category \mathcal{C} is the smallest diagram which includes all the objects and arrows of D , but which also has an identity arrow on each object, and for any two of its composable arrows, it also contains their composition. △

In other words, the closure of a diagram D in \mathcal{C} is what you get by adding identity arrows where necessary, forming composites of any composable arrows you now have, then forming composites of what you have at the next stage, and so on and so forth. Since the associativity of the composition operation will be inherited from \mathcal{C} , it is immediate that

Theorem 45. *The closure of a diagram D in \mathcal{C} is a subcategory of \mathcal{C} .*

A little more interestingly, though almost equally easily, we have:

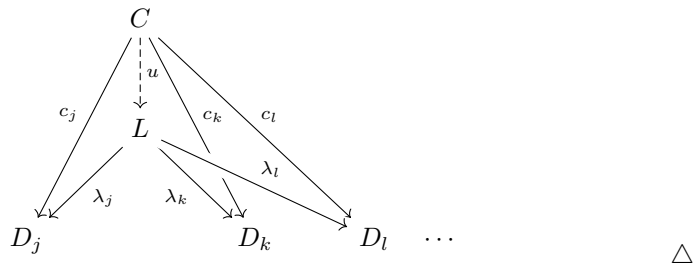
Theorem 46. *If $[C, c_j]$ is a cone over D , then it is a cone over the closure of D .*

Proof. The closure of D has no additional objects, so $[C, c_j]$ still has a leg from the vertex C to each object in the closure. It is trivial that, given an identity arrow $1_k: D_k \rightarrow D_k$, we have $c_k = 1_k \circ c_k$. So we just need to show a cone over composable arrows is still a cone when their composite is added. So suppose we have a cone over a diagram including the arrows $d: D_k \rightarrow D_l$ and $d': D_l \rightarrow D_m$. That means $c_l = d \circ c_k$ and $c_m = d' \circ c_l$. Hence $c_m = (d' \circ d) \circ c_k$. So the cone is still a cone if we add the composite arrow $d' \circ d: D_k \rightarrow D_m$. \square

10.2 Defining limit cones

(a) There can be many cones, with different vertices, over a given diagram D . But, in just the same spirit as our earlier definitions of products and equalizers, we can define a limiting case, by means of a universal mapping property:

Definition 56. A cone $[L, \lambda_j]$ over a diagram D in \mathcal{C} is a *limit (cone) over D* iff any cone $[C, c_j]$ over D uniquely factors through it, so there is a unique mediating arrow $u: C \rightarrow L$ such that for each index j , $\lambda_j \circ u = c_j$. In other words, for each D_j in D , the corresponding triangle whose other vertices are C and L commutes:



(b) Let's immediately confirm that our three announced examples of limits so far are indeed limit cones in the sense just defined.

- (1) We start with the null case. Take the empty diagram in \mathcal{C} – zero objects and so, necessarily, no arrows. Then a cone over the empty diagram is simply an object C , a lonely vertex (there is no further condition to fulfil), and an arrow between such minimal cones C, C' is just an arrow $C \rightarrow C'$. Hence L is a limit cone just if there is a unique arrow to it from any other object – i.e. just if L is a terminal object in \mathcal{C} !
- (2) Consider now a diagram which is just *two* objects we'll call ' D_1 ', ' D_2 ', still with no arrow between them. Then a cone over such a diagram is just a wedge into D_1, D_2 ; and a limit cone is simply a product of D_1 with D_2 .

Limits and colimits defined

(We could equally have considered the reflexive transitive closure of this two object diagram, i.e. the discrete category with two objects plus their identity arrows: by our last theorem, it would make no difference.)

- (3) Next consider a diagram which again has just two objects, but now with two parallel arrows between them, which we can represent $D_1 \begin{array}{c} \xrightarrow{d} \\ \xrightarrow{d'} \end{array} D_2$.

Then a cone over this diagram, or over its closure, is a commuting diagram like this:

$$\begin{array}{ccc} & C & \\ c_1 \swarrow & & \searrow c_2 \\ D_1 & \begin{array}{c} \xrightarrow{d} \\ \xrightarrow{d'} \end{array} & D_2 \end{array}$$

If there is such a diagram, then we must have $d \circ c_1 = d' \circ c_1$: and vice versa, if that identity holds, then we can put $c_2 = d \circ c_1 = d' \circ c_1$ to complete the commutative diagram. Hence we have a cone from the vertex C to our diagram iff $C \xrightarrow{c_1} D_1 \begin{array}{c} \xrightarrow{d} \\ \xrightarrow{d'} \end{array} D_2$ is a fork. Since c_1 fixes what c_2 has to be to complete the cone, we can focus on the cut-down cone consisting of just $[C, c_1]$.

What is the corresponding cut-down limit cone? It consists in $[E, e]$ such there is a unique u such that $c_1 = e \circ u$. Hence $[E, e]$ is an equalizer of the parallel arrows $D_1 \begin{array}{c} \xrightarrow{d} \\ \xrightarrow{d'} \end{array} D_2$.

- (c) We can now give a direct proof, along now hopefully entirely familiar lines, for the predictable result

Theorem 47. *Limit cones over a given diagram D are unique up to a unique isomorphism commuting with the cones's arrows.*

Proof. As usual, we first note that a limit cone $[L, \lambda_j]$ factors through itself via the mediating identity $1_L: L \rightarrow L$. But by definition, a cone over D uniquely factors through the limit, so that means that

- (i) if $\lambda_j \circ u = \lambda_j$ for all indices j , then $u = 1_L$.

Now suppose $[L', \lambda'_j]$ is another limit cone over D . Then $[L', \lambda'_j]$ uniquely factors through $[L, \lambda_j]$, via some f , so

- (ii) $\lambda_j \circ f = \lambda'_j$ for all j .

And likewise $[L, \lambda_j]$ uniquely factors through $[L', \lambda'_j]$ via some g , so

- (iii) $\lambda'_j \circ g = \lambda_j$ for all j .

Whence

(iv) $\lambda_j \circ f \circ g = \lambda_j$ for all j .

Therefore

(v) $f \circ g = 1_L$.

And symmetrically

(vi) $g \circ f = 1_{L'}$.

Whence f is not just unique (by hypothesis, the only way of completing the relevant diagrams to get the arrows to commute) but an isomorphism. \square

10.3 Limit cones as terminal objects

We have already seen that

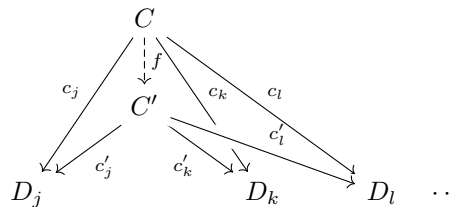
- (1) A terminal object in \mathcal{C} is ... wait for it! ... terminal in the given category \mathcal{C} .
- (2) The product of X with Y in \mathcal{C} is a terminal object in the derived category $\mathcal{C}_{W(X,Y)}$ of wedges to X and Y .
- (3) The equalizer of parallel arrows through X to Y in \mathcal{C} are (parts of) terminal objects in the derived category $\mathcal{C}_{F(X,Y)}$ of forks through X to Y .

Predictably, limit cones more generally are terminal objects in appropriate categories.

To spell this out, we first note that the cones $[C, c_j]$ over a given diagram D in \mathcal{C} form a category in a very natural way:

Definition 57. Given a diagram D in category \mathcal{C} , the derived category $\mathcal{C}_{C(D)}$ – the category of cones over D – has the following data:

- (1) Its objects are the cones $[C, c_j]$ over D .
- (2) An arrow from $[C, c_j]$ to $[C', c'_j]$ is any \mathcal{C} -arrow $f: C \rightarrow C'$ such that $c'_j \circ f = c_j$ for all indices j . In other words, for each D_j, D_k, D_l, \dots , in D , the corresponding triangle with remaining vertices C and C' commutes:



The identity arrow on a cone $[C, c_j]$ is the \mathcal{C} -arrow 1_C . And composition for arrows in $\mathcal{C}_{C(D)}$ is just composition of the corresponding \mathcal{C} -arrows. \triangle

Limits and colimits defined

It is entirely routine to confirm that $\mathcal{C}_{C(D)}$ is indeed a category. We can then recast our earlier definition of a limit cone as follows:

Definition 58. A *limit cone* for D in \mathcal{C} is a cone which is a terminal object in $\mathcal{C}_{C(D)}$. \triangle

And we now have an alternative proof of our last uniqueness result, Theorem 47:

Proof. Since a limit cone over D is terminal in $\mathcal{C}_{C(D)}$, it is unique in $\mathcal{C}_{C(D)}$ up to a unique isomorphism. But such an isomorphism in $\mathcal{C}_{C(D)}$ must be an isomorphism in \mathcal{C} commuting with the cones's arrows. \square

10.4 Results about limits

(a) Let's first prove two further simple theorems:

Theorem 48. Suppose $[L, \lambda_j]$ is a limit cone over a diagram D in \mathcal{C} , and $[L', \lambda'_j]$ is another cone over D which factors through $[L, \lambda_j]$ via an isomorphism f . Then $[L', \lambda'_j]$ is also a limit cone.

Proof. Take any cone $[C, c_j]$ over D . We need to show that (i) there is an arrow $v: C \rightarrow L'$ such that for all indices j for objects D_j in D , $c_j = \lambda'_j \circ v$, and (ii) v is unique.

But we know that there is a unique arrow $u: C \rightarrow L$ such that for j , $c_j = \lambda_j \circ u$. And we know that $f: L' \rightarrow L$ and $\lambda'_j = \lambda_j \circ f$ (so $\lambda_j = \lambda'_j \circ f^{-1}$).

Therefore put $v = f^{-1} \circ u$, and that satisfies (i).

Now suppose there is another arrow $v': C \rightarrow L'$ such that $c_j = \lambda'_j \circ v'$. Then we have $f \circ v': C \rightarrow L$, and also $c_j = \lambda_j \circ f \circ v'$. Therefore $[C, c_j]$ factors through $[L, \lambda_j]$ via $f \circ v'$, so $f \circ v' = u$. Whence $v' = f^{-1} \circ u = v$. Which proves (ii). \square

Theorem 49. Suppose $[L, \lambda_j]$ is a limit cone over a diagram D in \mathcal{C} . Then the cones over D with vertex C correspond one-to-one with \mathcal{C} -arrows from C to L .

Proof. Take any arrow $u: C \rightarrow L$. If there is an arrow $d: D_k \rightarrow D_l$ in the diagram D , then (since $[L, \lambda_j]$ is a cone), $\lambda_l = d \circ \lambda_k$, whence $(\lambda_l \circ u) = d \circ (\lambda_k \circ u)$. Since this holds generally, $[C, \lambda_j \circ u]$ is a cone over D . But (again since $[L, \lambda_j]$ is a limit) every cone over D with vertex C is of the form $[C, \lambda_j \circ u]$ for unique u . Hence there is indeed a one-one correspondence between arrows $u: C \rightarrow L$ and cones over D with vertex C . (Moreover, the construction is a natural one, involving no arbitrary choices.) \square

(b) We pause for a fun exercise and reality check, by remarking that the whole category \mathcal{C} can be thought of as the limiting case of a diagram in itself, and then

Theorem 50. *A category \mathcal{C} has an initial object if and only if \mathcal{C} , thought of as a diagram in \mathcal{C} , has a limit.*

Proof. Suppose \mathcal{C} has an initial object I . Then for every \mathcal{C} -object C , there is a unique arrow $\lambda_C: I \rightarrow C$. $[I, \lambda_C]$ is a cone (since for any arrow $f: C \rightarrow D$, the composite $f \circ \lambda_C$ is an arrow from I to D and hence has to be equal to the unique λ_D). Further, $[I, \lambda_C]$ is a limit cone. For suppose $[A, a_C]$ is any other cone over the whole of \mathcal{C} . Then since it is a cone, the triangle

$$\begin{array}{ccc} & A & \\ a_I \swarrow & & \searrow a_C \\ I & \xrightarrow{\lambda_C} & C \end{array}$$

has to commute for all C . But that's just the condition for $[A, a_C]$ factoring through $[I, \lambda_C]$ via a_I . And moreover, suppose $[A, a_C]$ also factors through by some u . Then in particular,

$$\begin{array}{ccc} & A & \\ u \swarrow & & \searrow a_I \\ I & \xrightarrow{1_I} & I \end{array}$$

commutes, and so $u = a_I$. So the factoring is unique, and $[I, \lambda_C]$ is a limit cone.

Now suppose, conversely, that $[I, \lambda_C]$ is a limit cone over the whole of \mathcal{C} . Then there is an arrow $\lambda_C: I \rightarrow C$ for each C in \mathcal{C} . If we can show it is unique, I will indeed be initial.

Suppose then that there is an arrow $k: I \rightarrow C$ for a given C . Then since $[I, \lambda_C]$ is a cone, the diagram

$$\begin{array}{ccc} & I & \\ \lambda_I \swarrow & & \searrow \lambda_C \\ I & \xrightarrow{k} & C \end{array}$$

has to commute. Considering the case where $k = \lambda_C$, we see that $[I, \lambda_C]$ factors through itself via λ_I ; but it also factors via 1_D , so the uniqueness of factorization entails $\lambda_I = 1_D$. Hence the diagram shows that for any $k: I \rightarrow C$ has to be identical to λ_C . So I is initial. \square

(c) Before proceeding further, let's introduce some standard notation:

Definition 59. We denote the limit object at the vertex of a given limit cone for the diagram D with objects D_j by ' $\lim_{\leftarrow j} D_j$ '. \triangle

Do note, however, that since limit cones are only unique up to isomorphism, different but isomorphic objects can be denoted in different contexts by ' $\lim_{\leftarrow j} D_j$ '.

The projection arrows from this limit object to the various objects D_j will then naturally be denoted ' $\lambda_i: \mathop{\text{Lim}}_{\leftarrow j} D_j \rightarrow D_i$ ', and the limit cone could therefore be represented by ' $[\mathop{\text{Lim}}_{\leftarrow j} D_j, \lambda_j]$ '. (The direction of the arrow under ' Lim ' in this notation is perhaps unexpected, but we just have to learn to live with it.)

10.5 Colimits defined

The headline, and thoroughly predictable, story about duals is: reverse the relevant arrows and you get a definition of colimits.

So, dualizing §10.2 and wrapping everything together, we get:

Definition 60. Let D be a diagram in category \mathcal{C} . Then a *cocone under D* is a \mathcal{C} -object C , together with an arrow $c_j: D_j \rightarrow C$ for each object D_j in D , such that whenever there is an arrow $d: D_k \rightarrow D_l$ in D , the following diagram commutes:

$$\begin{array}{ccc} D_k & \xrightarrow{\quad d \quad} & D_l \\ & \searrow c_k & \swarrow c_l \\ & & C \end{array}$$

The cocones under D form a category with objects the cocones $[C, c_j]$ and an arrow from $[C, c_j]$ to $[C', c'_j]$ being any \mathcal{C} -arrow $f: C \rightarrow C'$ such that $c'_j = f \circ c_j$ for all indexes j . A colimit for D is an initial object in the category of cocones under D . It is standard to denote the object at the vertex of the colimit cocone for the diagram D by ' $\mathop{\text{Lim}}_{\rightarrow j} D_j$ '. \triangle

It is now routine to confirm that our earlier examples of initial objects, co-products and co-equalizers do count as colimits.

- (1) The null case where we start with the empty diagram in \mathcal{C} gives rise to a cocone which is simply an object in \mathcal{C} . So the category of cocones over the empty diagram is just the category \mathcal{C} we started with, and a limit cocone is just an initial object in \mathcal{C} !
- (2) Consider now a diagram which is just *two* objects we'll call ' D_1 ', ' D_2 ', still with no arrow between them. Then a cocone over such a diagram is just a corner from D_1, D_2 (in the sense we met in §7.7); and a limit cocone in the category of such cocones is simply a coproduct.
- (3) And if we start with the diagram $D_1 \begin{array}{c} \xrightarrow{d} \\ \xrightarrow{d'} \end{array} D_2$ then a limit cocone over this diagram gives rise to a co-equalizer.

10.6 Pullbacks

(a) Let's illustrate all this by briefly exploring another kind of limit (in this section) and its dual (in the next section).

A co-wedge or, as I prefer to say (§7.7), a corner D in category \mathcal{C} is a diagram which can be represented like this:

$$\begin{array}{ccc} & D_2 & \\ & \downarrow e & \\ D_1 & \xrightarrow{d} & D_3 \end{array}$$

Now, a cone over our corner diagram has a rather familiar shape, i.e. it is a commutative square:

$$\begin{array}{ccc} C & \xrightarrow{c_2} & D_2 \\ \downarrow c_1 & \searrow c_3 & \downarrow e \\ D_1 & \xrightarrow{d} & D_3 \end{array}$$

Though note, we needn't really draw the diagonal here, for if the sides of the square commute thus ensuring $d \circ c_1 = e \circ c_2$, then we know a diagonal $c_3 = d \circ c_1$ exists making the triangles commute.

And a limit for this type of cone will be a cone with vertex $L = \lim_{\leftarrow j} D_j$ and three projections $\lambda_j: L \rightarrow D_j$ such that for any cone $[C, c_j]$ over D , there is a unique $u: C \rightarrow L$ such that this diagram commutes:

$$\begin{array}{ccccc} C & & \xrightarrow{c_2} & & D_2 \\ & \searrow u & & & \downarrow e \\ & & L & \xrightarrow{\lambda_2} & D_2 \\ & & \downarrow \lambda_1 & & \downarrow e \\ & & D_1 & \xrightarrow{d} & D_3 \end{array}$$

And note that if this commutes, there's just one possible $\lambda_3: L \rightarrow D_3$ and one possible $c_3: C \rightarrow D_3$ which can add to make a diagram that still commutes.

Definition 61. A limit for a corner diagram is a *pullback*. The square formed by the original corner and its limit, with or without its diagonal, is a *pullback square*. \triangle

(b) Let's immediately have a couple of examples of pullback squares living in the category \mathbf{Set} .

Limits and colimits defined

- (1) Changing the labelling, consider a corner comprising three sets X, Y, Z and a pair of functions which share the same codomain, thus:

$$\begin{array}{ccc} & & Y \\ & & \downarrow g \\ X & \xrightarrow{f} & Z \end{array}$$

We know from the previous diagram that the limit object L must be product-like (with any wedge over X, Y factoring through the wedge with vertex L). Hence to get the other part of the diagram to commute, the pullback square must have at its apex L something isomorphic to $\{\langle x, y \rangle \in X \times Y \mid f(x) = g(y)\}$ with the obvious projection maps to X and Y .

So suppose first that in fact both X and Y are subsets of Z , and the arrows into Z are both inclusion functions. And we then get a pullback square

$$\begin{array}{ccc} L & \longrightarrow & Y \\ \downarrow & & \downarrow i_2 \\ X & \xrightarrow{i_1} & Z \end{array}$$

with $L \cong \{\langle x, y \rangle \in X \times Y \mid x = y\} = \{\langle z, z \rangle \mid z \in X \cap Y\} \cong X \cap Y$. Hence, in **Set**, the intersection of a pair of sets is their pullback object (fixed, as usual, up to isomorphism).

- (2) Take another case in **Set**. Suppose we have a corner as before but with $Y = Z$ and $g = 1_Z$. Then

$$L \cong \{\langle x, z \rangle \in X \times Z \mid f(x) = z\} \cong \{x \mid \exists z f(x) = z\} \cong f^{-1}[Z],$$

i.e. a pullback object for this corner is, up to isomorphism, the inverse image of Z , and we have a pullback square

$$\begin{array}{ccc} f^{-1}[Z] & \longrightarrow & Z \\ \downarrow & & \downarrow 1_Z \\ X & \xrightarrow{f} & Z \end{array}$$

Hence in **Set**, the inverse image of a function is also a pullback object.

We will meet another simple example of pullbacks in **Set** in §12.4

- (c) Why ‘pullback’? Look at e.g. the diagram in (2). We can say that we get to $f^{-1}[Z]$ from Z by pulling back along f – or more accurately, we get to the arrow $f^{-1}[Z] \rightarrow X$ by pulling back the identity arrow on Z along f .

In this sense,

Theorem 51. *Pulling back a monomorphism yields a monomorphism.*

In other words, if we start with the same corner $X \xrightarrow{f} Z \xleftarrow{g} Y$ with g monic, and can pullback g along f to give a pullback square

$$\begin{array}{ccc} L & \xrightarrow{b} & Y \\ \downarrow a & & \downarrow g \\ X & \xrightarrow{f} & Z \end{array}$$

then the resulting arrow a is monic. (Note, this does not depend on the character of f .)

Proof. Suppose, for some arrows $C \xrightarrow{j} L, C \xrightarrow{k} L, a \circ j = a \circ k$. Then $g \circ b \circ j = f \circ a \circ j = f \circ a \circ k = g \circ b \circ k$. Hence, given that g is monic, $b \circ j = b \circ k$.

It follows that the two cones over the original corner, $X \xleftarrow{a \circ j} C \xrightarrow{b \circ j} Y$ and $X \xleftarrow{a \circ k} C \xrightarrow{b \circ k} Y$ are in fact the *same* cone, and hence must factor through the limit L via the same unique arrow $C \rightarrow L$. Which means $j = k$.

In sum, $a \circ j = a \circ k$ implies $j = k$, so a is monic. □

Here’s another result about monomorphisms and pullbacks:

Theorem 52. *The arrow $f: X \rightarrow Y$ is a monomorphism in \mathcal{C} if and only if the following is a pullback square:*

$$\begin{array}{ccc} X & \xrightarrow{1_X} & X \\ \downarrow 1_X & & \downarrow f \\ X & \xrightarrow{f} & Y \end{array}$$

Proof. Suppose this is pullback diagram. Then any cone $X \xleftarrow{a} C \xrightarrow{b} X$ over the corner $X \xrightarrow{f} Y \xleftarrow{f} X$ must uniquely factor through the limit with vertex X . That is to say, if $f \circ a = f \circ b$, then there is a u such that $a = 1_X \circ u$ and $b = 1_X \circ u$, hence $a = b$ – so f is monic.

Conversely, if f is monic, then given any cone $X \xleftarrow{a} C \xrightarrow{b} X$ over the original corner, $f \circ a = f \circ b$, whence $a = b$. But that means the cone factors through the cone $X \xleftarrow{1_X} X \xrightarrow{1_X} X$ via the unique a , making that cone a limit and the square a pullback square. □

(d) We’ve explained, up to a point, the label ‘pullback’. It should now be noted in passing that a pullback is sometimes called a *fibred product* (or fibre product) because of a construction of this kind on fibre bundles in topology. Those who know some topology can chase up the details.

Limits and colimits defined

But here's a way of getting products into the story, using an idea that we already know about. Remind yourself what slice categories are (Defn. 18). Then:

Theorem 53. *A pullback of a corner with vertex Z in a category \mathcal{C} is a product in the slice category \mathcal{C}/Z .*

Proof. Recall, an object of \mathcal{C}/Z , on the economical definition, is a \mathcal{C} -arrow $f: C \rightarrow Z$, and an arrow of \mathcal{C}/Z from $f: X \rightarrow Z$ to $g: Y \rightarrow Z$ is a \mathcal{C} -arrow $h: X \rightarrow Y$ such that $f = g \circ h$ in \mathcal{C} .

Now the pullback of the corner with vertex Z formed by f and g in \mathcal{C} is a pair of arrows $a: L \rightarrow X$ and $b: L \rightarrow Y$ such that $f \circ a = g \circ b (= k)$ and which form a wedge such that any other wedge $a': L' \rightarrow X$, $b': L' \rightarrow Y$ such that $f \circ a' = g \circ b' (= k')$ factors uniquely through it.

Looked at as a construction in \mathcal{C}/Z , this means taking two \mathcal{C}/Z -objects f and g and getting a pair of \mathcal{C}/Z -arrows $a: k \rightarrow f$, $b: k \rightarrow g$. And this pair of arrows forms a wedge such that any other wedge $a': k' \rightarrow f$, $b': k' \rightarrow g$ factors uniquely through it. In other words, the pullback in \mathcal{C} constitutes a product in \mathcal{C}/Z . \square

(e) Because of that kind of connection, product notation is often used for pullbacks, thus:

$$\begin{array}{ccc} X \times_Z Y & \longrightarrow & Y \\ \downarrow & \lrcorner & \downarrow \\ X & \longrightarrow & Z \end{array}$$

with the subscript giving the vertex of the corner we are taking a limit over, and with the little corner-symbol in the diagram conventionally indicating it is indeed a pullback square.

10.7 Pushouts

Pullbacks are limits for corners. What is a colimit for a corner? Check the relevant diagram and it is obviously the corner itself. So the potentially interesting dualization of the notion of a pullback is when we take the colimit of 'co-corners', i.e. wedges.

Suppose then we take a wedge D , i.e. a diagram $D_1 \xleftarrow{d} D_3 \xrightarrow{e} D_2$. A cocone under this diagram is another commutative square (omitting again the diagonal arrow which is fixed by the others).

$$\begin{array}{ccc} D_3 & \xrightarrow{e} & D_2 \\ \downarrow d & & \downarrow c_2 \\ D_1 & \xrightarrow{c_1} & C \end{array}$$

And a limit cocone of this type will be a cocone with apex $L = \mathop{\text{Lim}}_{\rightarrow j} D_j$ and projections $\lambda_j: D_j \rightarrow L$ such that for any cocone $[C, c_j]$ under D , there is a unique $u: L \rightarrow C$ such that the obvious dual of the whole pullback diagram above commutes.

Definition 62. A limit for a wedge diagram is a *pushout*. △

Now, in **Set**, we get the limit object for a corner diagram $X \xrightarrow{f} Z \xleftarrow{g} Y$ by taking a certain *subset* of a *product* $X \times Y$. Likewise we get the colimit object for a wedge diagram $X \xleftarrow{f} Z \xrightarrow{g} Y$ by taking a certain *quotient* of a *coproduct* $X \amalg Y$. We won't, however, pause further over this now. Though it does again illustrate how taking colimits can tend to beget messier constructions than taking limits.

11 The existence of limits

We have seen that a whole range of very familiar constructions from various areas of ordinary mathematics can be regarded as instances of taking limits or colimits of (very small) diagrams in appropriate categories. Examples so far include: forming cartesian products or logical conjunctions, taking disjoint unions or free products, quotienting out by an equivalence relation, taking intersections, taking inverse images.

Not *every* familiar kind of construction in a category \mathcal{C} involves taking a limit cone or cocone in \mathcal{C} : we'll meet a couple of important exceptions in the next two chapters. But plainly we are mining a very rich seam here – and we are already making good on our promise to show how category theory helps reveal recurring patterns across different areas of mathematics. So what more can we say about limits?

It would get tedious to explore case by case what it takes for a category to have limits for various further kinds of diagram, even if we just stick to considering limits over tiny diagrams. But fortunately we don't need to do such a case-by-case examination. In this chapter we show that if a category has certain basic limits of kinds that we have already met, then it has *all* finite limits (or more).

11.1 Pullbacks, products and equalizers related

(a) Here's an obvious definition:

Definition 63. The category \mathcal{C} has *all finite limits* if for any finite diagram D – i.e. for any diagram whose objects are D_j for indices $j \in J$, where J is a finite set – \mathcal{C} has a limit over D . A category with all finite limits is said to be *finitely complete*. \triangle

Our main target theorems for this chapter are then as follows:

Theorem 54. *If \mathcal{C} has a terminal object, and has all binary products and equalizers, it is finitely complete.*

Theorem 55. *If \mathcal{C} has a terminal object, and has a pullback for any corner, it is finitely complete.*

11.1 Pullbacks, products and equalizers related

(These theorems explain why we have chosen exactly our earlier examples of limits to explore!) Later, in §11.3, we will see how that we can very easily get an analogous result for limits over infinite diagrams; but it will help to fix ideas if we initially focus on the finite case. And of course, our theorems will have the predictable duals: we briefly mention them in §11.4.

We begin though, in this section, by proving the following much more restricted versions of our two stated theorems, versions which talk just about products, equalizers and pullbacks rather than about limits more generally:

Theorem 56. *If a category \mathcal{C} has all binary products and equalizers, then it has a pullback for any corner.*

Theorem 57. *If \mathcal{C} has a terminal object, and has a pullback for any corner, then it has all binary products and all equalizers.*

Proving these cut-down results first will have a double pay-off:

- (1) We afterwards only need prove one of Theorems 54 and 55, since in the presence of the restricted theorems, the stronger theorems evidently imply each other. We will in fact later concentrate on proving Theorem 54 (leaving Theorem 55 as a simple corollary given Theorem 57).
- (2) Our proof of the restricted Theorem 56 will provide an instructive guide to how to do establish the more general Theorem 54.

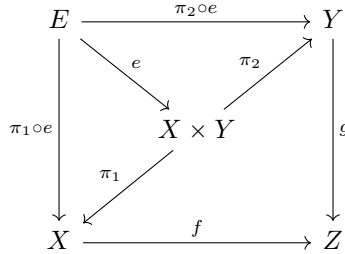
(b) For those rather nobly trying, as we go along, to prove stated theorems before looking at the proofs, the results in this chapter do require a little more thought than what's gone before. Even so, a little exploration should still reveal the only reasonable proof-strategies.

Proof for Theorem 56. Given an arbitrary corner $X \xrightarrow{f} Z \xleftarrow{g} Y$ we need to construct a pullback.

There is nothing to equalize yet. So our only option is to start by constructing some product. By assumption, \mathcal{C} has binary products, so there will in particular be a product $X \times Y$ and also a triple product $X \times Y \times Z$. Now in fact, when we come to generalize our proof strategy for this theorem to prove Theorem 54, it will be the product of every object in sight that we'll need to work with. But because of special features of the present case, it is enough to consider the simpler product. So: take the product $X \times Y$ with the usual projections $\pi_1: X \times Y \rightarrow X$ and $\pi_2: X \times Y \rightarrow Y$.

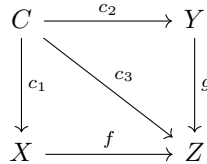
This immediately gives us parallel arrows $X \times Y \begin{array}{c} \xrightarrow{f \circ \pi_1} \\ \xrightarrow{g \circ \pi_2} \end{array} Z$. And because \mathcal{C} has equalizers, this parallel pair must have an equalizer $[E, e]$, for which $f \circ \pi_1 \circ e = g \circ \pi_2 \circ e$. Which in turn means that the following diagram commutes:

The existence of limits

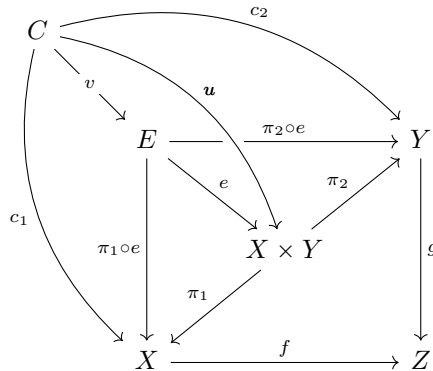


Claim: the wedge formed by E with the projections $\pi_1 \circ e, \pi_2 \circ e$ is indeed a pullback of the corner $X \xrightarrow{f} Z \xleftarrow{g} Y$.

From this point, the argument is just fairly routine checking. Consider any other cone over the original corner



In other words, leaving the diagonals to take care of themselves, consider any wedge $X \xleftarrow{c_1} C \xrightarrow{c_2} Y$ with $f \circ c_1 = g \circ c_2$: we need to show that this factors uniquely through E .



Now, our wedge certainly uniquely factors through the product $X \times Y$, so there is a unique $u: C \rightarrow X \times Y$ such that $c_1 = \pi_1 \circ u, c_2 = \pi_2 \circ u$. Hence $f \circ \pi_1 \circ u = g \circ \pi_2 \circ u$. Therefore $C \xrightarrow{u} X \times Y \xrightarrow{\frac{f \circ \pi_1}{g \circ \pi_2}} Z$ is a fork, which must factor uniquely through the equalizer E via some v .

That is to say, there is a $v: C \rightarrow E$ such that $e \circ v = u$. Hence $\pi_1 \circ e \circ v = \pi_1 \circ u = c_1$. Similarly $\pi_2 \circ e \circ v = c_2$. Therefore the wedge with vertex C indeed factors through E , as we need.

11.1 Pullbacks, products and equalizers related

To finish the proof, we have to establish the uniqueness of the mediating arrow v . Suppose then that $v': C \rightarrow E$ also makes $\pi_1 \circ e \circ v' = c_1$, $\pi_2 \circ e \circ v' = c_2$. Then the wedge $X \xleftarrow{c_1} C \xrightarrow{c_2} Y$ factors through $X \times Y$ via $e \circ v'$; but we know the wedge factors uniquely through the product $X \times Y$ by u . Therefore $e \circ v' = u = e \circ v$.

But equalizers are monic by Theorem 42, so $v' = v$, and we are done. \square

Proof for Theorem 57. Given that \mathcal{C} has a terminal object, what corners are guaranteed to exist, for any given X, Y ? Evidently $X \longrightarrow 1 \longleftarrow Y$. So take a pullback over this corner. Applying the definition, we immediately find that a pullback for such a corner is indeed just the product $X \times Y$ with its usual projection arrows.

To show that \mathcal{C} has equalizers, given that it has pullbacks and hence products, start by thinking of the parallel arrows we want to equalize, say $X \xrightarrow{f} Y \xrightarrow{g}$, as a wedge $Y \xleftarrow{f} X \xrightarrow{g} Y$. This wedge will factor uniquely via an arrow $\langle f, g \rangle$ through the product $Y \times Y$ (which exists by hypothesis).

So now consider the corner $X \xrightarrow{\langle f, g \rangle} Y \times Y \xleftarrow{\delta_Y} Y$, where δ_Y is the ‘diagonal’ arrow (see Defn. 42). This is nice to think about since (to arm-wave a bit!) the first arrow is evidently related to the parallel arrows we want to equalize, and the second arrow does some equalizing.

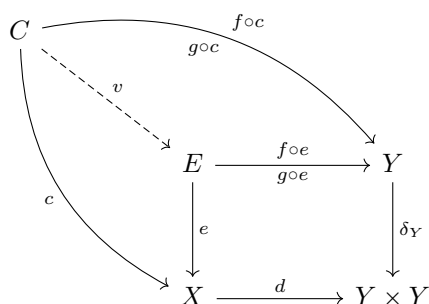
Now take this corner’s pullback (the only thing to do with it!):

$$\begin{array}{ccc} E & \xrightarrow{q} & Y \\ \downarrow e & & \downarrow \delta_Y \\ X & \xrightarrow{\langle f, g \rangle} & Y \times Y \end{array}$$

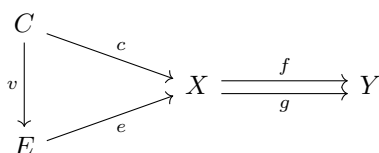
Intuitively speaking, $E \xrightarrow{e} X \xrightarrow{\langle f, g \rangle} Y \times Y$ sends something in E to a pair of equals. So, morally, $[E, e]$ ought to be an equalizer for $X \xrightarrow{f} Y \xrightarrow{g}$. And, from this point on, it is a routine proof to check that it indeed is an equalizer. Here goes:

By the commutativity of the pullback square, $\delta_Y \circ q = \langle f, g \rangle \circ e$. Appealing to Theorems 30, 32 and 33, it follows that $\langle q, q \rangle = \langle f \circ e, g \circ e \rangle$, and hence $f \circ e = q = g \circ e$. Therefore $E \xrightarrow{e} X \xrightarrow{f} Y \xrightarrow{g}$ is a fork. It remains to show that it is a limit fork.

Take any other fork $C \xrightarrow{c} X \xrightarrow{f} Y \xrightarrow{g}$. The wedge $X \xleftarrow{c} C \xrightarrow{f \circ c} Y$ must factor through E (because E is the vertex of the pullback) via a unique mediating arrow v :



It follows that v makes this diagram commute:



And any $v': C \rightarrow E$ which makes the latter diagram commute will also be a mediating arrow making the previous diagram commute, so $v' = v$ by uniqueness of mediators in pullback diagrams. Hence $[E, e]$ is indeed an equalizer. \square

11.2 Categories with all finite limits

Our target now is to establish the promised main result:

Theorem 54. *If \mathcal{C} has a terminal object, and has all binary products and equalizers, it is finitely complete.*

This is indeed our first Big Result. To prove it, we are going to generalize the strategy pursued in proving the cut-down result that having binary products and equalizers implies at least having pullbacks. So, the outline plan is this:

Given a finite diagram D , we start by forming the product P of the objects from D (which we can do since \mathcal{C} has all finite products). We then find some appropriate parallel arrows out of this product P . Then we take an equalizer $[E, e]$ of these arrows (which we can do since \mathcal{C} has all equalizers). We then aim to use E as the vertex of the desired limit cone over the diagram D on the model of the proof of Theorem 56.

The devil, of course, is in the details! And to be frank, you won't lose much if you skip past them.

Consider again the proof of Theorem 56. There we started with a mini-diagram D , i.e. a corner with two arrows sharing a target, $f: X \rightarrow Z$, $g: Y \rightarrow Z$. We

got parallel arrows which share a source as well as a target by taking a product, thereby getting $X \times Y \begin{array}{c} \xrightarrow{f \circ \pi_1} \\ \xrightarrow{g \circ \pi_2} \end{array} Z$. And *then* we could look for an equalizer.

Now, in an arbitrary finite diagram D there could be lots of arrows of the kind $d: D_k \rightarrow D_l$ with a variety of different sources and targets. But we still want to end up by constructing out of them a pair of parallel arrows with the same source and same target so that we can then take an equalizer. To construct the single source and single target we use products again.

At the source end, we have two apparent options – we could take the product $[P, p_j]$ of *all* the objects in D , or we could take the product $[P', p'_j]$ of those objects in D which are sources of arrows in D . It turns out, after a bit of exploration, that in the general case the first is the one to go for. At the target end, the natural thing to do is to define $[Q, q_l]$ as the product of all the objects D_l which are targets for arrows in D . (We can make these constructions of course as we are assuming we are working in a category with all finite products).

So the name of the game is now to define a pair of parallel arrows

$$P \begin{array}{c} \xrightarrow{v} \\ \xrightarrow{w} \end{array} Q$$

which we are going to equalize by some $[E, e]$.

However, there are in fact only two naturally arising arrows from P to Q .

- (1) Consider first a certain cone over the objects D_l which contribute to the product Q – namely, the cone with vertex P and with an arrow $p_l: P \rightarrow D_l$ for each D_l . This cone (by definition of the product $[Q, q_l]$) must factor through the product by a unique mediating arrow v , so that $p_l = q_l \circ v$ for each l .
- (2) Consider secondly the cone over the same objects with vertex P and an arrow $d \circ p_k: P \rightarrow D_l$ for each arrow $d: D_k \rightarrow D_l$ in D . This cone too must factor through the product $[Q, q_l]$ by a unique mediating arrow w , so that $d \circ p_k = q_l \circ w$ for each arrow $d: D_k \rightarrow D_l$.

Since we are assuming that all parallel arrows have equalizers in \mathcal{C} , we can take the equalizer of v and w , namely $[E, e]$.

And now the big claim, modelled exactly on the key claim in our proof of Theorem 56: $[E, p_j \circ e]$ will be a limit cone over D .

Let's state this as a theorem:

Theorem 58. *Let D be a finite diagram in a category \mathcal{C} which has a terminal object, binary products and equalizers. Let $[P, p_j]$ be the product of the objects D_j in D , and $[Q, q_l]$ be the product of the objects D_l which are targets of arrows in D . Then there are arrows*

$$P \begin{array}{c} \xrightarrow{v} \\ \xrightarrow{w} \end{array} Q$$

such that the following diagrams commute for each $d: D_k \rightarrow D_l$:

The existence of limits

$$\begin{array}{ccc}
 P & \xrightarrow{v} & Q \\
 & \searrow p_i & \downarrow q_i \\
 & & D_l
 \end{array}
 \qquad
 \begin{array}{ccc}
 P & \xrightarrow{w} & Q \\
 p_k \downarrow & & \downarrow q_l \\
 D_k & \xrightarrow{d} & D_l
 \end{array}$$

Let the equalizer of v and w be $[E, e]$. Then $[E, p_j \circ e]$ will be a limit cone over D in \mathcal{C} .

Proof. We have already shown that v and w exist such that the given diagrams commute and that an equalizer $[E, e]$ for them exists. So next we confirm $[E, p_j \circ e]$ is a cone. Suppose then that there is an arrow $d: D_k \rightarrow D_l$. For a cone, we require $d \circ p_k \circ e = p_l \circ e$.

But indeed $d \circ p_k \circ e = q_l \circ w \circ e = q_l \circ v \circ e = p_l \circ e$, where the inner equation holds because e is an equalizer of v and w and the outer equations are given by the commuting diagrams above.

Second we show that $[E, p_j \circ e]$ is a limit cone. So suppose $[C, c_j]$ is any other cone over D . Then there must be a unique $u: C \rightarrow P$ such that every c_j factors through the product and we have $c_j = p_j \circ u$.

Since $[C, c_j]$ is a cone, for any $d: D_k \rightarrow D_l$ in D we have $d \circ c_k = c_l$. Hence $d \circ p_k \circ u = p_l \circ u$, and hence for each q_l , $q_l \circ w \circ u = q_l \circ v \circ u$. But then we can apply the obvious generalized version of Theorem 31, and conclude that $w \circ u = v \circ u$. Which means that

$$C \xrightarrow{u} P \begin{array}{c} \xrightarrow{v} \\ \xrightarrow{w} \end{array} Q$$

is a fork, which must therefore uniquely factor through the equalizer $[E, e]$. That is to say, there is a unique $s: C \rightarrow E$ such that $u = e \circ s$, and hence for all j , $c_j = p_j \circ u = p_j \circ e \circ s$. That is to say, $[C, c_j]$ factors uniquely through $[E, p_j \circ e]$ via s . Therefore $[E, p_j \circ e]$ is indeed a limit cone. \square

This more detailed result of course trivially implies the less specific Theorem 54. And that in turn, given Theorem 57, gives us Theorem 55. So we are done.

Given ingredients from our previous discussions, since the categories in question have terminal objects, binary products and equalizers,

Theorem 59. *Set and FinSet are finitely complete, as are categories of algebraic structured sets such as Mon, Grp, Ab, Rng. Similarly Top is finitely complete.*

While e.g. a poset-as-a-category may lack many products and hence not be finitely complete.

11.3 Infinite limits

Now we extend our key Theorem 54 to reach beyond the finite case. First, we need:

Definition 64. The category \mathcal{C} has all small limits if for any diagram D whose objects are D_j for indices $j \in I$, for some set I , then \mathcal{C} has a limit over D . A category with all small limits is also said to be *complete*. \triangle

Again, as in talking of small products, small limits can be huge – we just mean no-bigger-than-set-sized. An easy inspection of the proof in the last section shows that, given our requirement that the objects in a diagram D can be indexed by a set, the argument will continue to go through just as before – assuming, that is, that we are still dealing with a category like **Set** which has products for all set-sized collections of objects (so we can still form the products $[P, p_j]$ and $[Q, q_i]$) and also all equalizers.

Hence, without further ado, we can state:

Theorem 60. *If \mathcal{C} has all small products and has equalizers, then it has all small limits, i.e. is complete.*

We can similarly extend Theorem 59 to show that

Theorem 61. *Set is complete – as are the categories of structured sets **Mon**, **Grp**, **Ab**, **Rng**. **Top** too is complete.*

We have already met a category which, by contrast, is finitely complete but is evidently not complete, namely **FinSet**.

11.4 Dualizing again

Needless to say by this stage, our results in this chapter dualize in obvious ways. Thus we need not delay over the further explanations and proofs of

Theorem 62. *If \mathcal{C} has initial objects, binary coproducts and co-equalizers, then it has all finite colimits, i.e. is finitely cocomplete. If \mathcal{C} has all small coproducts and has co-equalizers, then it has all small colimits, i.e. is cocomplete.*

Theorem 63. *Set is cocomplete – as are the categories of structured sets **Mon**, **Grp**, **Ab**, **Rng**. **Top** too is cocomplete.*

But note that a category can of course be (finitely) complete without being (finitely) cocomplete and vice versa. For a generic source of examples, take again a poset (P, \leq) considered as a category. This automatically has all equalizers (and coequalizers) – see §9.1 Ex. (5). But it will have other limits (colimits) depending on which products (coproducts) exists, i.e. which sets of elements have suprema (infima). For a simple case, take a poset with a maximum element and such that every pair of elements has a supremum: then considered as a category it has all finite limits (but maybe not infinite ones). But it need not have a minimal element and/or infima for all pairs of objects: hence it can lack some finite colimits despite having all finite limits.

12 Subobjects

We have seen how to treat the results of various familiar operations, such as forming products or taking quotients, as limits or colimits. But as we said at the beginning of the last chapter, not every familiar kind of construction when treated categorically straightforwardly involves taking a limit or colimit. We'll consider a couple of examples. In the next chapter, we look at exponentials. But first, in this chapter, we consider taking subobjects (as in subsets, subgroups, subspaces, etc.).

12.1 Subsets revisited

(a) We start though in familiar vein, still thinking about limits (or more particularly, equalizers). In §9.1, we saw that in **Set**, given two parallel arrows from an object X , a certain subset of X together with the trivial inclusion function provides an equalizer for those arrows – and §9.2 tells us that this is the unique equalizer, up to isomorphism.

We now note that a reverse result holds too:

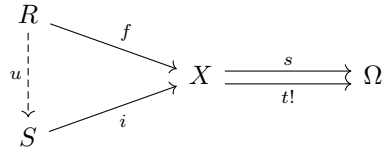
Theorem 64. *In **Set**, any subset S of X , taken together with its natural inclusion map $i: S \rightarrow X$, forms an equalizer for certain parallel arrows from X .*

Proof. Let Ω be some *truth-value object*, i.e. a two-object set with members identified as *true* and *false*. Setting $\Omega = \{0, 1\}$, with 1 as *true* and 0 as *false* is of course the choice hallowed by tradition.

Then a subset $S \subseteq X$ has an associated *characteristic function* $s: X \rightarrow \Omega$ which sends $x \in X$ to *true* if $x \in S$ and sends x to *false* otherwise.

Let $t: 1 \rightarrow \Omega$ be the map which sends the sole object in the singleton 1 to *true*, and let $t!$ be the composite map $X \xrightarrow{!x} 1 \xrightarrow{t} \Omega$.

We show that $[S, i]$ is an equalizer for the parallel arrows $s, t!: X \rightarrow \Omega$. First, it is trivial that $s \circ i = t! \circ i$, so as required $S \xrightarrow{i} X \begin{array}{c} \xrightarrow{s} \\ \xrightarrow{t!} \end{array} \Omega$ is indeed a fork. It remains to confirm that any upper fork in this next diagram factors through the lower fork via a unique mediating u :



Recycling an argument we've seen before, since $s \circ f = t! \circ i$ by assumption, it is immediate that $f[R] \subseteq S \subseteq X$. Hence, if we define $u: R \rightarrow S$ to agree with $f: R \rightarrow X$ on all inputs, then the diagram commutes. And this u is evidently the only arrow to give us a commuting diagram. \square

(b) Now, given these results relating subsets to certain equalizers, we might perhaps expect to meet at this point a general account of subobjects in terms of equalizers. And yes, we do indeed get a general connection, in appropriate categories, between subobjects and limits involving so-called truth-value objects like Ω . However, as we will later explain in §12.4, *this connection has to be read as fixing the general notion of a truth-value object in terms of the notion of a subobject rather than the other way around*. Hence we need a prior account of subobjects: we give it in the next section.

12.2 Subobjects as monic arrows

(a) Work in **Set** again. And note that any injective set-function $f: S \rightarrow X$ sets up a bijection $j: S \xrightarrow{\sim} f[S] \subseteq X$. In other words, any monic arrow $S \rightarrow X$ generates an isomorphism between S and a subset of X . So, if we only care about identifications up to isomorphism (the typical situation in category theory), then an object S together with a monic arrow $S \rightarrow X$ might as well be treated as a subobject of X in **Set**. And then noting that an arrow determines its source so we needn't really mention that separately, and generalizing to other categories, this suggests a very simple definition:

Definition 65. A *subobject*₁ of an object X in the category \mathcal{C} is just a monomorphism $S \rightarrow X$. \triangle

(b) Subobjects are arrows and so we can't immediately talk about subobjects of subobjects. But there is a natural definition of subobject inclusion:

Definition 66. If $f: A \rightarrow X$ and $g: B \rightarrow X$ are subobjects₁ of X , then f is *included in* g , in symbols $f \subseteq g$ iff f factors through g , i.e. there is an arrow $h: A \rightarrow B$ such that $f = g \circ h$. \triangle

Question: Wouldn't it be more natural to also require the mediating arrow h to be monic too? Answer: We don't need to write that into the definition because h is monic by Theorem 9 (3).

It is then trivial to check that inclusion of subobjects, so defined, is reflexive and transitive. So far so good.

12.3 Subobjects as isomorphism classes

(a) However, if we adopt our first definitions of subobject and subobject-inclusion, we get some oddities.

- (1) In **Set**, for example, the singleton set $\{1\}$ would have not two subobjects as you might expect (the empty set and itself) but infinitely many. Indeed it would have too many subobjects to form a set, since there are as many monic arrows $S \rightarrow \{1\}$ as there are singleton sets S , and there are too many singletons to form a set.
- (2) Again in **Set** for example, two subobjects of X , $f: A \rightarrow X$ and $g: B \rightarrow X$, can be such that $f \subseteq g$ and $g \subseteq f$ even though $f \neq g$.

We know from Theorem 15 that if $f \subseteq g$ and $g \subseteq f$, i.e. if the two arrows factor through each other, then they factor via an isomorphism, so we'll have $A \cong B$. But we needn't have $A = B$ which would be required for the arrows f, g to be identical. So the subobjects of X ordered by inclusion needn't form a poset.

Arguably, neither is a happy consequence of our definitions so far.

(b) An obvious suggestion for keeping tallies of subobjects under control is to say that the monic arrows $f: S \rightarrow X$, $g: S' \rightarrow X$ should count as representing the same subobject of X iff $S \cong S'$. Or by Theorem 15 again, we could equivalently say:

Definition 67. A *subobject*₂ of X is a class of *subobjects*₁ of X which factor through each other. △

We can then show that

Theorem 65. In **Set**, the *subobjects*₂ of X correspond one-to-one with the subsets of X .

Proof. First, we remark that monic arrows $f: S \rightarrow X$, $g: S' \rightarrow X$ belong to the same *subobject*₂ of X if and only if f and g have the same image.

For suppose there is an isomorphism $i: S \rightarrow S'$ such that $f = g \circ i$. Therefore if $x \in f[S]$, then there is an $s \in S$ such that $x = f(s) = g(i(s))$ where $i(s) \in S'$, so $x \in g[S']$. Hence $f[S] \subseteq g[S']$. Likewise $g[S'] \subseteq f[S]$. Hence if f and g belong to the same *subobject*₂, they have the same image.

Conversely, suppose the monic arrows $f: S \rightarrow X$, $g: S' \rightarrow X$ have the same image. In **Set** monics are injections; so we can define a map i which sends s to the unique s' which that $g(s') = f(s)$, and then trivially $f = g \circ i$. Likewise g factors through f , and hence f and g belong to the same *subobject*₂ of X .

Now take any subset $S \subseteq X$. There is a corresponding monic inclusion function $f_S: S \rightarrow X$. So consider the map that sends a subset S to the *subobject*₂ which contains f_S . This is one-one and onto. It is one-one because if the *subobject*₂

12.4 Subobjects, equalizers, and pullbacks

which contains f_S is the subobject₂ which contains $f_{S'}$, then f_S has the same image as $f_{S'}$, and being inclusions it follows that $S = S'$. It is onto because the functions in any subobject₂ of X with the shared image $S \subseteq X$ will contain such an f_S . \square

We get parallel results in other categories too. For example, subobject₂s in the category \mathbf{Grp} correspond one-to-one to subgroups, in the category \mathbf{Vect}_k correspond to vector subspaces, and so on. (But topologists might like to work out why in \mathbf{Top} the subobject₂s don't straightforwardly correspond to subspaces.)

Suppose we now add

Definition 68. If $\llbracket f \rrbracket$ and $\llbracket g \rrbracket$ are subobject₂s of X , respectively the isomorphism classes containing $f: A \rightarrow X$ and $g: B \rightarrow X$, then $\llbracket f \rrbracket$ is included in $\llbracket g \rrbracket$, in symbols $\llbracket f \rrbracket \subseteq \llbracket g \rrbracket$ iff $f \subseteq g$. \triangle

It is routine to check that this definition of an order relation on isomorphism classes is independent of the chosen exemplar of the class. And then inclusion so defined is indeed reflexive, and \subseteq is a partial order – and hence the subobject₂s of X , with this ordering, form a poset as intuitively we want.

(c) Given the way subobject₂s more naturally line up with subsets, subgroups, etc., as normally conceived, many authors prefer Defn. 67 as their official categorical account of subobjects – see for example (Goldblatt, 2006, p. 77), Leinster (2014, Ex. 5.1.40). But some authors prefer the first simple definition of subobject as monics as is given by e.g. Awodey (2006, §5.1). While Johnstone (2002, p. 18) says that ‘like many writers on category theory’ he will be deliberately ambiguous between the two definitions in his use of ‘subobject’, which sounds an unpromising line but in practice works quite well!

12.4 Subobjects, equalizers, and pullbacks

(a) How does our official account of subobjects in either form relate to our previous thought that we can treat subobjects, or at least subsets, as special equalizers?

Working in \mathbf{Set} again, it is easily checked that if $i: S \rightarrow X$ is any monic arrow into X (not necessarily an inclusion map), and $s: X \rightarrow \Omega$ is now the map that sends $x \in X$ to *true* iff $x \in i[S]$, then $S \xrightarrow{i} X \begin{matrix} \xrightarrow{s} \\ \xrightarrow{t!} \end{matrix} \Omega$ is still a fork.

Indeed, it is a limit fork such that any other fork through $s, t!$ factors uniquely through it. For take again the diagram

$$\begin{array}{ccccc}
 R & & & & \\
 \vdots & \searrow f & & & \\
 u \downarrow & & & X & \begin{matrix} \xrightarrow{s} \\ \xrightarrow{t!} \end{matrix} \Omega \\
 S & \nearrow i & & &
 \end{array}$$

Subobjects

Since $s \circ f = t \circ f$ by assumption, it is immediate that $f[R] \subseteq i[S] \subseteq X$. Hence, if we define u to send an object $r \in R$ to the pre-image of $f(r)$ under i (which is unique since i is monic), then the diagram commutes. And this u is evidently the only arrow to give us a commuting diagram. So, the subobject₁ $i: S \rightarrow X$ (together with its source) is still an equalizer in \mathbf{Set} (and so a subobject₂ can be thought of as a class of equalizers).

(b) It is now interesting to note an equivalent way of putting the situation in \mathbf{Set} . For note that the map $t! \circ i: S \rightarrow \Omega$, which sends everything in S to the value *true*, is of course trivially equal to the composite map $S \xrightarrow{!_S} 1 \xrightarrow{t} \Omega$ with 1 a terminal object in the category. Similarly for the map $t! \circ f: R \rightarrow \Omega$. Hence, the claim that $[S, i]$ equalizes $s, t!$ in \mathbf{Set} is equivalent to the following. For any $f: R \rightarrow X$ such that $s \circ f = t \circ 1_R$ there is a unique u which makes the whole diagram commute:

$$\begin{array}{ccccc}
 R & & & & \\
 \downarrow f & \searrow u & & \searrow !_R & \\
 & S & \xrightarrow{!_S} & 1 & \\
 & \downarrow i & & \downarrow t & \\
 & X & \xrightarrow{s} & \Omega &
 \end{array}$$

And after our work in §10.6, we know a snappy way of putting that: *the lower square is a pullback square.*

(c) Now, we can indeed carry this last idea across to other categories. We can say that, in a category \mathcal{C} with a terminal object, then given a truth-value object Ω and a *true*-selecting map $t: 1 \rightarrow \Omega$, then for any subobject₁ of X , i.e. for any monic $i: S \rightarrow X$, there is a unique ‘characteristic function’ $s: X \rightarrow \Omega$ which makes

$$\begin{array}{ccc}
 S & \xrightarrow{!_S} & 1 \\
 \downarrow i & & \downarrow t \\
 X & \xrightarrow{s} & \Omega
 \end{array}$$

a pullback square.

However, now to pick up the thought we trailed at the end of §12.1, we *can't* regard this as an alternative definition of a subobject in terms of a limit – since that would presuppose we *already* have a handle on a general notion of truth-value object, and we don't. Rather, we need to look at things the other way about. What we have here is a general characterization of what can sensibly be counted as a ‘truth-value object’ Ω and an associated *true*-selecting map

$t: 1 \rightarrow \Omega$ in a category \mathcal{C} with a terminal object. We define such things across categories by requiring that they work as ‘subobject classifiers’, i.e. by requiring they together ensure the displayed square is a pullback for a unique s given any monic subobject arrow i . We will eventually return to this point.

12.5 Elements and subobjects

A final remark. Earlier we noted that, in **Set**, functions $\vec{x}: 1 \rightarrow X$ correspond one-to-one with elements of X , and so started treating arrows \vec{x} as the categorial version of *set elements*. And inspired by that, we then called arrows $f: S \rightarrow X$ *generalized elements* of X . Yet now we have some of those same arrows, namely the monic ones, $i: S \rightarrow X$ being offered as the categorial version of *subsets*.

Now, one of the things that is drilled into us early is that we must very sharply distinguish the notion of element from the notion of subset. Yet here we seem to be categorially assimilating the notions – elements and subsets of X both get rendered by arrows in **Set**, and a subset-of- X -qua-subobject will count as a special kind of (generalized) element-of- X . Is this a worry? For the moment we just flag up the apparent anomaly: this is something else we will want to say more about later, in talking about the category theorist’s view of sets more generally.

13 Exponentials

We will eventually have much more to say about limits, and in particular about how they can get ‘carried over’ from one category to another by maps *between* categories. For the moment, however, we pause to consider another categorial notion that applies *within* a category, one that is also defined in terms of a ‘universal mapping property’, but which isn’t straightforwardly a limit – namely the notion of an exponential.

13.1 Two-place functions

First however, let’s pause to revisit the issue of two-place functions in category theory which we shelved in §7.3 (d).

It might in fact be helpful to recall how a couple of other familiar frameworks manage to do without genuine multi-place functions by providing workable substitutes:

- (1) Set-theoretic orthodoxy models a two-place total function from numbers to numbers (addition, say) as a function $f: \mathbb{N}^2 \rightarrow \mathbb{N}$. Here, \mathbb{N}^2 is the cartesian product of \mathbb{N} with itself, i.e. is the set of ordered pairs of numbers. And an ordered pair is *one* thing not two things. So a function $f: \mathbb{N}^2 \rightarrow \mathbb{N}$ is in fact strictly speaking a *unary function*, a function that maps *one* argument, an ordered pair object, to a value, not a real binary function.

Of course, in set-theory, for any two things there is a pair-object that codes for them – we usually choose a Kuratowski pair – and so we can indeed trade in a function from two objects for a related function from the corresponding pair-object. And standard notational choices can make the trade quite invisible. Suppose we adopt, as we earlier did, the modern convention of using ‘ (m, n) ’ as our notation for the ordered pair of m with n . Then ‘ $f(m, n)$ ’ invites being parsed either way, as representing a two-place function $f(\cdot, \cdot)$ with arguments m and n , or as a corresponding one-place function $f\cdot$ with the single argument, the pair (m, n) . But note: the fact that the trade between the two-place and the one-place function is notationally glossed over doesn’t mean that it isn’t being made.

- (2) Versions of type theory deal with two-place functions in a different way, by a type-shifting trick. Addition for example – naively a binary function that just deals in numbers – is traded in for a function of the type $N \rightarrow (N \rightarrow N)$. This is a unary function which takes one number (of type N) and outputs something of a higher type, i.e. a unary *function* (of type $N \rightarrow N$). We then get from two numbers as input to a numerical output in two steps, by feeding the first number to a function which delivers another function as output and then feeding the second number to the second function.

This so-called ‘currying’ trick of course is also perfectly adequate for certain formal purposes. But again a trade is being made. Here’s a revealing quote from *A Gentle Introduction to Haskell* on the haskell.org site (Haskell being one those programming languages where what we might think of naturally as binary functions are curried):

Consider this definition of a function which adds its two arguments:

```
add :: Integer → Integer → Integer
add x y = x + y
```

So we have the declaration of type – we are told that `add` sends a number to a function from numbers to numbers. We are then told how this curried function acts ... but how? By appeal, of course, to our prior understanding of the familiar school-room two-place addition function! The binary function remains a rung on the ladder by which we climb to an understanding of what’s going on in the likes of Haskell (even if we propose to throw away the ladder after we’ve climbed it).

So now back to categories. We don’t have native binary morphisms in category theory. Nor do we get straightforward currying within a category, at least in the sense that we won’t have an arrow inside a category whose target is another *arrow* of that category (though we will meet a reflection of the idea of currying in this chapter). Hence, as we have already seen, then, we need to use a version of the set-theoretic trick. We can in a noncircular way give a categorial treatment of pair-objects as ingredients of products. And with such objects now to hand, an arrow of the kind $f: X \times Y \rightarrow Z$ is indeed available to do duty for a two-place function from an object in X and an object in Y to a value in Z . So this, as already announced, will have to be our implementation device.

13.2 Exponentials defined

- (a) It is standard to use the notation ‘ C^B ’ in set theory to denote the set of functions $f: B \rightarrow C$. But why is the exponential notation apt?

Here is one reason. ‘ C^n ’ is of course natural notation for the n -times Cartesian product of C with itself, i.e. the set of n -tuples of elements from C . But an n -tuple of C -elements can be regarded as equivalent to a function from an indexing

Exponentials

set n , i.e. from the set $\{0, 1, 2, \dots, n-1\}$, to C . Therefore C^n , the set of n -tuples, can indeed be thought of as equivalent to C^n , re-defined as the set of functions $f: n \rightarrow C$. And is then natural to extend this notation to the case where the indexing set B is no longer a number n .

Four more observations, still in informal set-theory:

- (1) For all sets B, C there is a set C^B .
- (2) There is a *two*-place evaluation function $ev(\cdot, \cdot)$ which takes an element $f \in C^B$ and an element $b \in B$, evaluates the first argument f at the selected second argument b , and so returns the value $f(b) \in C$.
- (3) Take any *two*-place function $g(\cdot, \cdot)$ that maps an element of A and an element of B to some value in C : informally notate that binary function $g: A, B \rightarrow C$. Then, fixing an element $a \in A$ determines a derived *one*-place function $g(a, \cdot): B \rightarrow C$.
- (4) So, for any such binary $g: A, B \rightarrow C$ there is a unique associated one-place function, its *exponential transpose* $\bar{g}: A \rightarrow C^B$, which sends $a \in A$ to $g(a, \cdot): B \rightarrow C$. We then have $ev(\bar{g}(a), b) = g(a, b)$.

These elementary observations pretty much tell us how to characterize categorially an ‘exponential object’ C^B in **Set**. We simply need to remember that categorially we regiment two-place functions as arrows from products.

Hence, we can say this. In **Set**, for all B, C , there is an object C^B and an arrow $ev: C^B \times B \rightarrow C$ such that for any arrow $g: A \times B \rightarrow C$, there is a *unique* $\bar{g}: A \rightarrow C^B$ (g ’s exponential transpose) which makes the following diagram commute:

(Exp)

$$\begin{array}{ccc}
 A \times B & & \\
 \bar{g} \times 1_B \downarrow & \searrow g & \\
 C^B \times B & & \xrightarrow{ev} C
 \end{array}$$

The product arrow $\bar{g} \times 1_B$ here, which acts componentwise on pairs in $A \times B$, is defined categorially in §8.4.

(b) Now generalize in the obvious way:

Definition 69. Assume \mathcal{C} is a category with binary products. Then $[C^B, ev]$, with C^B an object and arrow $ev: C^B \times B \rightarrow C$, forms an *exponential* of C by B in \mathcal{C} iff the following holds, with all the mentioned objects and arrows being in \mathcal{C} : for every object A and arrow $g: A \times B \rightarrow C$, there is a unique arrow $\bar{g}: A \rightarrow C^B$ (g ’s transpose) such that $ev \circ \bar{g} \times 1_B = g$, i.e. such that the diagram (Exp) commutes. △

Note that, as with products, the square-bracket notation here is once more just punctuation for readability’s sake. More importantly, note that if we change the

objects B, C the evaluation arrow $ev: C^B \times B \rightarrow C$ changes, since the source and/or target will change. (It might occasionally help to think of the notation ‘ ev ’ as really being lazy shorthand for something like ‘ $ev_{C,B}$ ’.)

Definition 70. A category \mathcal{C} has all exponentials iff for all \mathcal{C} -objects B, C , there is a corresponding exponential $[C^B, ev]$. \triangle

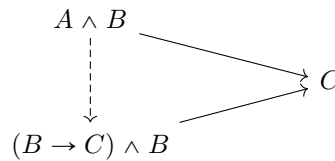
(c) Exponentials in \mathcal{C} aren’t defined in terms of a type of cones (or cocone) in \mathcal{C} . But just as a limit cone over D is defined in terms of every cone over D ‘factoring through’ the limit via a unique arrow, so an exponential of C with B is defined in terms of every arrow from some $A \times B$ to C ‘factoring through’ the exponential via a unique arrow. In short, limits and exponentials alike are defined in terms of every relevant item factoring through via a unique map. That’s why we can speak of both the properties of being a limit and being an exponential as examples of universal mapping properties.

13.3 Examples of exponentials

Let’s immediately give three easy examples of categories which it is easy to see have exponentials:

- (1) Defns. 69 and 70 were purpose-built to ensure that **Set** counts as having all exponentials – a categorial exponential of C by B is provided by the set C^B (in the standard set-theoretic sense) equipped with the set function ev as described before. But we can note now that this construction applies equally in **FinSet**, the category of finite sets, since the set C^B is finite if both B and C are finite, and hence C^B is also in **FinSet**. Therefore **FinSet** has all exponentials.
- (2) In §7.3 (5) we met the category $\mathbf{Prop}_{\mathcal{L}}$ whose objects are wffs of a given first-order language \mathcal{L} , and where there is a unique arrow from A to B iff $A \models B$. Assuming \mathcal{L} has the usual rules for conjunction and implication, then for any B, C , the conditional $B \rightarrow C$ provides an exponential object C^B , with the corresponding evaluation arrow $ev : C^B \times B \rightarrow C$ reflecting the modus ponens entailment $B \rightarrow C, B \models C$.

Why does this work? Recall that products in $\mathbf{Prop}_{\mathcal{L}}$ are conjunctions. And note that, given $A \wedge B \models C$, then by the standard rules $A \models B \rightarrow C$ and hence – given the trivial $B \models B$ – we have $A \wedge B \models (B \rightarrow C) \wedge B$. We therefore get the required commuting diagram of this shape,



Exponentials

where the down arrow is the product of the implication arrow from A to $B \rightarrow C$ and the trivial entailment from B to B .

- (3) Relatedly, take a Boolean algebra $(B, \neg, \wedge, \vee, 0, 1)$, and put $a \leq b =_{\text{def}} (a \wedge b) = a$ for all $a, b \in B$. Then, treated as a partially ordered set with that order, the Boolean algebra corresponds to a poset category, with a unique arrow between a and b when $a \leq b$. In this category, $a \wedge b$, with the only possible projection arrows, is the categorial product of a, b

Such a poset category based on a Boolean algebra has an exponential for each pair of objects, namely (to use a suggestive notation) the object $b \Rightarrow c =_{\text{def}} \neg b \vee c$, together with the evaluation arrow ev the unique arrow corresponding to $(b \Rightarrow c) \wedge b \leq c$.

To check this claim, we need first to show that we have indeed well-defined the evaluation arrow ev for every b, c , i.e. show that we always have $(b \Rightarrow c) \wedge b \leq c$. However, as we want,

$$(\neg b \vee c) \wedge b = (\neg b \wedge b) \vee (c \wedge b) = 0 \vee (c \wedge b) = (c \wedge b) \leq c$$

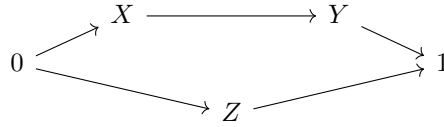
by Boolean rules and the definition of \leq .

Second, we need to verify that the analogous diagram to the last one commutes, which crucially involves showing that if $a \wedge b \leq c$ then $a \wedge b \leq (b \Rightarrow c) \wedge b$. That's more Boolean algebra, which can perhaps be left as a brain-teaser.

So Boolean-algebras-treated-as-poset-categories have all exponentials. Working through the details, however, we find that the required proofs *don't* call on the Boolean principle $\neg\neg a = a$, so the claim about Boolean algebras can be strengthened to the claim that Heyting-algebras-treated-as-poset-categories have all exponentials (where a Heyting algebra is, in effect, what you get when you drop the 'double negation' rule from the Boolean case: we will return later to talk about this important case from logic).

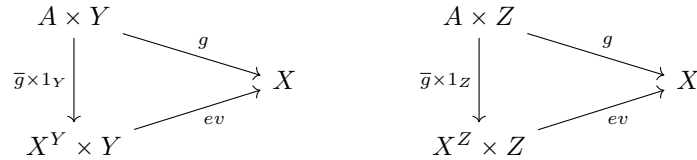
Now these first examples are of categories which have *all* exponentials. But of course, a category may lack exponentials entirely (for example, take a poset category with no products). Or it may have just trivial exponentials (we'll see in the next section that, if a category has a terminal object 1 , then it will automatically have at least the trivial exponentials X^1 and 1^X). And as we'll now see, it can also be the case that a category has *some* non-trivial exponentials, though not *all* exponentials.

- (4) For an initial toy example, we might consider the poset category arising from a five-element non-distributive lattice, which has the following arrows (plus the necessary identity arrows and composites):



In this category, X^Y doesn't exist, but $X^Z = Y$. It is perhaps a useful reality check to pause to show this:

Proof. Consider these two putative diagrams as imagined instances of (Exp):



Suppose there is an exponential object X^Y . Then for every arrow $g: A \times Y \rightarrow X$ there must exist a unique $\bar{g}: A \rightarrow X^Y$ making the left-hand diagram commute. Since $Z \times Y = 0$, there is indeed an arrow $g_1: Z \times Y \rightarrow X$; and since $X \times Y = X$ there is an arrow $g_2: X \times Y \rightarrow X$. Therefore we need arrows $\bar{g}_1: Z \rightarrow X^Y$ and $\bar{g}_2: X \rightarrow X^Y$, which implies $X^Y = 1$. But $X^Y \times Y = Y$, and hence there is no possible arrow $ev: X^Y \times Y \rightarrow X$. Hence there is no exponential object X^Y , and the left-hand diagram is a mirage!

Now put $X^Z = Y$, with the arrow ev the sole arrow from 0 to X . Then it is easily checked that for each arrow $g: A \times Z \rightarrow X$ (that requires $A = 0, X,$ or Y) there is a corresponding unique $\bar{g}: A \rightarrow Y$ making the diagram on the right commute. Just remember we are in a poset category so arrows with the same source and target are equal. \square

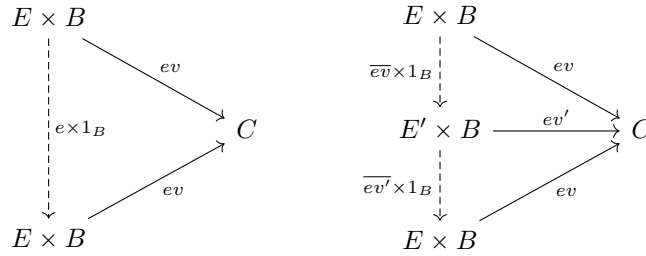
- (5) Consider next **Count**, the category of sets which are no larger than countably infinite, and of set-functions between them. If the **Count**-objects B and C are in fact finite sets, then there is another finite set C^B which, with the obvious function ev , will serve as an exponential. But if B is a countably infinite set, and C has at least two members, then the set C^B is uncountable, so won't be available to be an exponential in **Count** – and evidently, nothing smaller will so.
- (6) The standard example, however, of an interesting category which has some but not all exponentials is **Top**. If X is a space living in **Top**, then it is 'exponentiable', meaning that Y^X exists for all Y , if and only if it is so-called *core-compact* – and not all spaces are core-compact. It would, however, take us far too far afield to explain and justify this example.

13.4 Exponentials are unique

(a) Defn. 69 talks of ‘an’ exponential of C with B . But exponentials – as we might expect by now, given that the definition is by a universal mapping property – are in fact unique, at least up to unique isomorphism:

Theorem 66. *In a category \mathcal{C} with exponentiation, if given objects B, C have exponentials $[E, ev]$ and $[E', ev']$, then there is a unique isomorphism between E and E' compatible with the evaluation arrows.*

Proof. Two commuting diagrams encapsulate the core of the argument, which parallels the proof of Theorem 26:



By definition $[E, ev]$ is an exponential of C by B iff there is a unique mediating arrow $e: E \rightarrow E$ such that $ev \circ e \times 1_B = ev$. But as the diagram on the left reminds us, 1_E will serve as a mediating arrow. Hence $e = 1_E$.

The diagram on the right then reminds us that $[E, ev]$ and $[E', ev']$ factor uniquely through each other, and putting the two commuting triangles together, we get

$$ev \circ (\overline{ev'} \times 1_B) \circ (\overline{ev} \times 1_B) = ev.$$

Applying Theorem 37, we know that $(\overline{ev'} \times 1_B) \circ (\overline{ev} \times 1_B) = (\overline{ev'} \circ \overline{ev}) \times 1_B$, and hence

$$ev \circ (\overline{ev'} \circ \overline{ev}) \times 1_B = ev.$$

And now applying the uniqueness result from the first diagram

$$\overline{ev'} \circ \overline{ev} = 1_E.$$

Similarly, by interchanging E and E' in the second diagram, we get

$$\overline{ev} \circ \overline{ev'} = 1_{E'}.$$

Whence $\overline{ev}: E \rightarrow E'$ is an isomorphism. □

(b) When we were talking about e.g. products and equalizers, we gave two types of proof for their uniqueness (up to unique isomorphism). One was a direct proof from the definitions. For the other proof, we noted that products are terminal objects in a category of wedges, equalizers terminal objects in a category of forks, and then appealed to the uniqueness of terminal objects.

13.5 Further results about exponentials

We have now given a proof of the first type, a direct proof, of the uniqueness of exponentials. Can we give a proof of the second type? Well, consider:

Definition 71. Given objects B and C in the category \mathcal{C} , then the category $\mathcal{C}_{E(B,C)}$ of parametrized maps from B to C has the following data:

1. Objects $[A, g]$ comprising a \mathcal{C} -object A , and a \mathcal{C} -arrow $g: A \times B \rightarrow C$,
2. An arrow from $[A, g]$ to $[A', g']$ is any arrow \mathcal{C} -arrow $h: A \rightarrow A'$ which makes the following diagram commute:

$$\begin{array}{ccc}
 A \times B & \xrightarrow{g} & C \\
 \downarrow h \times 1_B & & \nearrow g' \\
 A' \times B & &
 \end{array}$$

The identity arrows and composition are as in \mathcal{C} . △

It is easily checked that this indeed defines a category, and then we evidently have

Theorem 67. *An exponential $[C^B, ev]$ is a terminal object in the category $\mathcal{C}_{E(B,C)}$.*

Since exponentials are terminal in a suitable category that yields the second type of proof of their uniqueness.

So in summary the situation is this. Exponentials in \mathcal{C} are *not* a type of limit in \mathcal{C} as characterized in Defn. 56 (for that definition talks of limit cones over diagrams in that same category, and an exponential isn't such a thing). But exponentials *can* be thought of as limits in *another*, derived, category of the kind $\mathcal{C}_{E(B,C)}$.

13.5 Further results about exponentials

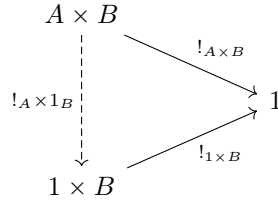
(a) We now show, as promised, that any category with a terminal object has at least trivial exponentials as follows:

Theorem 68. *If the category \mathcal{C} has a terminal object 1 , then for any \mathcal{C} -object B, C , we have (1) $1^B \cong 1$ and (2) $C^1 \cong C$.*

Perhaps we should put that more carefully. The claim (1) is that if there is a terminal object 1 then there exists an exponential $[1^B, ev]$; and for any such exponential object 1^B , $1^B \cong 1$. Similarly for (2).

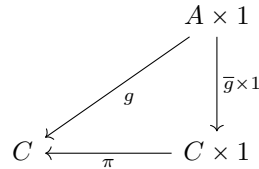
Proof for (1). Using, as before, $!_X$ for the unique arrow from X to the terminal object 1 , consider the following diagram:

Exponentials



This has to commute, whatever A is (because there is only one arrow from $A \times B$ to a terminal object). Since there is only one possible arrow from A to 1 , this means that $[1, \text{!}_{1 \times B}]$ can serve as an exponential for 1 by B . Hence there exists an exponential 1^B , and by the uniqueness theorem, for any such exponential object 1^B , $1^B \cong 1$. \square

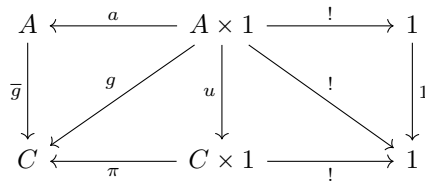
Proof for (2). Here's the natural proof-strategy. Suppose we are given an arrow $g: A \times 1 \rightarrow C$. Show that there is always a unique \bar{g} making this commute,



where π is the projection from the product. Then $[C, \pi]$ serves as an exponential of C by 1 and hence, by the uniqueness theorem, any $C^1 \cong C$.

But there's an isomorphism a' which sends A to $A \times 1$ (the inverse of the projection from the product); so put $\bar{g} = g \circ a'$, and then the diagram will commute. And that's the unique possibility, so we are done. \square

If it isn't obvious why our definition of \bar{g} does the trick in the last proof, perhaps we should expand the argument. So: the wedge $C \xleftarrow{g} A \times 1 \xrightarrow{\text{!}} 1$ must factor through the product wedge $C \xleftarrow{\pi} C \times 1 \xrightarrow{\text{!}} 1$ via a unique mediating u , making the lower triangles in the following diagram commute:



Complete the diagram with the product wedge $A \xleftarrow{a} A \times 1 \xrightarrow{\text{!}} 1$ as shown, and – recalling that a and π must be isomorphisms by Theorem 29 – put $\bar{g} = g \circ a'$ where a' is the inverse of a . Then the whole diagram commutes.

This means that $u = \bar{g} \times 1$ by definition of the operation \times on arrows in §8.4. Hence for each $g: A \times 1 \rightarrow C$ there is indeed a corresponding \bar{g} making our first diagram commute. Moreover \bar{g} is unique. If $k \times 1$ makes the second

diagram commute then (i) it must equal u , and so $k \times 1 = \pi^{-1} \circ g$, but also by its definition, $\pi \circ k \times 1 = k \circ a$. Hence $g = k \circ a$, so $k = g \circ a' = \bar{g}$.

(b) We next need to establish a crucial general result:

Theorem 69. *If there exists an exponential of C by B in the category \mathcal{C} , then, for any object A in the category, there is a one-one correlation between arrows $A \times B \rightarrow C$ and arrows $A \rightarrow C^B$.*

There is also a one-one correlation between arrows $A \rightarrow C^B$ and arrows $B \rightarrow C^A$.

Proof. By definition of the exponential $[C^B, ev]$, an arrow $g: A \times B \rightarrow C$ is associated with a unique ‘transpose’ $\bar{g}: A \rightarrow C^B$ making the diagram (Exp) commute.

The map $g \mapsto \bar{g}$ is injective. For suppose $\bar{g} = \bar{h}$. Then $g = ev \circ (\bar{g} \times 1_B) = ev \circ (\bar{h} \times 1_B) = h$.

The map $g \mapsto \bar{g}$ is also surjective. Take any $k: A \rightarrow C^B$; then if we put $g = ev \circ (k \times 1_B)$, \bar{g} is the unique map such that $ev \circ (\bar{g} \times 1_B) = g$, so $k = \bar{g}$.

Hence $g \mapsto \bar{g}$ is the required bijection between arrows $A \times B \rightarrow C$ and arrows $A \rightarrow C^B$, giving us the first part of the theorem.

For the second part, we just note that arrows $A \times B \rightarrow C$ are in one-one correspondence with arrows $B \times A \rightarrow C$, in virtue of the isomorphism between $A \times B$ and $B \times A$ (see Theorems 17 and 27). We then apply the first part of the theorem. \square

This last theorem gives us a categorial analogue of the idea of currying that we met in §13.1, where a two-place function of type $A, B \rightarrow C$ gets traded in for a one-place function of type $A \rightarrow (B \rightarrow C)$.

13.6 Cartesian closed categories

Categories like **Set**, **Prop** and **Bool** which have all exponentials (which presupposes having binary products) and which also have a terminal object (and hence all finite products) are important enough to deserve a standard label:

Definition 72. A category \mathcal{C} is a *Cartesian closed category* iff it has all finite products and all exponentials.¹ \triangle

Such categories have nice properties meaning that exponentials there indeed behave as exponentials ‘ought’ to behave. For a start:

Theorem 70. *If \mathcal{C} is a Cartesian closed category, then for all $A, B, C \in \mathcal{C}$*

¹Terminological aside: some call a category with all finite products a *Cartesian category* – but this term is also used in other ways so is probably best avoided. By contrast, the notion of a *Cartesian closed category* has a settled usage.

Exponentials

- (1) If $B \cong C$, then $A^B \cong A^C$,
- (2) $(A^B)^C \cong A^{B \times C}$,
- (3) $(A \times B)^C \cong A^C \times B^C$.

Proof of (1). Here's the basic idea for a brute force proof. We know that there exists an arrow $ev: A^B \times B \rightarrow A$. Since $B \cong C$, there is a derived arrow $g: A^B \times C \rightarrow A$. This has a unique associated transpose, $\bar{g}: A^B \rightarrow A^C$. Similarly, there is an arrow $\bar{h}: A^C \rightarrow A^B$. It remains to confirm that these arrows are (as you'd expect) inverses of each other, whence $A^B \cong A^C$.

To spell that out, consider the following diagram (where $j: B \rightarrow C$ is an isomorphism witnessing that $B \cong C$):

$$\begin{array}{ccc}
 A^B \times B & & \\
 \downarrow 1 \times j & \searrow ev & \\
 A^B \times C & & \\
 \downarrow \bar{g} \times 1 & \searrow g & \\
 A^C \times C & \xrightarrow{ev'} & A \\
 \downarrow 1 \times j^{-1} & \searrow h & \\
 A^C \times B & & \\
 \downarrow \bar{h} \times 1 & \searrow ev & \\
 A^B \times B & &
 \end{array}$$

Here we've omitted subscripts on labels for identity arrows to reduce clutter. It is easy to see that since 1 and j are isomorphisms, so is $1 \times j$, and then if we put $g = ev \circ (1 \times j)^{-1}$ the top triangle commutes. The next triangle commutes by definition of the transpose \bar{g} ; the third commutes if we now put $h = ev' \circ (1 \times j^{-1})^{-1}$; and the bottom triangle commutes by the definition of the transpose \bar{h} .

Products of arrows compose componentwise, as shown in Theorem 37. Hence the composite vertical arrow reduces to $(\bar{h} \circ \bar{g}) \times 1$. However, by the definition of the exponential $[A^B, ev]$ we know that there is a unique mediating arrow, k such that this commutes:

$$\begin{array}{ccc}
 A^B \times B & & \\
 \downarrow k \times 1 & \searrow ev & \\
 A^B \times B & & A \\
 & \searrow ev & \\
 & & A
 \end{array}$$

We now have two candidates for k which make the diagram commute, the identity arrow and $\bar{h} \circ \bar{g}$. Hence by uniqueness, $\bar{h} \circ \bar{g} = 1$.

A similar argument shows that $\bar{g} \circ \bar{h}$. We are therefore done. \square

Proofs of (2) and (3). We can give a similarly direct proof of (2), along the following lines. Start with the evaluation arrow $ev: A^{B \times C} \times (B \times C) \rightarrow A$. We can shuffle terms in the product to derive an arrow $(A^{B \times C} \times C) \times B \rightarrow A$. Transpose this once to get an arrow $A^{B \times C} \times C \rightarrow A^B$ and transpose again to get an arrow $A^{B \times C} \rightarrow (A^B)^C$. Then similarly find an arrow from $(A^B)^C \rightarrow A^{B \times C}$, and show the two arrows are inverses of each other.

We can, however, leave it as an exercise for enthusiasts to work out details here. That's because we will eventually be able to bring to bear some heavier-duty general apparatus which will yield fast-track proofs of (2) and (3), and indeed of (1) again. \square

Theorem 71. *If \mathcal{C} is a Cartesian closed category with terminal object 1, then for all $A, B, C \in \mathcal{C}$*

- (1) $1^B \cong 1$,
- (2) $C^1 \cong C$,

And if \mathcal{C} also has an initial object 0, then

- (3) $A \times 0 \cong 0 \cong 0 \times A$,
- (4) $A^0 \cong 1$,
- (5) *if there is an arrow $A \rightarrow 0$, then $A \cong 0$,*
- (6) *there exists an arrow $1 \rightarrow 0$ iff \mathcal{C} is category whose objects are all isomorphic to each other.*

The first two results are just particular cases of Theorem 68. But it is worth noting that if we are assuming we are working in a Cartesian closed category, and hence assuming that 1^B exists, then we can instead use this slick argument:

Proof of (1). By the Theorem 69, for each A , there is a one-one correlation between arrows $A \rightarrow 1^B$ and arrows $A \times B \rightarrow 1$. But since 1 is terminal, there is exactly one arrow $A \times B \rightarrow 1$; hence, for each A , there is exactly one arrow $A \rightarrow 1^B$. Therefore 1^B is terminal, and hence $1^B \cong 1$. \square

Proof of (3). Since $A \times 0$ and $0 \times A$ exist by hypothesis, and are isomorphic by Theorem 27 (2), we need only prove $0 \times A \cong 0$.

By Theorem 69, for all C , there is a one-one correspondence between arrows $0 \rightarrow C^A$ and arrows $0 \times A \rightarrow C$. But 0 is initial, so there is exactly one arrow $0 \rightarrow C^A$. Hence for all C there is exactly one arrow $0 \times A \rightarrow C$, making $0 \times A$ initial too. Whence $0 \times A \cong 0$. \square

Exponentials

Proof of (4). By Theorem 69 again, for all C , there is a bijection between arrows $C \rightarrow A^0$ and arrows $C \times 0 \rightarrow A$. And by (3) and Theorem 17 there is a bijection between arrows $C \times 0 \rightarrow A$ and arrows $0 \rightarrow A$. Since 0 is initial there is exactly one arrow $0 \rightarrow A$, and hence for all C there is exactly one arrow $C \rightarrow A^0$, so A^0 is terminal and $A^0 \cong 1$. \square

Proof of (5). By assumption, there exists a wedge $A \xleftarrow{1_A} A \xrightarrow{f} 0$, and this will factor uniquely through the product $A \times 0$, as in

$$\begin{array}{ccccc}
 & & A & & \\
 & \nearrow^{1_A} & \downarrow \langle 1_A, f \rangle & \searrow^f & \\
 A & \xleftarrow{\pi_1} & A \times 0 & \xrightarrow{\pi_2} & 0
 \end{array}$$

So $\pi_1 \circ \langle 1_A, f \rangle = 1_A$. But $A \times 0 \cong 0$, so $A \times 0$ is an initial object, so there is a unique arrow $A \times 0 \rightarrow A \times 0$, namely $1_{A \times 0}$. Hence (travelling round the left triangle) $\langle 1_A, f \rangle \circ \pi_1 = 1_{A \times 0}$. Therefore $\langle 1_A, f \rangle: A \rightarrow A \times 0$ has a two-sided inverse. Whence $A \cong A \times 0 \cong 0$. \square

Proof of (6). One direction is trivial. For the other, suppose there is an arrow $f: 1 \rightarrow 0$. Then, for any A there must be a composite arrow $A \longrightarrow 1 \xrightarrow{f} 0$, hence by (5), $A \cong 0$. So every object in the category is isomorphic. \square

Here's a quick application of the result (6), that in a Cartesian closed category with an arrow $1 \rightarrow 0$, all objects are isomorphic:

Theorem 72. *The category \mathbf{Grp} is not Cartesian closed.*

Proof. The one-element group is both initial and terminal in \mathbf{Grp} , so here $1 \cong 0$, and hence there is an arrow $1 \rightarrow 0$ in \mathbf{Grp} . But trivially, not all groups are isomorphic! Therefore the category \mathbf{Grp} cannot be Cartesian closed. \square

14 Group objects, natural number objects

We have seen how to define categorially a variety of familiar constructions using universal mapping properties; in particular, we have defined products and exponentials (to mention just the two cases which will feature again most often in this chapter).

We will next see how to use the apparatus that we now have available to characterize two familiar kinds of mathematical structure in categorial terms. We first give a definition of so-called *group objects* living in categories, and explore these just a little. Then we turn to say something equally introductory about that most basic of structures, *the natural numbers*. We won't take these discussions very far for the moment: our aim here in each case is simply to illustrate how we can begin to explore types of well-known mathematical structures from inside category theory.

14.1 Groups in Set

We informally think of a group as a collection of objects equipped with a binary operation of group 'multiplication' and with a designated element which is an identity for the operation. The group operation is associative, and every element has a two-sided inverse.

So how can we characterize such a structure as living in the category **Set**? We need an object G to provide a collection of group-elements, and we need three arrows (which are functions in this category):

- (i) $m: G \times G \rightarrow G$ (here, once again, we have to trade the informal two-place operation of 'multiplication' for an arrow from a corresponding single source, i.e. from a product);
- (ii) $e: 1 \rightarrow G$ (this element-as-arrow from a terminal object picks out a particular group-element in G – we'll also call this distinguished member of the group ' e ', allowing context to disambiguate);
- (iii) $i: G \rightarrow G$ (this is the arrow which sends a group-element to its inverse).

We then need to impose constraints on these arrows corresponding to the usual group axioms:

Group objects, natural number objects

- (1) We require the group operation m to be associative. Categorially, consider the following diagram:

$$(G1) \quad \begin{array}{ccc} (G \times G) \times G & \xrightarrow{\cong} & G \times (G \times G) \\ \downarrow m \times 1_G & & \downarrow 1_G \times m \\ G \times G & \xrightarrow{m} & G \xleftarrow{m} G \times G \end{array}$$

Here the arrow at the top represents the naturally arising isomorphism between the two triple products that is established by the proof of Theorem 27 (3) in §8.5.

Remembering that we are working in \mathbf{Set} , take an element $((j, k), l) \in (G \times G) \times G$. Going round on the left, that gets sent to $(m(j, k), l)$ and then to $m(m(j, k), l)$. Going round the other direction we get to $m(j, m(k, l))$. So requiring the diagram to commute captures the associativity of m .

- (2) Informally, we next require e to act like a multiplicative identity.

To characterize this condition categorially, start by defining the map $e!: G \rightarrow G$ by composing $G \xrightarrow{!} 1 \xrightarrow{e} G$. In \mathbf{Set} we can think of $e!$ as the function which sends anything in the G to its designated identity element e . We then have the following product diagram:

$$\begin{array}{ccccc} & & G & & \\ & \swarrow 1_G & \downarrow \langle 1_G, e! \rangle & \searrow e! & \\ G & \xleftarrow{\pi_1} & G \times G & \xrightarrow{\pi_2} & G \end{array}$$

So we can think of the mediating arrow $\langle 1_G, e! \rangle$ as sending an element $g \in G$ to the pair (g, e) .

The element e then behaves like a multiplicative identity on the right if m sends this pair (g, e) in turn back to g – i.e. if the top triangle in the following diagram commutes:

$$(G2) \quad \begin{array}{ccc} G & \xrightarrow{\langle 1_G, e! \rangle} & G \times G \\ \downarrow \langle e!, 1_G \rangle & \searrow 1_G & \downarrow m \\ G \times G & \xrightarrow{m} & G \end{array}$$

Similarly the lower triangle commutes just if e behaves as an identity on the left. So, for e to behave as a two-sided identity element, it is enough that the whole diagram commutes.

- (3) Finally, we informally require that every element $g \in G$ has an inverse g^{-1} or $i(g)$ such that $m(g, i(g)) = e = m(i(g), g)$. Categorially, we can express this by requiring that the following commutes:

$$(G3) \quad \begin{array}{ccccc} G \times G & \xleftarrow{\delta_G} & G & \xrightarrow{\delta_G} & G \times G \\ \downarrow 1_G \times i & & \downarrow e! & & \downarrow i \times 1_G \\ G \times G & \xrightarrow{m} & G & \xleftarrow{m} & G \times G \end{array}$$

For take an element $g \in G$. Going left, the diagonal arrow δ_G (from Defn. 42) maps it to the pair (g, g) , which is mapped in turn by $1_G \times i$ to $(g, i(g))$ and then by m to $m(g, i(g))$. The central vertical arrow meanwhile simply sends g to e . Therefore, the requirement that the left square commutes tells us, as we want, that $m(g, i(g)) = e$. Similarly the requirement that the right square commutes tells us that $m(i(g), g) = e$.

In summary then, the informal group axioms correspond to the commutativity of our last three diagrams.

But note immediately that this categorial treatment of groups only requires that we are working in a category with binary products and a terminal object. So it is natural to generalize, as follows:

Definition 73. Suppose \mathcal{C} is a category which has binary products and a terminal object. Let G be a \mathcal{C} -object, and $m: G \times G \rightarrow G$, $e: 1 \rightarrow G$ and $i: G \rightarrow G$ be \mathcal{C} -arrows. Then $[G, m, e, i]$ is a *group-object* in \mathcal{C} iff the three diagrams (G1), (G2), (G3) commute, where $e!$ in the latter two diagrams is the composite map $G \xrightarrow{!} 1 \xrightarrow{e} G$. △

Here, ‘group object’ is the standard terminology (though some alternatively say ‘internal group’).

Then, if we don’t fuss about the type-difference between an arrow $e: 1 \rightarrow G$ (in a group object) and a designated element e (in a group), we have established the summary result

Theorem 73. *In the category \mathbf{Set} , a group object is a group.*

And conversely, every group – or to be really pernickety, every group which hasn’t got too many elements to form a set – can be regarded as a group object in \mathbf{Set} .

14.2 Groups in other categories

- (a) Here are just a few more examples of group objects:

Theorem 74. (1) *In the category \mathbf{Top} , which comprises topological spaces with continuous maps between them, a group object is a topological group in the standard sense.*

(2) *In the category \mathbf{Man} , which comprises smooth manifolds with smooth maps between them, a group object is a Lie group.*

(3) *In the category \mathbf{Grp} , a group object is an abelian group.*

The proofs of the first two claims are predictably straightforward if you know the usual definitions of topological groups and Lie groups, and we will not pause over them here. The third claim, by contrast, is probably unexpected. However, the proof is relatively straightforward, quite cute, and a rather useful reality-check:

Proof of (3). Suppose $[G, m, e, i]$ is a group-object in \mathbf{Grp} . Then the object G is already a group, i.e. a set of objects \dot{G} equipped with a group operation and an identity element. We'll use ordinary multiplication notation for that operation, as in ' $x \cdot y$ ', and we'll dub the identity ' $\dot{1}$ ' (so the innards of the group G are notated with dots!). The arrow $e: 1 \rightarrow G$ in the group object must also pick out a distinguished element of \dot{G} , call it ' $\underline{1}$ ', an identity for m .

Now, each arrow in the group-object $[G, m, e, i]$ lives in \mathbf{Grp} , so is a group homomorphism. That means in particular m is a homomorphism from $G \times G$ (the product group, with group operation \times) to G . So take the elements $x, y, z, w \in \dot{G}$. Then,

$$m(x \cdot z, y \cdot w) = m((x, y) \times (z, w)) = m(x, y) \cdot m(z, w)$$

The first equation holds because of how the operation \times is defined for the product group; the second equation holds because m is a homomorphism.

For vividness, let's rewrite $m(x, y)$ as $x \star y$ (so $\underline{1}$ is the unit for \star). Then we have established the interchange law

$$(x \cdot z) \star (y \cdot w) = (x \star y) \cdot (z \star w).$$

We will now use this law twice over (the proof from this point on uses what is standardly called the Eckmann–Hilton argument, a general principle applying when we have such an interchange law between two binary operations with units). First, we have

$$\dot{1} = \dot{1} \cdot \dot{1} = (\underline{1} \star \dot{1}) \cdot (\dot{1} \star \underline{1}) = (\underline{1} \cdot \dot{1}) \star (\dot{1} \cdot \underline{1}) = \underline{1} \star \underline{1} = \underline{1}$$

We can therefore just write 1 for the shared unit, and show secondly that

$$\begin{aligned} x \cdot y &= (x \star 1) \cdot (1 \star y) = (x \cdot 1) \star (1 \cdot y) = x \star y \\ &= (1 \cdot x) \star (y \cdot 1) = (1 \star y) \cdot (x \star 1) = y \cdot x. \end{aligned}$$

We have shown, then, that if $[G, m, e, i]$ is a group object in \mathbf{Grp} , G 's own group operation commutes, and m is the same operation so that must also commute. Therefore the group object is indeed an abelian group. \square

A similar argument, we might note, proves the reverse result: any abelian group can be regarded as a group object in \mathbf{Grp} .

14.3 A very little more on groups

(a) We can continue the story, defining further group-theoretic notions in categorical terms.

- (1) For a start, we can categorially define the idea of a homomorphism between group objects in a category.

Suppose $[G, m, e, i]$ and $[G', m', e', i']$ are group objects in Set . Then a homomorphism between them is a \mathcal{C} -arrow $h: G \rightarrow G'$ which ‘preserves structure’ by appropriately commuting with the group objects’ arrows. More precisely, a moment’s reflection shows that h is a homomorphism just if the following three diagrams commute:

$$\begin{array}{ccc}
 G \times G & \xrightarrow{h \times h} & G' \times G' \\
 m \downarrow & & \downarrow m' \\
 G & \xrightarrow{h} & G'
 \end{array}
 \qquad
 \begin{array}{ccc}
 & 1 & \\
 e \swarrow & & \searrow e' \\
 G & \xrightarrow{h} & G'
 \end{array}
 \qquad
 \begin{array}{ccc}
 G & \xrightarrow{h} & G' \\
 i \downarrow & & \downarrow i' \\
 G & \xrightarrow{h} & G'
 \end{array}$$

- (2) Recall another group-theoretic idea, the key notion of the action of a group G on a set X . Informally, a (left) action is a two-place function $a: G, X \rightarrow X$ such that $a(e, x) = x$ where e is the group identity and $x \in X$, and $a(g \cdot h, x) = a(g, a(h, x))$ for any group elements g, h . This isn’t the place to review the importance of the idea of a group action! Rather, we just note that we can categorially define e.g. the action of a group object $[G, m, e, i]$ on a set X in Set as an arrow $a: G \times X \rightarrow X$ which makes the following two diagrams commute:

$$\begin{array}{ccc}
 1 \times X & \xrightarrow{e \times 1_X} & G \times X \\
 \cong \searrow & & \downarrow a \\
 & & X
 \end{array}
 \qquad
 \begin{array}{ccc}
 (G \times G) \times X & \xrightarrow{m \times 1} & G \times X \\
 \cong \downarrow & & \searrow a \\
 G \times (G \times X) & \xrightarrow{1 \times a} & G \times X \\
 & & \nearrow a \\
 & & X
 \end{array}$$

And so it goes: along these lines, core group-theoretic ideas can be recast into a categorial framework.

(b) The explorations we have begun here could be continued in various directions. First, for example, we could similarly define other kinds of algebraic objects and their morphisms within categories. Second, noting that we can now define group-objects and group-homomorphisms inside a given category like Set , we could go on to categorially define categories of groups living in other categories. And then, generalizing that second idea, we can define the idea of internal categories. But in either of these directions, things begin to get pretty abstract (and not in a way that is particularly helpful for us at this stage in the proceedings). So in the rest of this chapter, we consider something much more basic and much more ‘concrete’, namely ...

14.4 Natural numbers

Our aim is to categorially characterize what are standardly called *natural number objects*. Like group objects in a category, natural number objects in a category aren't naked objects but rather objects-with-arrows. Which arrows? Intuitively, we need an arrow-as-element to pick out an initial object, a 'zero', and we need an arrow-as-operation which takes an element to its 'successor'. That will at least give us sequences – so we say:

Definition 74. If \mathcal{C} is a category with a terminal object, then $[X, i, f]$ is a *sequence object* in \mathcal{C} if X is a \mathcal{C} -object, and i, f are \mathcal{C} -arrows $i: 1 \rightarrow X$ and $f: X \rightarrow X$. \triangle

If we are working in the category **Set**, for example, the arrow i picks out the initial element of a sequence, call this element i too; and f then generates a sequence $i, f(i), f^2(i), f^3(i), \dots$

However, such a sequence could eventually repeat or cycle round; our task is therefore to categorially characterize the limiting case of sequence objects corresponding to non-repeating sequences $f^n(i)$ which look like the natural numbers (i.e. which are ω -sequences). To do this, we start with another definition:

Definition 75. If \mathcal{C} is a category with a terminal object, then the derived category \mathcal{C}_{Seq} has as objects all of \mathcal{C} 's sequence objects $[X, i, f]$, and an arrow $u: [X, i, f] \rightarrow [Y, j, g]$ is a \mathcal{C} -arrow u which makes the following diagram commute in \mathcal{C} :

$$\begin{array}{ccccc}
 1 & \xrightarrow{i} & X & \xrightarrow{f} & X \\
 & \searrow j & \downarrow u & & \downarrow u \\
 & & Y & \xrightarrow{g} & Y
 \end{array}
 \quad \triangle$$

It is routine to check that this definition is in good order and \mathcal{C}_{Seq} is indeed a category (with \mathcal{C}_{Seq} 's identity arrow on $[X, i, f]$ being \mathcal{C} 's identity arrow on X , and composition in \mathcal{C}_{Seq} being composition in \mathcal{C} .)

Three observations about this:

- (1) Suppose we have such a commuting diagram in **Set**. Then u sends a sequence $i, f(i), f^2(i), f^3(i), \dots$ living in X to the sequence $j, g(j), g^2(j), g^3(j), \dots$ living in Y . And given u is functional, if $g^m(j) \neq g^n(j)$ then $f^m(i) \neq f^n(i)$. In other words, the sequence object $[X, i, f]$ can't be *more* constrained by equations of the form $f^m(i) = f^n(i)$ in the sequence than $[Y, j, g]$ is constrained by similar equations between *its* elements.
- (2) So if \mathbf{Set}_{Seq} has an initial object, call it $[N, 0, s]$, then this will have to be as unconstrained a sequence as possible, governed by no additional equations of the form $s^m(0) = s^n(0)$ (where $m \neq n$), and so never repeating. In other words, this initial object will have to correspond to an ω -sequence.

- (3) Conversely, consider the standard implementation of the natural numbers $\mathbb{N} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \dots\}$ in **Set**, together with the arrow $0: 1 \rightarrow \mathbb{N}$ which sends the object in the singleton to \emptyset , and the arrow $s: \mathbb{N} \rightarrow \mathbb{N}$ which sends a set $n \in \mathbb{N}$ to the set $n \cup \{n\}$. Then $[\mathbb{N}, 0, s]$ evidently form an initial object in \mathcal{C}_{Seq} . Given any other sequence object $[Y, j, g]$ in **Set**, setting u to be the arrow $n \mapsto g^n(j)$ makes the diagram commute, and evidently u is unique.

Which all goes to motivate the following general definition:

Definition 76. If \mathcal{C} is a category with a terminal object, then a *natural number object* in \mathcal{C} is an initial object of the derived category \mathcal{C}_{Seq} .

That is to say (with objects and arrows in \mathcal{C}) a natural number object $[N, 0, s]$ comprises an object N and two arrows $0: 1 \rightarrow N$ and $s: N \rightarrow N$ such that for any object Y and arrows $j: 1 \rightarrow Y$ and $g: Y \rightarrow Y$ there is a *unique* arrow u which makes the following diagram commute:

$$\begin{array}{ccccc}
 1 & \xrightarrow{0} & N & \xrightarrow{s} & N \\
 & \searrow j & \downarrow u & & \downarrow u \\
 & & Y & \xrightarrow{g} & Y
 \end{array}
 \quad \triangle$$

Being initial objects of the derived category \mathcal{C}_{Seq} , it follows that if $[N, 0, s]$ and $[N', 0', s']$ are natural number objects in \mathcal{C} then $N \cong N'$ (and indeed there is a unique isomorphism commuting with the arrows in the obvious way).

14.5 The Peano postulates revisited

(a) Let's pause to recall the informal Peano postulates as presented to budding mathematicians. These postulates tell us that the natural numbers N include a distinguished zero object 0 and come equipped with a successor function s , and are such that:

- (1) 0 is a number;
- (2) If n is a number, so is its successor sn ;
- (3) 0 is not a successor of any number;
- (4) Two numbers n, m with the same successor are equal;
- (5) For any property P of natural numbers, if 0 has P , and if sn has P whenever n does, then P holds for all natural numbers.

Here, we should understand 'property' in the generous sense according to which any arbitrary subset A of numbers defines a property (the property of being a member of A). So we can take (5) as equivalent to

Group objects, natural number objects

(5') For any set A of natural numbers, if $0 \in A$, and if $n \in A \Rightarrow sn \in A$, then $A = N$.

A familiar informal set-theoretic argument now shows that the Peano postulates characterize the structure $N, 0, s$ up to isomorphism. And another familiar argument which we also won't repeat here shows that we can deduce the so-called Recursion Theorem:

For any objects Y , selected object j among Y , and function g with Y as domain and codomain, there is a unique function $u: N \rightarrow Y$ such that $u(0) = j$ and $u(sn) = g(u(n))$.

Or in other words, definition by (simple) primitive recursion well-defines a function.

(b) That last observation tells us, of course, that if we take the arrow $0: 1 \rightarrow N$ to send the member of the singleton to the Peano zero, then the resulting $[N, 0, s]$ is a natural number object in **Set**.

What about the converse? Suppose $[N, 0, s]$ is a natural number object in **Set**. Then identifying the Peano zero with the image of the member of 1 under the arrow $0: 1 \rightarrow N$, we of course get (1) and (2) for free. As we noted before, $[N, 0, s]$ can't both be an initial object in the category of sequence objects and be constrained by equations of the form $s^m(0) = s^n(0)$ where $m \neq n$; and that gives us (3) and (4). Which just leaves the induction principle.

Suppose (i) there is an injection $i: A \rightarrow N$, (ii) $0 \in A$, (iii) $n \in A \Rightarrow sn \in A$. We need to show that $A = N$.

By the third supposition, s sends arguments in A to values in A and hence there is a function $s': A \rightarrow A$ which is the restriction of $s: N \rightarrow N$ to A . So (iii) means the square in

$$\begin{array}{ccccc}
 1 & \xrightarrow{0'} & A & \xrightarrow{s'} & A \\
 & \searrow 0 & \downarrow i & & \downarrow i \\
 & & N & \xrightarrow{s} & N
 \end{array}$$

commutes. While (ii) tells us that there is an arrow $0': 1 \rightarrow A$ which makes the triangle commute. Hence the following diagram commutes for some unique u (the top half by the universal property of the natural number object):

$$\begin{array}{ccccc}
 & & N & \xrightarrow{s} & N \\
 & \nearrow 0 & \vdots u & & \vdots u \\
 1 & \xrightarrow{0'} & A & \xrightarrow{s'} & A \\
 & \searrow 0 & \downarrow i & & \downarrow i \\
 & & N & \xrightarrow{s} & N
 \end{array}$$

Which means that the natural number object $[N, 0, s]$ factors through itself via the mediating arrow $i \circ u$. But trivially, it factors through itself by 1_N and hence, since the mediating arrow is unique, $i \circ u = 1_N$. Therefore i is a left inverse and so by Theorem 11 it is epic. Hence (since we are in **Set**) i is surjective. Which means that $A = N$, as we require.

14.6 More on recursion

(a) We have defined natural number objects in an intuitively appealing categorical way, and shown that at least in **Set** we thereby characterize a structure that satisfies the Peano postulates. So far, so good. But there's work still to be done.

For consider next the following pattern for the recursive definition of a *two*-place function $f: N, N \rightarrow N$ in terms of a couple of given one-place functions $g, h: N \rightarrow N$:

- (1) $f(m, 0) = g(m)$
- (2) $f(m, sn) = h(f(m, n))$.

Here's a very familiar example: if $g(m) = m$ and h is the successor function s again, then our equations give us a recursive definition of addition.

We can call this type of definition a *definition by parameterized recursion*, since there is a parameter m which we hold fixed as we run the recursion on n . And intuitively our equations do indeed well-define a determinate binary function f , given any determinate monadic functions g and h (and we can prove that from the Peano Postulates given enough ambient informal set theory).

Now, to characterize this kind of definition by parameterized recursion in a categorical framework, we will evidently have to replace the two-place function with an arrow f from a product. Suppose then that we are again working in some category \mathcal{C} which has a natural number object $[N, 0, s]$. And now suppose too that (P): given any arrows $g: N \rightarrow N$ and $h: N \rightarrow N$, there is a unique arrow $f: N \times N \rightarrow N$ in \mathcal{C} which makes this diagram commute

$$\begin{array}{ccccc}
 N & \xrightarrow{\langle 1_N, 0! \rangle} & N \times N & \xrightarrow{1_N \times s} & N \times N \\
 & \searrow g & \downarrow f & & \downarrow f \\
 & & N & \xrightarrow{h} & N
 \end{array}$$

where $0!$ is the composite map $N \xrightarrow{!} 1 \xrightarrow{0} N$. Saying the triangle commutes is the categorical equivalent of saying that (1) holds (since $\langle 1_N, 0! \rangle$ sends m to the pair $(m, 0)$ – cf. Theorem 36). And saying the square commutes is the equivalent of saying that (2) holds. Hence if a category \mathcal{C} satisfies condition (P), then in effect parameterised recursion well-defines functions in \mathcal{C} . But it doesn't

Group objects, natural number objects

follow from a category's having a natural number object that it will automatically satisfy (P) as well. In other words, while having a natural number object in a category ensures that definitions by *simple* recursion work there, this does *not* automatically ensure that definitions by *parameterized* recursion are also allowed in \mathcal{C} .

(b) However, we do have the following general result:

Theorem 75. *If \mathcal{C} is a Cartesian closed category with a natural number object $[N, 0, s]$, then given any objects A, C , and arrows $g: A \rightarrow C$ and $h: C \rightarrow C$, then there is a unique f which makes the following diagram commute:*

$$\begin{array}{ccccc}
 A & \xrightarrow{\langle 1_A, 0! \rangle} & A \times N & \xrightarrow{1_A \times s} & A \times N \\
 & \searrow g & \downarrow f & & \downarrow f \\
 & & C & \xrightarrow{h} & C
 \end{array}$$

Our previous diagram of course illustrates the special case where $A = C = N$. So in a Cartesian closed category with a natural number object we certainly can warrant the elementary kind of parameterized recursive definition we met at the beginning of the section. And in particular, since **Set** is Cartesian closed, such definitions will be permitted in **Set**-theoretic arithmetic (as we'd of course expect, having already noted that such an arithmetic will satisfy the full Peano postulates).

To prove our theorem we exploit the associations between arrows $A \times N \rightarrow C$ and arrows $N \times A \rightarrow C$ and between those and arrows $N \rightarrow C^A$ which are available in categories with exponentials. The idea is simple; the details are tiresome:

Proof. We suppose, then, that we working in a category \mathcal{C} which has all exponentials (and hence binary products), which has a natural number object $[N, 0, s]$, and which also has two arrows $g: A \rightarrow C$ and $h: C \rightarrow C$.

By hypothesis, the exponential $[C^A, ev]$ exists. Let i be an isomorphism from $1 \times A$ to A . We now use g and h to define

$$g' = \overline{g \circ i}: 1 \rightarrow C^A, \quad h' = \overline{h \circ ev}: C^A \rightarrow C^A,$$

where, remember, overlining notates exponential transposes. These somewhat mysterious definitions can be explained by two commutative diagrams:

$$\begin{array}{ccc}
 1 \times A & & C^A \times A \\
 \downarrow \overline{g \circ i} \times 1_A & \searrow i & \downarrow \overline{h \circ ev} \times 1_A \\
 & A & C \\
 & \searrow g & \downarrow h \\
 C^A \times A & \xrightarrow{ev} & C \\
 & & \downarrow ev \\
 & & C
 \end{array}$$

By the universal property of \mathcal{C} 's natural number object, we know that there is a unique map u which makes the following commute:

$$\begin{array}{ccccc}
 1 & \xrightarrow{0} & N & \xrightarrow{s} & N \\
 & \searrow^{g'} & \downarrow u & & \downarrow u \\
 & & C^A & \xrightarrow{h'} & C^A
 \end{array}$$

So now the name of the game is to define an arrow $f: A \times N \rightarrow C$ in terms of $u: N \rightarrow C^A$ in such a way that the fact that our last diagram commutes will entail that the diagram in the statement of the theorem commutes.

The obvious way to start is to define an arrow $f^o: N \times A \rightarrow C$ by putting $f^o = ev \circ (u \times 1_A)$ so u is the exponential transpose of f^o . Which doesn't quite give us what we want. But there is an isomorphism $o: A \times N \rightarrow N \times A$, and we can put $f = f^o \circ o$.

We now need to show that (i) $f \circ \langle 1_A, 0! \rangle = g$, and (ii) $f \circ (1_A \times s) = h \circ f$. For (i), note first that the following diagram commutes (we've not labelled all the projection arrows, and compare the proof of Theorem 27 (2)):

$$\begin{array}{ccccc}
 & & A & & \\
 & \swarrow ! & \downarrow i^{-1} & \searrow 1_A & \\
 1 & \longleftarrow & 1 \times A & \longrightarrow & A \\
 \downarrow 0 & & \downarrow 0 \times 1_A & & \downarrow 1_A \\
 N & \longleftarrow & N \times A & \longrightarrow & A \\
 \downarrow 1_N & & \downarrow o^{-1} & & \downarrow 1_A \\
 N & \longleftarrow \pi_2 & A \times N & \longrightarrow \pi_1 & A
 \end{array}$$

So $A \xleftarrow{1_A} A \xrightarrow{0!} N$ factors through the product $A \xleftarrow{\pi_1} A \times N \xrightarrow{\pi_2} N$ via the composite of the vertical arrows. Hence $\langle 1_A, 0! \rangle = o^{-1} \circ (0 \times 1_A) \circ i^{-1}$. Therefore using Theorem 37 we can argue:

$$\begin{aligned}
 f \circ \langle 1_A, 0! \rangle &= ev \circ (u \times 1_A) \circ o \circ o^{-1} \circ (0 \times 1_A) \circ i^{-1} \\
 &= ev \circ (u \times 1_A) \circ (0 \times 1_A) \circ i^{-1} \\
 &= ev \circ ((u \circ 0) \times (1_A \circ 1_A)) \circ i^{-1} \\
 &= ev \circ (g' \times 1_A) \circ i^{-1} \\
 &= ev \circ (\overline{g \circ i} \times 1_A) \circ i^{-1} \\
 &= g \circ i \circ i^{-1} \\
 &= g.
 \end{aligned}$$

Group objects, natural number objects

For (ii), we can appeal to Theorem 35 to show that $o \circ (1_A \times s) = (s \times 1_A) \times o$. Then we can argue:

$$\begin{aligned} f \circ (1_A \times s) &= ev \circ (u \times 1_A) \circ o \circ (1_A \times s) \\ &= ev \circ (u \times 1_A) \circ (s \times 1_A) \circ o \\ &= ev \circ ((u \circ s) \times (1_A \times 1_A)) \circ o \\ &= ev \circ ((h' \circ u) \times (1_A \times 1_A)) \circ o \\ &= ev \circ (h' \times 1_A) \circ (u \times 1_A) \circ o \\ &= ev \circ (\overline{h \circ ev} \times 1_A) \circ (u \times 1_A) \circ o \\ &= h \circ ev \circ (u \times 1_A) \circ o \\ &= h \circ f. \end{aligned}$$

Finally, we need to confirm f 's uniqueness. But perhaps, with all the ingredients to hand, we can leave that as an exercise! \square

Our theorem can now be extended in the same vein to cover not just definitions by recursion that carry along a single parameter but also the most general kind of definitions by primitive recursions. Therefore in a Cartesian closed category with a natural number object we can start doing some serious arithmetic. And this is just the beginning: Cartesian closed categories with extra features turn out to be suitable worlds in which to do great swathes of mathematics. About which a lot more in due course.

15 Functors introduced

We have so far been looking *inside* categories and characterizing various kinds of construction to be found there (products, equalizers, exponentials, and the like, and then even e.g. groups and natural number objects). We have seen the same constructions appearing and reappearing in various guises in different categories. An obvious next task is to develop some apparatus for relating categories by mapping such recurrent constructions from one category to another. After all, the spirit of category theory is to understand objects of a certain kind via the morphisms between them: so, in that spirit, we should surely now seek to understand more about categories by thinking about the maps or morphisms between *them*. The standard term for a structure-preserving map *between* categories is ‘functor’. This chapter introduces such maps.

15.1 Functors defined

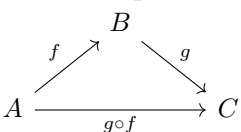
A category \mathcal{C} has two kinds of data, its objects and its arrows. So a functor F from category \mathcal{C} to category \mathcal{D} will need to have two components, one that operates on objects, one that operates on arrows. Hence:

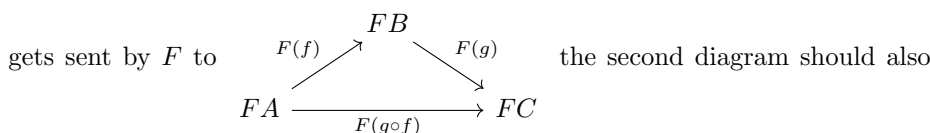
Definition 77. Given categories \mathcal{C} and \mathcal{D} , a *functor* $F: \mathcal{C} \rightarrow \mathcal{D}$ comprises the following data:

- (1) An operation or mapping F_{ob} whose value at the \mathcal{C} -object A is some \mathcal{D} -object we can represent as $F_{ob}(A)$ or, dropping the explicit subscript, as $F(A)$ or indeed simply as FA .
- (2) An operation or mapping F_{arw} whose value at the \mathcal{C} -arrow $f: A \rightarrow B$ is a \mathcal{D} -arrow from $F(A)$ to $F(B)$ which, again dropping the explicit subscript, we can represent as $F(f): F(A) \rightarrow F(B)$, or simply as $Ff: FA \rightarrow FB$.

But there’s more. If a functor is to preserve at least the most basic categorical structure, its component mappings must obey two obvious conditions. First they must map identity arrows to identity arrows. Second they should respect com-

position. That is to say, since the commutative diagram


$$\begin{array}{ccc} & B & \\ f \nearrow & & \searrow g \\ A & \xrightarrow{g \circ f} & C \end{array}$$



commute. Hence we want:

Definition 77 (continued). The data in F must satisfy the following conditions:

Preserving identities: for any \mathcal{C} -object A , $F(1_A) = 1_{FA}$;

Respecting composition: for any \mathcal{C} -arrows f, g such that their composition $g \circ f$ exists, $F(g \circ f) = Fg \circ Ff$. △

These conditions on F are often called, simply, *functoriality*.

15.2 Some elementary examples of functors

Our first example illustrates a broad class of cases:

(F1) There is a functor $F : \text{Mon} \rightarrow \text{Set}$ with the following data:

- i. F_{ob} sends the monoid $(M, \cdot, 1_M)$ to its carrier set M .
- ii. F_{arw} sends $f : (M, \cdot, 1_M) \rightarrow (N, \times, 1_N)$, i.e. a monoid homomorphism acting on elements on M , to the same map thought of as a set-function $\underline{f} : M \rightarrow N$.

So defined, F trivially obeys the axioms for being a functor. All it does is ‘forget’ about the structure carried by the collection of objects in a monoid. It’s a *forgetful functor*, for short.

There are equally forgetful functors from other categories of structured sets to the bare underlying sets. For example, there is the functor $F : \text{Grp} \rightarrow \text{Set}$ that sends groups to their underlying carrier sets and sends group homomorphisms to themselves as set function, forgetting about the group structure. Often, a forgetful functor such as this is called an *underlying* functor (and hence the common practice, which we shall occasionally adopt, of using the letter ‘ U ’ to denote such a functor).

Of course, these forgetful functors are not intrinsically very exciting! It will turn out, however, that they are the boring members of so-called adjoint pairs of functors where they are married to much more interesting companions. But that observation is for later chapters.

To continue just for a moment with the forgetful theme:

(F2) There is a functor $F : \text{Set} \rightarrow \text{Rel}$ which sends sets and triples (domain, graph, codomain) thought of as objects and arrows belonging to Set to the same items thought of as objects and arrows in Rel , forgetting that the arrows are functional.

15.2 Some elementary examples of functors

- (F3) There are also somewhat less forgetful functors, such as the functor from **Rng** to **Grp** that sends a ring to the additive group it contains, forgetting the rest of the ring structure. Or take the functor from **Ab**, the category of abelian groups, to **Grp**, that remembers about group structure but forgets about commutativity.

And now for some different kinds of functors:

- (F4) The powerset functor $P: \mathbf{Set} \rightarrow \mathbf{Set}$ maps a set X to its powerset $\mathcal{P}(X)$ and maps a set-function $f: X \rightarrow Y$ to the function which sends $U \in \mathcal{P}(X)$ to its f -image $f[U] = \{f(x) \mid x \in U\} \in \mathcal{P}(Y)$.
- (F5) Take monoids $(M, \cdot, 1_M)$ and $(N, \times, 1_N)$ and consider the corresponding categories \mathcal{M} and \mathcal{N} in the sense of §3.6.

So \mathcal{M} has a single object $\star_{\mathcal{M}}$, and its arrows are elements of M , where the composition of the arrows m_1 and m_2 is just $m_1 \cdot m_2$, and the identity arrow is the identity element of the monoid, 1_M .

Likewise \mathcal{N} has a single object $\star_{\mathcal{N}}$, and arrows are elements of N , where the composition of the arrows n_1 and n_2 is just $n_1 \times n_2$, and the identity arrow is the identity element of the monoid, 1_N .

So now we see that a functor $F: \mathcal{M} \rightarrow \mathcal{N}$ will need to do the following:

- i. F must send $\star_{\mathcal{M}}$ to $\star_{\mathcal{N}}$.
- ii. F must send the identity arrow 1_M to the identity arrow 1_N .
- iii. F must send $m_1 \circ m_2$ (i.e. $m_1 \cdot m_2$) to $Fm_1 \circ Fm_2$ (i.e. $Fm_1 \times Fm_2$).

Apart from the trivial first condition, that just requires F to be a monoid homomorphism. So any homomorphism between two monoids induces a corresponding functor between the corresponding monoids-as-categories.

- (F6) Take the posets (S, \leq) and (T, \sqsubseteq) considered as categories \mathcal{S} and \mathcal{T} . It is easy to check that a monotone function $f: S \rightarrow T$ (i.e. function such that $s \leq s'$ implies $f(s) \sqsubseteq f(s')$) induces a functor $F: \mathcal{S} \rightarrow \mathcal{T}$ which sends an \mathcal{S} -object s to the \mathcal{T} -object $f(s)$, and sends an \mathcal{S} -arrow, i.e. a pair (s, s') where $s \leq s'$, to the \mathcal{T} -arrow $(f(s), f(s'))$.
- (F7) Next, take the group $G = (G, \cdot, e)$ and now consider it as a category \mathcal{G} – see §5.2(b). Suppose $F: \mathcal{G} \rightarrow \mathbf{Set}$ is a functor. Then F must send \mathcal{G} 's unique object \star to some set X . And F must send a \mathcal{G} -arrow $m: \star \rightarrow \star$ (that's just a member m of G) to a function $F(m): X \rightarrow X$. Functoriality requires that $F(e) = 1_X$ and $F(m \cdot m') = F(m) \circ F(m')$. But those are just the conditions for F to constitute a group action of G on X . Conversely, a group action of G on X amounts to a functor from \mathcal{G} to \mathbf{Set} .
- (F8) There is a list functor $List: \mathbf{Set} \rightarrow \mathbf{Set}$, where $List_{ob}$ sends a set X to $List(X)$, the set of all finite lists or sequences of elements of X , including

the empty one. And $List_{arrow}$ sends a function $f: X \rightarrow Y$ to the function $List(f): List(X) \rightarrow List(Y)$ which sends the list $x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_n$ to $f x_0 \wedge f x_1 \wedge f x_2 \wedge \dots \wedge f x_n$ (where \wedge is concatenation).

Returning to the forgetful theme, we have seen cases of functors that simply forget (some of the) structure put on structured sets. We can also have a functor which obliterates some distinctions between objects or between arrows.

(F9) Suppose \mathcal{S} is a thin, pre-order, category (so has just one arrow between any source and target), and let \mathcal{C} be a fattened category which has the same objects as \mathcal{S} but in addition to the arrows of \mathcal{S} has perhaps extra arrows. Then there will be a functor F from \mathcal{C} back to the slimmed-down \mathcal{S} which takes objects to themselves, and maps every arrow from A to B in \mathcal{C} to the unique such arrow in \mathcal{S} . We could call this F a ‘thinning’ functor.

(F10) A more extreme case: suppose \mathcal{C} and \mathcal{D} are any (non-empty!) categories, and D is any object in \mathcal{D} . Then there is a corresponding collapse-to- D constant functor $\Delta_D: \mathcal{C} \rightarrow \mathcal{D}$ which sends every \mathcal{C} -object to D and every \mathcal{C} -arrow to 1_D .

As a special case, there is a functor $\Delta_0: \mathcal{C} \rightarrow \mathbf{1}$ which sends every object of \mathcal{C} to the sole object of one-object category $\mathbf{1}$, and sends every arrow in \mathcal{C} to the sole arrow of $\mathbf{1}$.

Those last two functors take us from richer categories to more meagre ones. Now for a couple more that go in the other direction again:

(F11) For each object C in \mathcal{C} there is a corresponding functor – overloading notation once more, we can usefully call it $C: \mathbf{1} \rightarrow \mathcal{C}$ – which sends the sole object of $\mathbf{1}$ to C , and sends the sole arrow of $\mathbf{1}$ to 1_C .

(F12) Suppose \mathcal{S} is a subcategory of \mathcal{C} (see §4.2). Then there is an inclusion functor $F: \mathcal{S} \rightarrow \mathcal{C}$ which sends objects and arrows in \mathcal{S} to the same items in \mathcal{C} .

15.3 What do functors preserve and reflect?

Later in this chapter we will look at three more interesting examples of functors. But let’s first make some general points.

A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ sends each \mathcal{C} -object C to its image $F(C)$ and sends each \mathcal{C} -arrow $f: C \rightarrow C'$ to its image $F(f): FC \rightarrow FC'$. These resulting images assemble into an overall image or representation of the category \mathcal{C} living in the category \mathcal{D} . But how good a representation do we get in the general case? What features of \mathcal{C} get carried over by a functor?

15.3 What do functors preserve and reflect?

(a) First a general observation worth highlighting as a theorem as it is easy to go wrong about this:

Theorem 76. *The image of \mathcal{C} in \mathcal{D} assembled by a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ need not be a subcategory of \mathcal{D} .*

Proof. A toy example establishes the point. Let \mathcal{C} be the category we can diagram as

$$A \longrightarrow B_1 \qquad B_2 \longrightarrow C$$

and \mathcal{D} be the category

$$A' \xrightarrow{\quad} B' \xrightarrow{\quad} C'$$

(where we omit the identity arrows). Suppose F_{ob} sends A to A' , both B_1, B_2 to B' , and C to C' ; and let F_{arrow} send identity arrows to identity arrows, and send the arrows $A \rightarrow B_1$ and $B_2 \rightarrow C$ respectively to $A' \rightarrow B'$ and $B' \rightarrow C'$. Trivially F with those components is functorial. But the image of \mathcal{C} under F is not a category (and so not a subcategory of \mathcal{D}), since it contains the arrows $A' \rightarrow B'$ and $B' \rightarrow C'$ but not their composition. \square

(b) We next introduce a pair of standard notions:

Definition 78. Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ and P is some property of arrows. Then

- (1) F *preserves* P iff, for any \mathcal{C} -arrow f , if f has property P , so does $F(f)$.
- (2) F *reflects* P iff, for any \mathcal{C} -arrow f , if $F(f)$ has property P , so does f .

We will say, for short, that F preserves (reflects) X s if F preserves (reflects) the property of being an X . \triangle

One special case gets a special bit of terminology:

Definition 79. A functor F is *conservative* iff it reflects all isomorphisms. \triangle

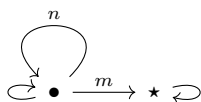
So what properties of arrows get preserved or reflected by functors in general?

Theorem 77. *Functors do not necessarily preserve or reflect monomorphisms and epimorphisms.*

Proof. First, remember 2, the two-object category which we can diagram like this:

$$\curvearrowright \bullet \longrightarrow \star \curvearrowright$$

Trivially, the non-identity arrow m here is monic. And now consider a category \mathcal{C} which adds to 2 another non-identity arrow n :



In \mathcal{C} , we have $m \circ n = m \circ 1_\bullet$ but not $n = 1_\bullet$, so m is not monic in \mathcal{C} . Hence the obvious inclusion functor from $\mathbf{2}$ to \mathcal{C} does not preserve monics.

Now consider the inclusion map $i_M: (\mathbb{N}, +, 0) \rightarrow (\mathbb{Z}, +, 0)$ in \mathbf{Mon} . We saw in §5.3, Ex. (2) that this is epic. But plainly the inclusion map $i_S: \mathbb{N} \rightarrow \mathbb{Z}$ in \mathbf{Set} is not epic (as it isn't surjective). Therefore the forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ maps an epic map (i_M) to a non-epic one (i_S), so does not preserve epics.

For an example of a functor which need not reflect monics or epics, consider a collapse functor which maps \mathcal{C} to $\mathbf{1}$, thereby sending arrows of all sorts to the trivially monic and epic identity arrow on the sole object of $\mathbf{1}$. \square

Theorem 78. *Functors preserve right inverses, left inverses, and isomorphisms. But functors do not necessarily reflect those.*

Proof. We show functors preserve right inverses. Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ is a functor and the arrow $f: C \rightarrow D$ is a right inverse in the category \mathcal{C} . Then for some arrow g , $g \circ f = 1_C$. Hence $F(g \circ f) = F(1_C)$. By functoriality, that implies $F(g) \circ F(f) = 1_{FC}$. So $F(f)$ is a right inverse in the category \mathcal{D} .

Duality gives the result that left inverses are preserved. And putting the two results together shows that isomorphisms are preserved.

For the negative result, just consider again the collapse functor sending \mathcal{C} to $\mathbf{1}$. The only arrow in $\mathbf{1}$, the identity arrow, is trivially an isomorphism (and so a left and right inverse). The \mathcal{C} -arrows sent to it will generally not be. \square

15.4 Faithful, full, and essentially surjective functors

The moral of the previous section is that in general a functor's image of \mathcal{C} inside another category \mathcal{D} may not tell us very much about \mathcal{C} . We are obviously going to be interested, then, in looking at some special kinds of functor which *do* preserve and/or reflect more.

Let's start by defining analogues for the notions of injective and surjective functions. First, as far as their behaviour on arrows is concerned, the useful notions for functors turn out to be these:

Definition 80. A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ is *faithful* iff given any \mathcal{C} -objects C, C' , and any pair of parallel arrows $f, g: C \rightarrow C'$, then if $F(f) = F(g)$, then $f = g$.

F is *full* (that's the standard term) iff given any \mathcal{C} -objects C, C' , then for any arrow $g: FC \rightarrow FC'$ there is an arrow $f: C \rightarrow C'$ such that $g = Ff$.

F is *fully faithful*, some say, iff it is full and faithful. \triangle

Note, a faithful functor needn't be, overall, injective on arrows. For suppose \mathcal{C} is in effect two copies of \mathcal{D} , and F sends each copy faithfully to \mathcal{D} : then F sends

15.4 Faithful, full, and essentially surjective functors

two copies of an arrow to the same image arrow. However, a faithful functor is, for each pair of objects C, C' , injective from the arrows $C \rightarrow C'$ to the arrows $FC \rightarrow FC'$. Likewise, a full functor needn't be, overall, surjective on arrows: but it is locally surjective from the arrows $C \rightarrow C'$ to the arrows $FC \rightarrow FC'$.

Second, in connection with the way functors treat objects, the notion worth highlighting is this:

Definition 81. A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ is *essentially surjective on objects* (e.s.o.) iff for any \mathcal{D} -object D , there is a \mathcal{C} -object C such that $FC \cong D$. \triangle

Plain surjectivity (defined by requiring an object C such that $FC = D$) is less interesting, given that we don't usually care, categorially speaking, whether \mathcal{D} has extra non-identical-but-isomorphic copies of objects. Injectivity on objects (defined in the obvious way by requiring $FC = FC'$ implies $C = C'$, for any \mathcal{C} -objects C and C') is not usually very exciting either.

Some examples:

- (1) The forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ is faithful, as F sends a set-function which happens to be a monoid homomorphism to itself, so different arrows in \mathbf{Mon} get sent to different arrows in \mathbf{Set} . But the functor is not full: there will be lots of arrows in \mathbf{Set} that don't correspond to a monoid homomorphism. Since any set can be trivially made into a monoid, F is essentially surjective on objects.
- (2) The forgetful functor $F: \mathbf{Ab} \rightarrow \mathbf{Grp}$ is faithful, full but not e.s.o.
- (3) The 'thinning' functor from §15.2 (F9), $F: \mathcal{C} \rightarrow \mathcal{S}$, is full but not faithful unless \mathcal{C} is already a pre-order category. But it will be e.s.o.
- (4) Suppose \mathcal{M} and \mathcal{N} are the categories that correspond to the monoids $(M, \cdot, 1_M)$ and $(N, \times, 1_N)$. And let f be a monoid homomorphism between those monoids which is surjective but not injective. Then the functor $F: \mathcal{M} \rightarrow \mathcal{N}$ corresponding to f is full but not faithful.
- (5) You might be tempted to say that the 'total collapse' functor $\Delta_0: \mathbf{Set} \rightarrow \mathbf{1}$ is full but not faithful. But it isn't full. Take C, C' in \mathbf{Set} to be respectively the singleton of the empty set and the empty set. There is a trivial identity map in $\mathbf{1}$, $1: \Delta_0 C \rightarrow \Delta_0 C'$; but there is no arrow in \mathbf{Set} from C to C' .
- (6) An inclusion functor $F: \mathcal{S} \rightarrow \mathcal{C}$ is faithful; if \mathcal{S} is a full subcategory of \mathcal{C} , then the inclusion map is fully faithful, but usually not e.s.o.

How then do faithful or fully faithful functors behave?

Theorem 79. A faithful functor $F: \mathcal{C} \rightarrow \mathcal{D}$ reflects monomorphisms and epimorphisms.

Proof. Suppose Ff is monic, and suppose $f \circ g = f \circ h$. Then $F(f \circ g) = F(f \circ h)$, so by functoriality $Ff \circ Fg = Ff \circ Fh$, and since Ff is monic, $Fg = Fh$. Since F is faithful, $g = h$. Hence f is monic. Dually for epics. \square

Theorem 80. *If a functor is fully faithful it reflects right inverses and left inverses, and hence is conservative.*

Proof. Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ is a fully faithful functor, and let Ff be a right inverse, with f an arrow in \mathcal{C} with source A . Since F is full, Ff must be the right inverse of Fg for some arrow g in \mathcal{C} . So $Fg \circ Ff = 1_{FA}$, whence $F(g \circ f) = 1_{FA} = F(1_A)$. Since F is faithful, it follows that $g \circ f = 1_A$, and f is a right inverse.

Dually, F reflects left inverses, and combining the two results shows that F reflects isomorphisms, i.e. is conservative. \square

Note, however, that the reverse of the last result is not true. A functor can reflect isomorphisms without being fully faithful. Example: consider the forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$. This is faithful but not full. But it is conservative because if the set function Ff is an isomorphism, so is the monoid homomorphism f – for a monoid homomorphism is an isomorphism if and only if its underlying function is.

15.5 A functor from Set to Mon

(a) For this and the next two sections we step back again from generalities to look at three more particular examples of functors. First, we define a functor going in the reverse direction to the forgetful functor in (F1), i.e. we construct a functor $F: \mathbf{Set} \rightarrow \mathbf{Mon}$.

There are trivial ways of doing this. For example just pick a monoid, any monoid, call it \mathcal{M} . Then there is a boring constant functor we could call $!_{\mathcal{M}}: \mathbf{Set} \rightarrow \mathbf{Mon}$ which sends every set X to \mathcal{M} and sends every set-function $f: X \rightarrow Y$ to the identity arrow $1_{\mathcal{M}}: \mathcal{M} \rightarrow \mathcal{M}$ (the identity homomorphism).

But it is instructive to try to come up with something more interesting. So, consider again how we might send sets to monoids, but this time *making as few assumptions as we possibly can* about the monoid that a given set gets mapped to.

Start then with a set S . Since we are making no more assumptions than we need to, we'll have to take the objects in S as providing us with an initial supply of objects for building our monoid, the monoid's *generators*. We now need to equip our incipient monoid with a two-place associative function $*$. But we are assuming as little as we can about $*$ too, so we don't even yet know that applying it keeps us inside the original set of generators S . So S will need to be expanded to a set M that contains not only the original members of S , e.g. x, y, z, \dots , but also all the possible 'products', i.e. everything like $x * x, x * y, y * x, y * z, x * y * x, x * y * x * z, x * x * y * y * z \dots$, etc., etc. – we know, however, that since $*$ is associative, we needn't distinguish between e.g. $x * (y * z)$ and $(x * y) * z$.

But even taking all those products is not enough, for (in our assumption-free state) we don't know whether any of the resulting elements of M will act as an

identity for the $*$ -function. So to get a monoid, we need to throw into M^* some unit 1. However, since we are making no assumptions, we can't assume either that any of the products in M are equal, or that there are any other objects in M other than those generated from the unit and members of S .

Now, here's a neat way to model the resulting monoid 'freely' generated from the set S . Represent a monoid element (such as $x * x * y * y * z$) as a *finite list of members of S* , so M gets represented by $List(S)$ – see (F8) above. Correspondingly, model the $*$ -function by simple concatenation \cap . The identity element will then be modelled by the null list. The resulting $(List(S), \cap, 1)$ is often simply called *the free monoid* on S – though perhaps it is better to say it is a standard exemplar of a free monoid.

Which all goes to motivate the following construction:

(F13) There is a 'free' functor $F : \text{Set} \rightarrow \text{Mon}$ with the following data:

- i. F_{ob} sends the set S to the monoid $(List(S), \cap, 1)$.
- ii. F_{arw} sends the arrow $f : S \rightarrow S'$ to $List(f)$ (see (F8) again), where this is now treated as an arrow from $(List(S), \cap, 1)$ to $(List(S'), \cap, 1)$.

It is now trivial to check that F is indeed a functor.

(b) Note, different set functions $f, g : X \rightarrow Y$ get sent to different functions $Ff, Fg : List(X) \rightarrow List(Y)$ (if $fx \neq gx$, then $Ff(\langle x \rangle) \neq Fg(\langle x \rangle)$, where $\langle x \rangle$ is the list whose sole element is x). So F is faithful.

Now consider a singleton set 1. This gets sent by F to the free monoid with a single generator – which is tantamount to $\mathcal{N} = (\mathbb{N}, +, 0)$. The sole set-function from 1 to itself, the identity function, gets sent by F to the identity monoid homomorphism on \mathcal{N} . But there are other monoid homomorphisms from \mathcal{N} to \mathcal{N} , e.g. $n \mapsto 2n$. So F is not full.

(c) We can generalize. There are similar functors that send sets to other *freely generated* structures on the set. For example there is a functor from Set to Ab which sends a set X to the freely generated abelian group on X (which is in fact the direct sum of X -many copies of $(\mathbb{Z}, +, 0)$ – the integers \mathbb{Z} with addition forming the paradigm free abelian group on a single generator). But we need not concern ourselves with the further details of such cases.

15.6 Products, exponentials, and functors

To develop two examples of a different type, let's consider again first products and then exponentials.

(F14) Assume \mathcal{C} has all products, and C is any object in the category. Then there is a functor $- \times C : \mathcal{C} \rightarrow \mathcal{C}$, which sends an object A to $A \times C$ and

Functors introduced

an arrow $f: A \rightarrow A'$ to $f \times 1_C: A \times C \rightarrow A' \times C$.

Similarly there is a functor $C \times -: \mathcal{C} \rightarrow \mathcal{C}$, which sends an object A to $C \times A$ and an arrow $f: A \rightarrow A'$ to $1_C \times f: C \times A \rightarrow C \times A'$.

Proof. Write $f \times C$ for $(-\times C)(f)$. To confirm functoriality the main thing is to show $(g \circ f) \times C = (g \times C) \circ (f \times C)$. But that is $g \circ f \times 1_C = (g \times 1_C) \circ (f \times 1_C)$, which follows from Theorem 37.

Similarly for the other functor. □

Suppose next that we are working in a category \mathcal{C} which has all exponentials (and all binary products). And suppose we have an arrow $f: C \rightarrow C'$ between a couple of \mathcal{C} -objects. Now pick another object B in the category. Then there is a commuting diagram which looks like this:

$$\begin{array}{ccc} C^B \times B & \xrightarrow{ev} & C \\ \overline{(f \circ ev)} \times 1_B \downarrow & & \downarrow f \\ C'^B \times B & \xrightarrow{ev'} & C' \end{array}$$

Why so? Trivially, there is a composite arrow $f \circ ev: C^B \times B \rightarrow C'$. But then, since $[C'^B, ev']$ is an exponential, there is by definition a *unique* transpose $\overline{f \circ ev}: C^B \rightarrow C'^B$ which makes the diagram commute.

In this way, for fixed B , there is a natural association between the objects C and C^B and another between the arrows $f: C \rightarrow C'$ and $\overline{f \circ ev}: C^B \rightarrow C'^B$. And, as we might hope, the associations are indeed functorial. In other words, we hope that the following is true:

(F15) Assume \mathcal{C} has all exponentials, and that B is a \mathcal{C} -object. Then there is a corresponding exponentiation functor $(-)^B: \mathcal{C} \rightarrow \mathcal{C}$ which sends an object C to C^B , and sends an arrow $f: C \rightarrow C'$ to $\overline{f \circ ev}: C^B \rightarrow C'^B$.

We need, however, to confirm that this is indeed correct:

Proof. We need to confirm that $(-)^B$ does indeed preserve identities and respect composition.

The first is easy. $(1_C)^B$ is by definition $\overline{1_C \circ ev}: C^B \rightarrow C^B$, so we have

$$\begin{array}{ccc} C^B \times B & \xrightarrow{ev} & C \\ (1_C)^B \times 1_B \downarrow & & \downarrow 1_C \\ C^B \times B & \xrightarrow{ev} & C \end{array}$$

But evidently, the arrow $1_{C^B} \times 1_B$ on the left would also make the diagram commute. So by the requirement that there is a unique filling for $-\times 1_B$ which makes the square commute, $(1_C)^B = 1_{C^B}$, as required for functoriality.

15.7 An example from algebraic topology

Second, we need to show that given arrows $f: C \rightarrow C'$ and $g: C' \rightarrow C''$, $(g \circ f)^B = g^B \circ f^B$.

Consider the following diagram where the top square, bottom square, and (outer, bent) rectangle commute:

$$\begin{array}{ccc}
 C^B \times B & \xrightarrow{ev} & C \\
 \downarrow f^B \times 1_B & & \downarrow f \\
 C'^B \times B & \xrightarrow{ev'} & C' \\
 \downarrow g^B \times 1_B & & \downarrow g \\
 C''^B \times B & \xrightarrow{ev''} & C''
 \end{array}$$

$(g \circ f)^B \times 1_B$ (indicated by a dashed curved arrow from $C^B \times B$ to $C''^B \times B$)

By Theorem 37, $(g^B \times 1_B) \circ (f^B \times 1_B) = (g^B \circ f^B) \times 1_B$. Hence $(g^B \circ f^B) \times 1_B$ is another arrow that makes a commuting rectangle. So again by the requirement that there is a unique filling for $- \times 1_B$ which makes the square commute, $(g \circ f)^B = g^B \circ f^B$. \square

15.7 An example from algebraic topology

(a) Here's another particular example of a functor, this time a classic example from algebraic topology. This can readily be skipped if you don't know the setting. Though to get a glimmer of what's going on, you just need the idea of the fundamental group of a topological space (at a point), as follows.

Given a space and a chosen base point in it, consider all directed paths that start at this base point then wander around and eventually loop back to their starting point. Such directed loops can be "added" together in an obvious way: you traverse the "sum" of two loops by going round the first loop, then round the second. Every loop has an "inverse" (you go round the same path in the opposite direction). Two loops are considered 'homotopically' equivalent if one can be continuously deformed into the other. Consider, then, the set of all such equivalence classes of loops – so-called homotopy equivalence classes – and define "addition" for these classes in the obvious derived way. This set, when equipped with addition, evidently forms a group: it is the *fundamental group* for that particular space, with the given basepoint. (Though for many spaces, the group is independent of the basepoint.)

Suppose, therefore, that \mathbf{Top}_* is the category of pointed topological spaces: an object in the category is a topological space X equipped with a distinguished base point x_0 , and the arrows in the category are continuous maps that preserve basepoints. Then here's our new example of a functor:

Functors introduced

(F16) There is a functor $\pi_1: \mathbf{Top}_* \rightarrow \mathbf{Grp}$, to use its standard label, with the following data

- i. π_1 sends a pointed topological space (X, x_0) – i.e. X with base point x_0 – to the fundamental group $\pi_1(X, x_0)$ of X at x_0 .
- ii. π_1 sends a basepoint-preserving continuous map $f: (X, x_0) \rightarrow (Y, y_0)$ to a corresponding group homomorphism $f_*: \pi_1(X, x_0) \rightarrow \pi_1(Y, y_0)$. (For arm-waving motivation: f maps a continuous loop based at x_0 to a continuous loop based at y_0 ; and since f is continuous it can be used to send a continuous deformation of a loop in (X, x_0) to a continuous deformation of a loop in (Y, y_0) – and that induces a corresponding association f_* between the homotopy equivalence classes of (X, x_0) and (Y, y_0) , and this will respect the group structure.)

We will suppose that we have done the work of checking that π_1 is indeed functorial.

(b) Here, then, is a nice application. We'll prove Brouwer's famed Fixed Point Theorem:

Theorem 81. *Any continuous map of the closed unit disc to itself has a fixed point.*

Proof. Suppose that there is a continuous map f on the two-dimensional disc D (considered as a topological space) without a fixed point, i.e. such that we always have $f(x) \neq x$.

Let the boundary of the disc be the circle S (again considered as a topological space). Then we can define a map that sends the point x in D to the point in S at which the ray from $f(x)$ through x intersects the boundary of the disc.

This map sends a point on the boundary to itself. Pick a boundary point to be the base point of the pointed space D_* and also of the pointed space S_* , then our map induces a map $r: D_* \rightarrow S_*$. Moreover, this map is evidently continuous (intuitively: nudge a point x and since f is continuous that just nudges $f(x)$, and hence the ray from $f(x)$ through x is only nudged, and the point of intersection with the boundary is only nudged). And r is a left inverse of the inclusion map $i: S_* \rightarrow D_*$ in \mathbf{Top}_* , since $r \circ i = 1$.

Functors preserve left inverses by Theorem 78, so $\pi_1(r)$ will be a left inverse of $\pi_1(i)$, which means that $\pi_1(i): \pi_1(S_*) \rightarrow \pi_1(D_*)$ is a right-inverse in \mathbf{Grp} , hence by Theorem 11 is monic, and hence by Theorem 7 is an injection.

But that's impossible. $\pi_1(S_*)$, the fundamental group of S_* , is [equivalent to] the group \mathbb{Z} of integers under addition (think of looping round a circle, one way or another, n times – each positive or negative integer corresponds to a different path); while $\pi_1(D_*)$, the fundamental group of D_* , is just a one element group (for every loop in the disk D_* can be smoothly shrunk to a point). And there is no injection between the integers and a one-element set! \square

(c) What, if anything, do we gain from putting the proof in category theoretic terms? It might be said: the proof crucially depends on facts of algebraic topology – continuous maps preserve homotopic equivalences in a way that makes π_1 a functor, and the fundamental groups of S^* and D^* are respectively \mathbb{Z} and the trivial group. And we could run the whole proof without actually mentioning categories at all. Still what we’ve done is, so to speak, very clearly demarcate those bits of the proof that depend on topic-specific facts of algebraic topology and those bits which depend on general proof-ideas about functoriality and about kinds of maps (inverses, monics, injections), ideas which are thoroughly *portable* to other contexts. And *that* surely counts as a gain in understanding.

15.8 Covariant vs contravariant functors

Here, finally, is another a very general question about functors. How do they interact with the operation of taking the opposite category?

Well, first we note:

Theorem 82. *A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ induces a functor $F^{op}: \mathcal{C}^{op} \rightarrow \mathcal{D}^{op}$.*

Proof. Recall, the objects of \mathcal{C}^{op} are exactly the same as the objects of \mathcal{C} . We can therefore define the object-mapping component of F^{op} as acting on \mathcal{C}^{op} -objects exactly as the object-mapping component of F acts on \mathcal{C} -objects. And then, allowing for the fact that taking opposites reverses arrows, we can define the arrow-mapping component of F^{op} as acting on the \mathcal{C}^{op} -arrow $f: C \rightarrow D$ exactly as the arrow-mapping component of F acts on the \mathcal{C} -arrow $f: D \rightarrow C$.

F^{op} will evidently obey the axioms for being a functor because F does. \square

By the way, had we shown this before, we could have halved the work in our proof of Theorem 77 that functors do not necessarily preserve monics or epics. After we’d shown that $F: \mathcal{C} \rightarrow \mathcal{D}$ doesn’t preserve monics, we could have just remarked that the $F^{op}: \mathcal{C}^{op} \rightarrow \mathcal{D}^{op}$ won’t preserve epics!

Now for a new departure. We introduce a variant kind of functor:

Definition 82. $F: \mathcal{C} \rightarrow \mathcal{D}$ is a *contravariant* functor from \mathcal{C} to \mathcal{D} if $F: \mathcal{C}^{op} \rightarrow \mathcal{D}$ is a functor in the original sense. So it comprises the following data:

- (1) A mapping F_{ob} whose value at the \mathcal{C} -object A is some \mathcal{D} -object $F(A)$.
- (2) A mapping F_{arw} whose value at the \mathcal{C} -arrow $f: B \rightarrow A$ is a \mathcal{D} -arrow $F(f): FA \rightarrow FB$. (NB the directions of the arrows!)

And this data satisfies the two axioms:

Preserving identities: for any \mathcal{C} -object A , $F(1_A) = 1_{F(A)}$;

Respecting composition: for any \mathcal{C} -arrows f, g such that their composition $g \circ f$ exists, $F(g \circ f) = Ff \circ Fg$. (NB the order of the two compositions!) \triangle

Functors introduced

Two comments. First, a functor in our *original* sense, when the contrast needs to be stressed, is also called a *covariant* functor. Second, it would of course be equivalent to define a contravariant functor from \mathcal{C} to \mathcal{D} to be a covariant functor from \mathcal{C} to \mathcal{D}^{op} .

Let's finish the chapter, then, with a couple of examples of naturally arising contravariant functors.

- (1) We have already met the covariant powerset functor. Its contravariant twin $\overline{P}: \mathbf{Set} \rightarrow \mathbf{Set}$ again maps a set to its powerset, and maps a set-function $f: Y \rightarrow X$ to the function which sends $U \in \mathcal{P}(X)$ to its inverse image $f^{-1}[U] \in \mathcal{P}(Y)$ (where $f^{-1}[U] = \{x \mid f(x) \in U\}$).
- (2) Take \mathbf{Vect} , the category whose objects are the finite dimension vector spaces over the reals, and whose arrows are linear maps between spaces.

Now recall, the dual space of given finite-dimensional vector space V over the reals is V^* , the set of all linear functions $f: V \rightarrow \mathbb{R}$ (where this set is equipped with vectorial structure in the obvious way). V^* has the same dimension as V (so, a fortiori, is also finite dimensional and belongs to \mathbf{Vect}). We'll construct a dualizing functor $D: \mathbf{Vect} \rightarrow \mathbf{Vect}$, where D_{ob} sends a vector-space to its dual.

So how is our functor D going to act on arrows in the category \mathbf{Vect} ? Take spaces V, W and consider any linear map $g: W \rightarrow V$. Then, over on the dual spaces, there will be a naturally corresponding map $(-\circ g): V^* \rightarrow W^*$ which maps $f: V \rightarrow \mathbb{R}$ to $f \circ g: W \rightarrow \mathbb{R}$. *But note the direction that the arrow g has to go in, if composition with f is to work.* This defines the action of a component D_{arw} for the dualizing functor D : it will send a linear map g to the map $(-\circ g)$.

And these components D_{ob} and D_{arw} evidently do give us a contravariant functor.

16 Categories of categories

We have seen how structured whatnots and structure-respecting maps between them can be assembled into categories. This gives us more structured data, the categories; and now we have also seen there are structure-respecting maps between *them*, i.e. functors. Can data of these last two sorts be assembled into further categories? Yes indeed. Quite unproblematically, there are at least some categories of categories.

However, just as we can have many sets of sets but arguably not, on pain of paradox, a set of *all* sets, so we can have many categories of categories but arguably not, on pain of paradox, a category of *all* categories. Some collections are, as the saying goes, ‘too big to be sets’; there are similar worries about some assemblies of categories being ‘too big’. We need then briefly to address these issues of size, which we have previously skated around once or twice.

16.1 Functors compose

Here are two simple theorems. In each case the proof is entirely straightforward from the definitions:

Theorem 83. *Given any category \mathcal{C} there is an identity functor $1_{\mathcal{C}}: \mathcal{C} \rightarrow \mathcal{C}$ which sends objects and arrows alike to themselves.*

Theorem 84. *Suppose there exist functors $F: \mathcal{C} \rightarrow \mathcal{D}$, $G: \mathcal{D} \rightarrow \mathcal{E}$. Then there is also a composite functor $G \circ F: \mathcal{C} \rightarrow \mathcal{E}$ with the following data:*

- (1) *A mapping $(G \circ F)_{ob}$ which sends a \mathcal{C} -object A to the \mathcal{E} -object GFA – i.e., if you prefer that with brackets, to $G(F(A))$.*
- (2) *A mapping $(G \circ F)_{arw}$ which sends a \mathcal{C} -arrow $f: A \rightarrow B$ to the \mathcal{E} -arrow $GFf: GFA \rightarrow GFB$ – i.e. to $G(F(f))$.*

Further, such composition of functors is associative.

By the way, again to reduce clutter, we will later often allow ourselves to write simply ‘ GF ’ for the composite functor rather than ‘ $G \circ F$ ’.

What happens if we compose two contravariant functors?

Theorem 85. *The composition of two contravariant functors, where defined, yields a covariant functor.*

That's immediate once we reflect that if the contravariant F and G compose, F sends an arrow $f: A \rightarrow B$ to $Ff: FB \rightarrow FA$ and then G sends that on to $GFf: GFA \rightarrow GFB$.

In other respects too, composition behaves just as you would expect on a moment's thought. For example:

Theorem 86. *The composition of full functors is full and the composition of faithful functors is faithful.*

Again the proof writes itself. Being full is being locally surjective, and compositions of surjective functions are surjective; similarly for faithfulness.

16.2 Categories of categories

The basic observations that there are identity functors, and that functors compose associatively now ensure that the following definition is in good order:

Definition 83. Suppose \mathcal{X} comprises two sorts of data:

- (1) *Objects:* some categories, $\mathcal{C}, \mathcal{D}, \mathcal{E}, \dots$,
- (2) *Arrows:* some functors, F, G, H, \dots , between those categories,

where the arrows (i) include the identity functor on each category, and (ii) also include $G \circ F$ for each included composable pair F and G (where F 's target is G 's source). Then \mathcal{X} is a *category of categories*. \triangle

Let's have some quick examples:

- (1) Trivially, there is a category of categories whose sole object is the category \mathcal{C} and whose sole arrow is identity functor $1_{\mathcal{C}}$.
- (2) We noted that every monoid can be thought of as itself being a category. Hence the familiar category Mon can also be regarded as a category of categories.
- (3) There is a category whose objects are the finite categories, and whose arrows are all the functors between finite categories.

So there certainly are *some* examples of categories of categories. But, as we have already indicated, there are limitations.

16.3 A universal category?

(a) Suppose we next say:

Definition 84. A category is *normal* iff it is not one of its own objects. \triangle

The categories which we have met in previous chapters have all been normal. Now ask: can all the normal categories be gathered together as the objects of one really big category?

The answer is given by

Theorem 87. *There is no category of all normal categories.*

Proof. Suppose there is a category \mathcal{N} whose objects are all the normal categories. Now ask, is \mathcal{N} normal? If it is, then it is one of the objects of \mathcal{N} , so \mathcal{N} is non-normal. So \mathcal{N} must be non-normal. But then it is not one of the objects of \mathcal{N} , so \mathcal{N} is normal after all. Contradiction. \square

This argument of course just re-runs, in our new environment, the very familiar argument from Russell's Paradox to the conclusion that there is no set of all the normal sets (where a set is normal iff it is not a member of itself).

It is worth stressing that the Russellian argument is *not* especially to do with sets, for at its core is a simple, purely logical, observation. Thus, take *any* two-place relation R defined over some objects; then there can be no object r among them which is related by R to all and only those objects which are not R -related to themselves. In other words, it is a simple logical theorem that $\neg\exists r\forall x(Rxr \leftrightarrow \neg Rxx)$. Russell's original argument applies this elementary general result to the particular set-theoretic relation R_1 , ' \dots is a set which is a member of the set \dots ', to show that there is no set of all normal (i.e. non-self-membered) sets. Our argument above now applies the same logical theorem to the analogous category-theoretic relation R_2 , ' \dots is a category which is an object of the category \dots ', to show that there is no category of all normal categories.

(b) Russell's original argument that there is no set of all *normal* sets is usually taken to entail that, a fortiori, there is no universal set, no set of *all* sets. The reasoning being that if there were a universal set then we should be able carve out of it (via a separation principle) a subset containing just those sets which are normal, which we now know can't be done.

To keep ourselves honest, however, we should note that this *further* argument can be, and has been, resisted. There are cogent set theories on the market which allow universal sets. How can this possibly be? Well, we can motivate restricting separation and can thus block the argument that, if there is a universal set of all sets, we should in particular be able to carve out from it a set of all normal sets: see Forster (1995) for a classic discussion of set theories with a universal set which work this way. But we can't discuss this type of deviant theory here. Henceforth we'll have to just assume a standard line on sets at least in this

Categories of categories

respect – there are ‘limitations of size’, i.e. there are some entities (e.g. the sets themselves) which are too many to form a set.

Now, similarly to the argument about sets, the Russellian argument that there is no category of all normal categories might naturally be taken to entail that there is no universal category in the naive sense:

Definition 85. A category \mathcal{U} is *universal* if it is a category of categories such that every category is an object of \mathcal{U} .

Theorem 88? *There is no universal category.*

The argument goes: suppose a universal category \mathcal{U} exists. Then we could carve out from it a subcategory whose objects are just the normal categories, to get a category of all normal categories. But we have shown there can be no such category.

Can this line of argument be resisted? Could we justify saying that even if there is a category of *all* categories, we can’t actually select out the normal categories and all the arrows between them to give us a subcategory of *normal* categories? Well, perhaps some themes in the debates about set theories with a universal set could be carried over to this case. But again, it would take us far too far away from mainstream concerns in category theory to try to explore this option any further here.

Let’s not fuss about the possibility of a universal category any more but simply take it that, at least in the naive sense of Defn. 85, there is no such thing. Instead, we turn our attention to defining two much more useful notions of large-but-less-than-universal categories-of-categories.

16.4 ‘Small’ and ‘locally small’ categories

(a) To repeat: when we talk here about sets, we assume we are working in a theory of sets which is standard at least in the respect of allowing that the sets are too many to themselves form a set.

We continue with a three new definitions:

Definition 86. A category \mathcal{C} is *finite* iff it has overall only a finite number of arrows.

A category \mathcal{C} is *small* iff it has overall only a ‘set’s worth’ of arrows – i.e. the arrows of \mathcal{C} can be put into one-one correspondence with the members of some set.

A category \mathcal{C} is *large* iff it isn’t small overall. But it counts as *locally small* iff for every pair of \mathcal{C} -objects C, D there is only a ‘set’s worth’ of arrows from C to D , i.e. those arrows can be put into one-one correspondence with the members of some set. \triangle

Some comments and examples:

- (1) The terms ‘small’ and ‘locally small’ are standard.
- (2) It would be more usual to say that in a small category the arrows themselves form a set. However, if our favoured set theory is a theory like pure ZFC where sets only have other sets as members, that would presuppose that arrows are themselves pure sets, and we might not necessarily want to make that assumption. So, for smallness, let’s officially require only that the arrows aren’t too many to be indexed by a set. Similarly for local smallness.
- (3) Since for every object in \mathcal{C} there is at least one arrow, namely the identity arrow on \mathcal{C} , a finite category must have a finite number of objects. And if there are too many objects of \mathcal{C} to be bijectively mapped to a set, then \mathcal{C} has too many arrows to be small. Contraposing, if \mathcal{C} is small, not only its arrows but its objects can be put into one-one correspondence with the members of some set (in fact, the set that indexes the identity arrows).
- (4) Among our examples in §3.6, tiny finite categories like **1** and **2** are of course small. But so too are the categories corresponding to an infinite but set-sized monoid or to an infinite pre-ordered set. Categories such as **Set** or **Mon**, however, have too many objects (and hence too many arrows) to be small.
- (5) While categories such as **Set** or **Mon** are not small, like all our other examples so far they are at least *locally* small. In **Set**, for example, the arrows between objects C to D are members of a certain subset of the powerset of $C \times D$: which makes **Set** locally small. (Indeed some authors build local smallness into their preferred definition of a category – see for example Schubert 1972, p. 1; Borceux 1994, p. 4; Adámek et al. 2009, p. 21.)

(b) Let’s propose two more definitions:

Definition 87. **Cat** is the category whose objects are small categories and whose arrows are the functors between them.

Cat* is the category whose objects are locally small categories and whose arrows are the functors between them. △

Are such definitions in good order?

Well, at least there aren’t Russellian problems. First, a discrete category (with just identity arrows) only has as many arrows as objects. Which implies that the discrete category on any set is small. But that in turn implies that there are at least as many small categories as there are sets. Hence the category **Cat** of small categories has at least as many objects as there are sets, and hence is itself determinately *not* small. Since **Cat** is unproblematically not small, no paradox arises for **Cat** as it did for the putative category of normal categories.

Second, take a one-element category **1**, which is certainly locally small. Then a functor from **1** to **Set** will just map the object of **1** to some particular set: and

there will be as many distinct functors $F: \mathbf{1} \rightarrow \mathbf{Set}$ as there are sets. In other words, arrows from $\mathbf{1}$ to \mathbf{Set} in \mathbf{Cat}^* are too many to be mapped one-to-one to a set. Hence \mathbf{Cat}^* is determinately *not* locally small. So again no Russellian paradox arises for \mathbf{Cat}^* .

16.5 Isomorphisms between categories

(a) It seems, therefore, that we can legitimately talk of the category of small categories \mathbf{Cat} . And if we don't build local smallness into the very definition of a category, as some do, then it seems that we can legitimately talk of the larger category of locally small categories \mathbf{Cat}^* . Maybe we can countenance still more inclusive categories of categories.

It will be handy to have some flexible notation to use, in a given context, for a suitable category of categories that includes at least all the categories which are salient in that context: let's use \mathbf{CAT} for this. We can then start applying familiar categorial definitions. For example,

Definition 88. A functor $F: \mathcal{C} \xrightarrow{\sim} \mathcal{D}$ is an *isomorphism* between categories in \mathbf{CAT} iff it has an inverse, i.e. there is a functor $G: \mathcal{D} \rightarrow \mathcal{C}$ where $G \circ F = 1_{\mathcal{C}}$ and $F \circ G = 1_{\mathcal{D}}$. \triangle

Here, $1_{\mathcal{C}}$ is of course the functor that sends every object to itself and every arrow to itself. And the definition makes the notion of being an isomorphism sensibly stable in the sense that if $F: \mathcal{C} \xrightarrow{\sim} \mathcal{D}$ is an isomorphism between categories in some \mathbf{CAT} it remains an isomorphism in a more inclusive category.

As we would expect,

Theorem 89. *If $F: \mathcal{C} \xrightarrow{\sim} \mathcal{D}$ is an isomorphism, it is full and faithful.*

Proof. First suppose we have parallel arrows in \mathcal{C} , namely $f, g: A \rightarrow B$. Supposing $Ff = Fg$, then $GFf = GFg$ – where G is F 's inverse (now suppressing the clutter of explicit composition signs). So $1_{\mathcal{C}}f = 1_{\mathcal{C}}g$ and hence $f = g$. Therefore F is faithful.

Suppose we are given an arrow $h: FA \rightarrow FB$. Put $f = Gh$. Then $Ff = FGH = 1_{\mathcal{D}}h = h$. So every such h in \mathcal{D} is the image under F of some arrow in \mathcal{C} . So F is full. \square

The converse doesn't hold, however. We noted that the inclusion functor from a full subcategory \mathcal{S} of \mathcal{C} into \mathcal{C} is fully faithful: but plainly that usually won't have an inverse.

(b) Just as we say that objects C and D inside a category are isomorphic iff there is an isomorphism $f: C \rightarrow D$, so we naturally say:

Definition 89. Categories \mathcal{C} and \mathcal{D} are *isomorphic* in \mathbf{CAT} , in symbols $\mathcal{C} \cong \mathcal{D}$, iff there is an isomorphism $F: \mathcal{C} \xrightarrow{\sim} \mathcal{D}$. \triangle

Let's have some examples:

- (1) Take the toy two-object categories with different pairs of objects which we can diagram as

$$\hookrightarrow \bullet \longrightarrow \star \curvearrowright \qquad \hookrightarrow a \longrightarrow b \curvearrowright$$

Plainly they are isomorphic (and indeed there is a unique isomorphic functor that sends the first to the second)! If we don't care about distinguishing copies of structures that are related by a unique isomorphism, then we'll count these as the same in a strong sense. Which to that extent warrants our earlier talk about *the* category $\mathbf{2}$ (e.g. in §3.6, Ex. (C7)).

- (2) Revisit the example in §4.3 of the coslice category $1/\mathbf{Set}$. This category has as objects all the arrows $\vec{x}: 1 \rightarrow X$ for any $X \in \mathbf{Set}$. And the arrows from $\vec{x}: 1 \rightarrow X$ to $\vec{y}: 1 \rightarrow Y$ are just the set-functions $j: X \rightarrow Y$ such that $j \circ \vec{x} = \vec{y}$.

Now we said before that this is in some strong sense 'the same as' the category \mathbf{Set}_* of pointed sets. And indeed the categories are isomorphic. For take the function F_{ob} from objects in $1/\mathbf{Set}$ to objects \mathbf{Set}_* which sends an object $\vec{x}: 1 \rightarrow X$ to the pointed set (X, x) , i.e. X -equipped-with-the-basepoint- x , where x is the value of the function \vec{x} for its sole argument. And take F_{arw} to send an arrow $j: X \rightarrow Y$ such that $j \circ \vec{x} = \vec{y}$ to an arrow $j': (X, x) \rightarrow (Y, y)$ agreeing at every argument and preserving base points. Then it is trivial to check that F is a functor $F: 1/\mathbf{Set} \rightarrow \mathbf{Set}_*$.

In the other direction, we can define a functor $G: \mathbf{Set}_* \rightarrow 1/\mathbf{Set}$ which sends (X, x) to the function $\vec{x}: 1 \rightarrow X$ which sends the sole object in 1 to the point x , and sends a basepoint-preserving function from X to Y to itself.

And it is immediate that these two functors F and G are inverse to each other. Hence, as claimed, $\mathbf{Set}_* \cong 1/\mathbf{Set}$.

- (3) For those who know just a little about Boolean algebras and the two alternative ways of presenting them: There is a category \mathbf{Bool} whose objects are algebras $(B, \neg, \wedge, \vee, 0, 1)$ constrained by the familiar Boolean axioms, and whose arrows are homomorphisms that preserve algebraic structure. And there is a category \mathbf{BoolR} whose objects are Boolean rings, i.e. rings $(R, +, \times, 0, 1)$ where $x^2 = x$ for all $x \in R$, and whose arrows are ring homomorphisms.

There is also a familiar way of marrying up Boolean algebras with corresponding rings, and vice versa. Thus if we start from $(B, \neg, \wedge, \vee, 0, 1)$, take the same carrier set and distinguished objects, put

- (i) $x \times y =_{\text{def}} x \wedge y$,
- (ii) $x + y =_{\text{def}} (x \vee y) \wedge \neg(x \wedge y)$ (exclusive 'or'),

then we get a Boolean ring. And if we apply the same process to two algebras B_1 and B_2 , it is elementary to check that this will carry a homo-

morphism of algebras $f_a: B_1 \rightarrow B_2$ to a corresponding homomorphism of rings $f_r: R_1 \rightarrow R_2$. We can equally easily go from rings to algebras, by putting

- (i) $x \wedge y =_{\text{def}} x \times y$,
- (ii) $x \vee y =_{\text{def}} x + y + (x \times y)$
- (iii) $\neg x =_{\text{def}} 1 + x$.

Note that going from an algebra to the associated ring and back again takes us back to where we started.

In summary, without going into any more details, we can in this way define a functor $F: \mathbf{Bool} \rightarrow \mathbf{BoolR}$, and a functor $G: \mathbf{BoolR} \rightarrow \mathbf{Bool}$ which are inverses to each other. So, as we'd surely have expected, the category \mathbf{Bool} is isomorphic to the category \mathbf{BoolR} .

- (4) We will meet two more examples of isomorphic categories in §19.3.

So far, so good then. We have examples of pairs of categories which, intuitively, 'come to just the same' and are indeed isomorphic by our definition. Looking ahead to Chapter 23, however, it turns out that being isomorphic is not the notion of 'amounting to the same category' which is most useful. We in fact need a rather more relaxed notion of equivalence of categories. More about this later.

- (c) For the moment, then, we just note that we can also carry over e.g. our categorial definition of initial and terminal objects and other limits to categories in \mathbf{CAT} . We can check the following, for example:

Theorem 90. *The empty category is initial in \mathbf{CAT} , and the trivial one-object category $\mathbf{1}$ is terminal.*

Theorem 91. *The category $\mathcal{C} \times \mathcal{D}$ (as defined in §4.2), equipped with the obvious projection functors $\Pi_1: \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$ and $\Pi_2: \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{D}$ forms a categorial binary product of \mathcal{C} with \mathcal{D} .*

16.6 An aside: other definitions of categories

- (a) Having at long last explicitly highlighted the theme of categories with too many objects to form a set, now is the moment to pause to revisit our definition of the very idea of a category to explain its relation to other, slightly different, definitions. For issues of size crop up again.

Our own preferred definition began like this:

Definition 4 A category \mathcal{C} comprises two kinds of things:

- (1) *Objects* (which we will typically notate by ' A ', ' B ', ' C ', ...).
- (2) *Arrows* (which we typically notate by ' f ', ' g ', ' h ', ...). ...

This accords with e.g. Awodey (2006, p. 4) and Lawvere and Schanuel (2009, p. 21). And this is given as a ‘direct description’ of categories by (Mac Lane, 1997, p. 289). However, it is at least as common to put things as follows:

Definition 4* A category \mathcal{C} consists of

- (1) A collection Obj of entities called *objects*.
- (2) A collection Arw of entities called *arrows*. . . .

See (Goldblatt, 2006, p. 24), and Simmons (2011, p. 2) for such definitions, and also e.g. Goedecke (2013).

Others prefer to talk of ‘classes’ here, but we probably shouldn’t read very much into *that* choice of wording, ‘collections’ vs ‘classes’. The real question is: what, if anything, is the difference between talking of a category as having as data some objects (plural) and some arrows (plural), and saying that a category consists in a collection/class (singular) of objects and a collection/class (singular) of arrows?

It obviously all depends what we mean here by ‘collections’. Because many paradigm categories have too many objects for there to be a set of them, the notion of collection can’t be just the standard notion of a set again. But that still leaves options. Is Defn. 4* in fact intended to involve only ‘virtual classes’, meaning that the apparent reference to classes is a useful fiction but can be translated away so that it ends up saying no more than is said by Defn. 4 which doesn’t refer to collections-as-special-objects at all? Or is Defn. 4* to be read as buying into some overall two-layer theory of sets-plus-bigger-classes which in some way takes *large* collections, classes-which-aren’t-sets, more seriously (and if so, then *how* seriously)?

Well, note that we have in fact been able to proceed quite far without making any clear assumption that categories are in some strong sense distinct entities over and above their objects and arrows (arguably, even talk of categories of categories doesn’t commit us to that). In other words, it isn’t obvious that we as yet *need* to buy in to a substantive theory of classes to get our theorizing about categories off the ground. For this reason, I prefer to stick to the overtly non-committal Defn. 4 as our initial definition, and thereby leave it as a separate question just when, and in what contexts, the category theorist eventually does make moves that require taking seriously collections-bigger-than-any-ordinary-set.

(b) While on the subject of variant definitions of category, here’s another common one. It starts like this:

Definition 4** The data for a category \mathcal{C} comprises:

- (1) A collection $ob(\mathcal{C})$, whose elements we will call *objects*.

Categories of categories

- (2) For every $A, B \in ob(\mathcal{C})$, a collection $\mathcal{C}(A, B)$, whose elements f we will call *arrows* from A to B . We signify that the arrow f belongs to $\mathcal{C}(A, B)$ by writing $f: A \rightarrow B$ or $A \xrightarrow{f} B$.
- (3) For every $A \in ob(\mathcal{C})$, an arrow $1_A \in \mathcal{C}(A, A)$ called the *identity* on A .
- (4) For any $A, B, C \in ob(\mathcal{C})$, a two-place *composition* operation, which takes arrows f, g , where $f: A \rightarrow B$ and $g: B \rightarrow C$, to an arrow $g \circ f: A \rightarrow C$, the composite of f and g \triangle

This is essentially the definition given by Leinster (2014, p. 10). Relatedly, consider Borceux (1994, p. 4) and Adámek et al. (2009, p. 18) who have a category consisting of a class of objects but who insist that each collection of arrows between specific objects is to be a *set* – so they build local smallness into the very definition of a category.

Leaving aside the last point, the key difference is that Defn. 1* has one all-in class of arrows, Defn. 1** has lots of different classes (or sets) of arrows, one for every pair of objects in the category.

Obviously if we start from Defn. 1 or Defn. 1*, we can then augment it by defining the collection $\mathcal{C}(A, B)$ of arrows from A to B as containing the \mathcal{C} -arrows f such that $src(f) = A$ and $tar(f) = B$. Note, though, on Defn. 1 or Defn. 1* the arrows $f: A \rightarrow B$ and $f': A' \rightarrow B'$ cannot be identical if $A \neq A'$ or $B \neq B'$. For if $src(f) \neq src(f')$, $f \neq f'$; likewise, of course, if $tar(f) \neq tar(f')$, $f \neq f'$. Hence, according to the now augmented Defn. 1 or Defn. 1*, if $A \neq A'$ or $B \neq B'$, $\mathcal{C}(A, B)$ and $\mathcal{C}(A', B')$ are disjoint. On the other hand, there's nothing in Defn. 1** which requires that. Which means that our two definitions don't quite line up. What to do?

The easy option is just to add to Defn. 1** the stipulation that the collections $\mathcal{C}(A, B)$ for different pairs of objects A, B are indeed disjoint. Adámek et al. (2009) adds just such a stipulation 'for technical convenience' and Leinster (2014) does the same. If we stick though to our original definition Defn. 1 (or to Defn. 1*, if you insist), then you get the same requirement for free.

17 Functors and limits

As we have seen, a functor $F: \mathcal{J} \rightarrow \mathcal{C}$ will, just in virtue of its functoriality, preserve/reflect some aspects of the categorial structure of \mathcal{J} as it sends objects and arrows into \mathcal{C} . And if the functor has properties like being full or faithful it will preserve/reflect more.

We now want to ask: how do things stand with respect to preserving/reflecting limits and colimits?

17.1 Diagrams redefined as functors

(a) Now that we have the notion of a functor to hand, we can redefine the notion of a diagram, and hence the notion of a (co)limit over a diagram, in a particularly neat way.

We can think of a functor from one category to another as producing a kind of image or representation of the first category which lives in the second category – see the beginning of §15.3. Or, to say the same thing in other words, a functor $D: \mathcal{J} \rightarrow \mathcal{C}$ produces a sort of diagram of the category \mathcal{J} inside \mathcal{C} . This thought in turn motivates overloading terminology in the following standard way:

Definition 90. Given a category \mathcal{C} , and a category \mathcal{J} , we say that a functor $D: \mathcal{J} \rightarrow \mathcal{C}$, is a *diagram (of shape \mathcal{J}) in \mathcal{C}* . \triangle

(Here we start following what seems a rather common font-convention, and use e.g. ‘ \mathcal{J} ’ rather than ‘ \mathcal{J} ’ when a small – often *very* small – category is likely to be in focus: some indeed would build the requirement that \mathcal{J} is small into our definition here of a diagram-as-functor.)

To go along with this definition of diagrams-as-functors, there are entirely predictable corresponding definitions of cones and limit cones (we just modify in obvious ways the definitions we met in §10.1, 10.2):

Definition 91. Suppose we are given a category \mathcal{C} , together with \mathcal{J} a (possibly very small) category, and a diagram-as-functor $D: \mathcal{J} \rightarrow \mathcal{C}$. Then:

- (1) A *cone over D* is an object $C \in \mathcal{C}$, together with an arrow $c_J: C \rightarrow D(J)$ for each \mathcal{J} -object J , such that for any \mathcal{J} -arrow $d: K \rightarrow L$, $c_L = D(d) \circ c_K$.

Functors and limits

We use $[C, c_J]$ (where ‘ J ’ is understood to run over objects in J) for such a cone.

- (2) A *limit cone over D* is a cone we can notate $[Lim_{\leftarrow J} D, \lambda_J]$ such that for every cone $[C, c_J]$ over D , there is a unique arrow $u: C \rightarrow Lim_{\leftarrow J} D$ such that, for all J -objects J , $\lambda_J \circ u = c_J$. △

(b) How does our talk of diagrams and limits, old and new, interrelate? Three points:

- (1) To repeat the motivating thought, a functor $D: J \rightarrow \mathcal{C}$ will send the objects and arrows of J to a corresponding handful of objects and arrows sitting inside \mathcal{C} and those latter objects will be indexed by the objects of J . So diagrams-as-functors of course generate diagrams-in-categories in the sense introduced rather loosely in §3.7 and then refined in §10.2.
- (2) But on the other hand, not every diagram-in- \mathcal{C} in the original sense corresponds to a diagram-as-functor. There’s a trivial reason. A diagram of shape J in \mathcal{C} will always carry over the required identity arrows on all the objects in J to identity arrows on all their images. But a diagram-in-a-category as we first defined it doesn’t have to have identity arrows on all (or indeed any) of its objects.
- (3) Still, the lack of a straight one-to-one correspondence between diagrams in the two senses makes no difference when thinking about limits. Limits over diagrams-as-functors will of course be limits in the old sense. And conversely, suppose $[L, \lambda_j]$ is a limit cone over some diagram D in \mathcal{C} (diagram in the original sense). Then by Theorem 46, $[L, \lambda_j]$ is a limit over the (reflexive, transitive) closure of D (because *every* cone over D is equally a cone over its closure). By Theorem 45, we can think of this closure as a subcategory J of \mathcal{C} . So take the inclusion functor $D_i: J \rightarrow \mathcal{C}$. Then, by our new definition, $[L, \lambda_j]$ is a limit cone over the diagram-as-functor $D_i: J \rightarrow \mathcal{C}$. In short, limits old and new are just the same.

If our prime interest is in limits, then, we can in fact take the neat notion of diagram just introduced in Defn. 90 to be the primary one. And indeed, this line is widely, though not universally, adopted (compare e.g. Borceux 1994 and Leinster 2014). We too will think of diagrams this way from now on.

17.2 Preserving limits

(a) Start with a natural definition, extending the notion of preservation we met in §15.3: we say a functor preserves limits if it sends limits of a given shape to limits of the same shape and preserves colimits if it sends colimits to colimits. More carefully,

Definition 92. A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves the limit $[L, \lambda_J]$ over $D: J \rightarrow \mathcal{C}$ iff $[FL, F\lambda_J]$ is a limit over $F \circ D: J \rightarrow \mathcal{D}$.

More generally, a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves limits of shape J in \mathcal{C} iff, for any diagram $D: J \rightarrow \mathcal{C}$, if $[L, \lambda_J]$ is some limit cone over D , then F preserves it.

A functor which preserves limits of shape J in \mathcal{C} for all finite (small) categories J is said to preserve all finite (small) limits (in \mathcal{C}).

Dually for preserving colimits. △

Preservation indeed behaves as you would expect in various respects. We will mention two:

Theorem 92. If F preserves products, then $F(A \times B) \cong FA \times FB$.

Proof. Assume F is a functor from \mathcal{C} to \mathcal{D} . Suppose $\bar{2}$ is the discrete category with two objects, call them 0 and 1. Then, in terms of our new notion of a diagram, a product in \mathcal{C} is a limit over some diagram $D: \bar{2} \rightarrow \mathcal{C}$. Take the diagram where $D(0) = A$ and $D(1) = B$. Then the product of course will be some $[A \times B, \pi_1, \pi_2]$.

By hypothesis, $[F(A \times B), F\pi_1, F\pi_2]$ is a limit over the diagram $F \circ D: \bar{2} \rightarrow \mathcal{D}$. That is to say it is a limit over the diagram in \mathcal{D} (in our old sense of diagram) with just the objects FA and FB and their identity arrows. So it is a product; and another product over that diagram is $[FA \times FB, \pi'_1, \pi'_2]$ with appropriate projection arrows. By Theorem 26, these two products are isomorphic, hence $F(A \times B) \cong FA \times FB$. □

Theorem 93. If $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves some limit over the diagram $D: J \rightarrow \mathcal{C}$, it preserves all limits over that diagram.

Proof. Suppose $[L, \lambda_J]$ is a limit cone over $D: J \rightarrow \mathcal{C}$. Then, by Theorem 47, if $[L', \lambda'_J]$ is another such cone, there is an isomorphism $f: L' \rightarrow L$ in \mathcal{C} such that $\lambda'_J = \lambda_J \circ f$.

Suppose now that F preserves $[L, \lambda_J]$ so $[FL, F\lambda_J]$ is a limit cone over $F \circ D$. Then F will send $[L', \lambda'_J]$ to $[FL', F\lambda'_J] = [FL', F\lambda_J \circ Ff]$. But then this factors through $[FL, F\lambda_J]$ via the isomorphism $Ff: FL' \rightarrow FL$ (remember, functors preserve isomorphisms). Hence, by Theorem 48, $[FL', F\lambda'_J]$ is also a limit over $F \circ D$. In other words, F preserves $[L', \lambda'_J]$ too. □

But these general conditional claims don't tell us anything about which particular products or other limits actually do get preserved by which functors: now we need to get down to cases.

(b) Here is a first very simple example and then two further (rather artificial) toy examples, which together nicely illustrate some general points about how functors can *fail* to preserve limits.

Functors and limits

- (1) Take the posets $(\{0, 1, 2\}, \leq)$ and (\mathbb{N}, \leq) thought of as categories. There is a trivial inclusion functor I from the first category to the second. Now, 2 is a terminal object in the first category, but $2 = I(2)$ is not terminal in the second. So I doesn't preserve that terminal object (the limit over the diagram-as-functor from the empty category).

I does, however, preserve products (recall the product of two elements in a poset, when it exists, is their greatest lower bound).

Two morals. First, since a functor need not preserve even terminal objects, functors certainly need not preserve limits generally. Second, a functor may preserve some limits and not others.

There is entertainment to be had in looking at a couple more illustrations of that second point:

- (2) Take the functor $P: \mathbf{Set} \rightarrow \mathbf{Set}$ which sends the empty set \emptyset to itself and sends every other set to the singleton 1, and acts on arrows in the only possible way if it is to be a functor (i.e. for $A \neq \emptyset$, it sends any arrow $\emptyset \rightarrow A$ to the unique arrow $\emptyset \rightarrow 1$, it sends the arrow $\emptyset \rightarrow \emptyset$ to itself and sends all other arrows to the identity arrow 1_1). Claim: P preserves binary products but not equalizers – i.e. it preserves all limits of the shape of the discrete two-object category but not all those of shape $\hookrightarrow \bullet \rightrightarrows \star \curvearrowright$.

Proving this claim is a routine exercise. For the first half, we simply consider cases. If neither A nor B is the empty set, then $A \times B$ is not empty either. P then sends the limit wedge $A \leftarrow A \times B \rightarrow B$ to $1 \leftarrow 1 \rightarrow 1$, and it is obvious that any other wedge $1 \leftarrow L \rightarrow 1$ factors uniquely through the latter. So P sends non-empty products to products.

If A is the empty set and B isn't, $A \times B$ is the empty set too. Then P sends the limit wedge $A \leftarrow A \times B \rightarrow B$ to $\emptyset \leftarrow \emptyset \rightarrow 1$. Since the only arrows in \mathbf{Set} with the empty set as target have the empty set as source, the only wedges $\emptyset \leftarrow L \rightarrow 1$ have $L = \emptyset$, so trivially factor uniquely through $\emptyset \leftarrow \emptyset \rightarrow 1$. So P sends products of the empty set with non-empty sets to products.

Likewise, of course, for products of non-empty sets with the empty set, and the product of the empty set with itself. So, taking all the cases together, P sends products to products.

Now consider the equalizer in \mathbf{Set} of two different maps $1 \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} 2$, where 2 is a two-membered set. Since f and g never agree, their equalizer is the empty set (with the empty inclusion map). But since P sends both the maps f and g to the identity map on 1, the equalizer of Pf and Pg is the set 1 (with the identity map). Which means that the equalizer of $P(f)$ and $P(g)$ is *not* the result of applying P to the equalizer of f and g .

- (3) Take the functor $Q: \mathbf{Set} \rightarrow \mathbf{Set}$ which sends any set X to the set $X \times 2$, and sends any arrow $f: X \rightarrow Y$ to $f \times 1_2: X \times 2 \rightarrow Y \times 2$ (the latter is

of course the function which acts on a pair $\langle x, n \rangle \in X \times 2$ by sending it to $\langle fx, n \rangle$. Claim: Q preserves equalizers but not binary products.

Concerning products, if a functor F preserves binary products in \mathbf{Set} , then by definition $F(X \times Y) \cong FX \times FY$. However, for X, Y finite, we have $Q(X \times Y) = (X \times Y) \times 2 \not\cong (X \times 2) \times (Y \times 2) = QX \times QY$.

Now note that the equalizer of parallel arrows $X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$ is essentially E , the subset of X on which f and g take the same value. And the equalizer of the parallel arrows $QX \begin{array}{c} \xrightarrow{Qf} \\ \xrightarrow{Qg} \end{array} QY$ is the subset of $X \times 2$ on which $f \times 1_2$ and $g \times 1_2$ take the same value, which will be $E \times 2$, i.e. QE . So indeed Q preserves equalizers.

Moral, to repeat: a functor may preserve some but not all limits. Preservation isn't in general an all or nothing business.

(c) Now for an example of a functor that does preserve all limits:

- (4) The forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ sends a terminal object in \mathbf{Mon} , a one-object monoid, to its underlying singleton set, which is terminal in \mathbf{Set} . So F preserves limits of the empty shape.

The same functor sends a product $(M, \cdot) \times (N, *)$ in \mathbf{Mon} to its underlying set of pairs of objects from M and N , which is a product in \mathbf{Set} . So the forgetful F also preserves limits of the shape of the discrete two object category.

Likewise for equalizers. As we saw in §9.1, Ex. (2), the equalizer of two parallel monoid homomorphisms $(M, \cdot) \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} (N, *)$ is (E, \cdot) equipped with the inclusion map $E \rightarrow M$, where E is the set on which f and g agree. Which means that the forgetful functor takes the equalizer of f and g as monoid homomorphisms to their equalizer as set functions. So F preserves equalizers.

So the forgetful $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ preserves terminal objects, binary products and equalizers – and hence, by appeal to the next theorem – this forgetful functor in fact preserves all finite limits.

At the last step we appeal to the fact that \mathbf{Mon} is a finitely complete category, together with the following theorem:

Theorem 94. *If \mathcal{C} is finitely complete, and a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves terminal objects, binary products and equalizers, then F preserves all finite limits.*

Proof. Suppose \mathcal{C} is finitely complete. Then any limit cone $[C, c_J]$ over a diagram $D: J \rightarrow \mathcal{C}$ is uniquely isomorphic to some limit cone $[C', c'_J]$ constructed from equalizers and finite products (see the proof of Theorem 58). Since F preserves terminal objects, binary products and equalizers, it sends the construction for

Functors and limits

$[C', c'_J]$ to a construction for a limit cone $[FC', Fc'_J]$ over $F \circ D: \mathbf{J} \rightarrow \mathcal{D}$. But F preserves isomorphisms, so $[FC, Fc_J]$ will be isomorphic to $[FC', Fc'_J]$ and hence is also a limit cone over $F \circ D: \mathbf{J} \rightarrow \mathcal{D}$. \square

(d) Note that by contrast, however, the same forgetful $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ does *not* preserve colimits with the ‘shape’ of the empty category, i.e. initial objects. For a one-object monoid is initial in \mathbf{Mon} but its underlying singleton set is not initial in \mathbf{Set} .

F does not preserve coproducts either – essentially because coproducts in \mathbf{Mon} can be larger than coproducts in \mathbf{Set} . Recall our discussion in §7.7 of coproducts in \mathbf{Grp} : similarly, $F(M \oplus N)$, the underlying set of a coproduct of monoids M and N , is (isomorphic to) the set of finite sequences of alternating non-identity elements from M and N . Contrast $FM \oplus FN$, which is just the disjoint union of the underlying sets.

Our example generalizes, by the way. A forgetful functor from a category of structured sets to \mathbf{Set} typically preserves finite limits but does not preserve all colimits.

(e) For the moment, we will finish on limit-preservation with a simple little result that we’ll need to appeal to later:

Theorem 95. *If the functor $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves pullbacks it preserves monomorphisms (i.e. sends monos to monos). Dually, if F preserves pushouts it preserves epimorphisms.*

Proof. We need only prove the first part. By Theorem 52, if $f: X \rightarrow Y$ in \mathcal{C} is monic then it is part of the pullback square on the left:

$$\begin{array}{ccc} X & \xrightarrow{1_X} & X \\ \downarrow 1_X & & \downarrow f \\ X & \xrightarrow{f} & Y \end{array} \Rightarrow \begin{array}{ccc} FX & \xrightarrow{1_{FX}} & FX \\ \downarrow 1_{FX} & & \downarrow Ff \\ FX & \xrightarrow{Ff} & FY \end{array}$$

By assumption F sends a pullback squares to pullback squares, so the square on the right is also a pullback square. So by Theorem 52 again, Ff is monic too. \square

17.3 Reflecting limits

(a) Here’s a companion definition to set alongside the definition of preserving limits, together with a couple of general theorems:

Definition 93. A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ *reflects limits of shape* \mathbf{J} iff, given a cone $[C, c_J]$ over a diagram $D: \mathbf{J} \rightarrow \mathcal{C}$, then if $[FC, Fc_J]$ is a limit cone over $F \circ D: \mathbf{J} \rightarrow \mathcal{D}$, $[C, c_J]$ is already a limit cone over D .

Reflecting colimits is defined dually. △

Theorem 96. *Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ is fully faithful. Then F reflects limits.*

Proof. Suppose $[C, c_J]$ is a cone over a diagram $D: J \rightarrow \mathcal{C}$, and $[FC, Fc_J]$ is a limit cone over $F \circ D: J \rightarrow \mathcal{D}$. We need to show that $[C, c_J]$ must already be a limit cone too.

Now take any other cone $[B, b_J]$ over D . F sends this to a cone $[FB, Fb_J]$ which must uniquely factor through the limit cone $[FC, Fc_J]$ via some $u: FB \rightarrow FC$ which makes $Fb_J = Fc_J \circ u$ for each $J \in J$. Since F is full and faithful, $u = Fv$ for some unique $v: B \rightarrow C$ such that $b_J = c_J \circ v$ for each J . So $[B, b_J]$ factors uniquely through $[C, c_J]$. Which shows that $[C, c_J]$ is a limit cone. □

Theorem 97. *Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves limits. Then if \mathcal{C} is complete and F reflects isomorphisms, then F reflects small limits.*

Proof. Since \mathcal{C} is complete there exists a limit cone $[B, b_J]$ over any diagram $D: J \rightarrow \mathcal{C}$ (where J is small), and so – since F preserves limits – $[FB, Fb]$ is a limit cone over $F \circ D: J \rightarrow \mathcal{D}$.

Now suppose that there is a cone $[C, c_J]$ over D such that $[FC, Fc_J]$ is another limit cone over $F \circ D$. Now $[C, c_J]$ must uniquely factor through $[B, b_J]$ via a map $f: C \rightarrow B$. Which means that $[FC, Fc_J]$ factors through $[FB, Fb]$ via Ff . However, since these are by hypothesis both limit cones over $F \circ D$, Ff must be an isomorphism. Hence, since F reflects isomorphisms, f must be an isomorphism. So $[C, c_J]$ must be a limit cone by Theorem 48. □

(b) Since the forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ preserves limits and reflects isomorphisms the last theorem shows that

- (1) The forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ reflects all limits. Similarly for some other forgetful functors from familiar categories of structured sets to \mathbf{Set} .

However, be careful! For we also have ...

- (2) The forgetful functor $F: \mathbf{Top} \rightarrow \mathbf{Set}$ which sends topological space to its underlying set *preserves* all limits but does not *reflect* all limits.

Here's a case involving binary products. Suppose X and Y are a couple of spaces with a coarse topology, and let Z be the space $FX \times FY$ equipped with a finer topology. Then, with the obvious arrows, $X \leftarrow Z \rightarrow Y$ is a wedge to X, Y but not the limit wedge in \mathbf{Top} : but $FX \leftarrow FX \times FY \rightarrow FY$ is a limit wedge in \mathbf{Set} .

Given the previous theorem, we can conclude that $F: \mathbf{Top} \rightarrow \mathbf{Set}$ doesn't reflect isomorphisms. (Which is also something we can show directly. Consider the continuous bijection from the half-open interval $[0, 1)$ to S^1 . Think of this bijection as a topological map f ; then f is not a homeomorphism in \mathbf{Top} . However, treating the bijection as a set-function, i.e. as $F'f$, it *is* an isomorphism in \mathbf{Set} .)

17.4 Creating limits

Alongside the natural notions of preserving and reflecting limits, we meet a related third notion which we should pause to explain:

Definition 94. A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ *creates limits of shape* J iff, for any diagram $D: J \rightarrow \mathcal{C}$, if $[M, m_J]$ is a limit cone over $F \circ D$, there is a unique cone $[C, c_J]$ over D such that $[FC, Fc_J] = [M, m_J]$, and moreover $[C, c_J]$ is a limit cone.

Creating colimits is defined dually. △

(Variant: some define creation of limits by only requiring that $[FC, Fc_J]$ is isomorphic to $[M, m_J]$ in the obvious sense.)

Why ‘creation’? The picture is that every limit over $F \circ D$ in \mathcal{D} is generated by F from a unique limit over D in \mathcal{C} . And while reflection is a condition on those limit cones over $F \circ D$ which take the form $[FC, Fc_J]$ for some cone $[C, c_J]$, creation is a similar condition on *any* limit cone over $F \circ D$. So as you would predict,

Theorem 98. *If the functor $F: \mathcal{C} \rightarrow \mathcal{D}$ creates limits of shape J , it reflects them.*

Proof. Suppose $[FC, Fc_J]$ is a limit cone over $F \circ D: J \rightarrow \mathcal{C}$ generated by the cone $[C, c_J]$ over D . Then, assuming F creates limits, $[C, c_J]$ has to be the unique cone over D such that $[FC, Fc_J]$ is generated by it, and has to be a limit cone. □

Theorem 99. *Suppose $F: \mathcal{C} \rightarrow \mathcal{D}$ is a functor, that \mathcal{D} has limits of shape J and F creates such limits. Then \mathcal{C} has limits of shape J and F preserves them.*

Proof. Take any diagram $D: J \rightarrow \mathcal{C}$. Then there is a limit $[M, m_J]$ over $F \circ D$ (since \mathcal{D} has all limits of shape J). Hence (since F creates limits), there is a limit cone $[C, c_J]$ over D where this is such that $[FC, Fc_J]$ is $[M, m_J]$ and hence is a limit cone too. □

18 Hom-functors

This chapter introduces the notion of a hom-functor, a type of functor which will turn out to play a rather special role in category theory. We show that, unlike the general run of functors, hom-functors do behave very nicely with (small) limits, always preserving them.

18.1 Hom-sets

(a) Suppose the category \mathcal{C} is locally small. Then there is only a set's worth of arrows between any two \mathcal{C} -objects. Moreover, in many familiar locally small categories, these \mathcal{C} -arrows will be an appropriate kind of homomorphism. So this explains the terminology in the following conventional definition:

Definition 95. Given a locally small category \mathcal{C} , and \mathcal{C} -objects A and B , then the *hom-set* $\mathcal{C}(A, B)$ is the set of \mathcal{C} -arrows from A to B . \triangle

The brusque but conventional notation we are using for collections of arrows between two objects has already made a fleeting appearance in §16.6: alternative and perhaps more reader-friendly notations are ‘ $\text{Hom}_{\mathcal{C}}(A, B)$ ’ or just ‘ $\text{Hom}(A, B)$ ’ when the relevant category is obvious.

(b) But although our definition is absolutely standard, it is not unproblematic. What kind of set is a hom-set? In categorical terms, in which category does a hom-set $\mathcal{C}(A, B)$ live? (We here return to a question already flagged-up in §16.4.)

The usual assumption, very often made with no comment at all, is that a hom-set lives in the category **Set**. “Where else?”, you might reasonably ask. But what category is **Set**? Remember, we didn’t fix this at the outset. We cheerfully said, just take your favourite universe of sets and functions between them, and the category **Set** can for now comprise *them*. But suppose – naturally enough – that you think of **Set** as containing just the sets you know and love from your basic set theory course in the delights of ZFC. In this case, **Set** is a category of *pure* sets, i.e. of sets whose members, if any, are sets whose members, if any, are sets ... all the way down. But if we think of $\mathcal{C}(A, B)$ as living in such a category of pure sets, then the arrows which are members of $\mathcal{C}(A, B)$ will themselves have

to be pure sets too. Yet do we really want to suppose that categorial arrows are inevitably just more sets?

It seems that we have at least three options here. In headline terms, we can for a start . . .

- (i) Bite the bullet. Take \mathbf{Set} to be a category of pure sets, and take $\mathcal{C}(A, B)$ to be a pure set living in \mathbf{Set} . Then \mathcal{C} -arrows themselves have to be pure sets.
- (ii) Backtrack. Take \mathbf{Set} after all to be a category of possibly impure sets, where the non-set elements can, inter alia, be arrows in any category. So again we can endorse the standard view that $\mathcal{C}(A, B)$ lives in \mathbf{Set} , but now without pre-supposing that all \mathcal{C} -arrows are sets.
- (iii) Re-interpret. As in (i), take \mathbf{Set} to be a category of pure sets. As in (ii), regard $\mathcal{C}(A, B)$ as, in general, an impure collection whose members are arrows (which needn't be themselves sets). But then we'll have to re-interpret the standard line that $\mathcal{C}(A, B)$ lives in \mathbf{Set} . We will say it isn't strictly speaking the hom-set as originally defined which lives in \mathbf{Set} but rather a pure set which represents or models or indexes it (that there can be such an indexing set is what we mean when we say that there is only a set's-worth of arrows in $\mathcal{C}(A, B)$).

We could even call this representing pure set $\mathcal{C}(A, B)$ too, with context deciding when we are talking about the 'true' impure hom-collection and when we are talking about its pure-set representation.

It is, to say the least, not entirely clear at the outset which of these options is the best way forward (or maybe we should be looking for a fourth way!).

Option (i) has weighty support. In his canonical 1997, Saunders Mac Lane initially gives a definition like our Defn. 4 as a definition of what *he* calls meta-categories, and then for him a category proper "will mean any interpretation of the category axioms within set theory". So for Mac Lane, at least at the outset, all the gadgets of categories proper will unproblematically live in the universe of set theory, and that applies to hom-sets in particular. Presumably this is the standard universe of pure sets. Mac Lane doesn't, I think, make that explicit: but e.g. Horst Schubert does in §3.1 of his terse but very lucid (1972), writing "One has to be aware that the set theory used here has no 'primitive (ur-)elements'; elements of sets . . . are always themselves sets." But, as we asked before, do we really want or need to suppose that categories are always and everywhere sets? Not if (as some do) we want to conceive of category theory as a more democratic way of organizing the mathematical universe, which provides an alternative to imperialistic set-theoretic reductionism. (Indeed, much later in his book, in the Appendix, Mac Lane suggests that we can perhaps after all use our Defn. 4, more or less, to describe categories directly, without going via set theory).

Option (ii), by contrast, avoids reducing everything to pure sets. But on the face of it, it is now quite unclear what *does* live in the universe of \mathbf{Set} , if it is just

a free-for-all at the level of urelements, and it is sheer mess at the bottom level of the hierarchy of sets. But maybe there is an option (ii') where we re-think our story about the nature of sets in a way which still in some sense allows urelements but abstracts away from their nature. More about this in due course.

Option (iii) might seem to let us have our cake and eat it – we keep **Set** as a tidy category of pure sets without urelements, we keep collections of arrows as impure sets, and we model one by the other in a familiar enough way. But it adds a layer of complication which might not be welcome.

We won't try to judge which is the best option at this point. And after all, such verdicts are often best given rather late in the game, when we can look back to see what really are the essential requirements of the load-bearing parts of the theory we have been developing. So what to do? For the moment, we will take the path of least resistance and proceed conventionally, *as if* hom-sets do live in **Set**; and we'll have to return later to think more about how we really want to construe this.

18.2 Hom-functors

(a) Now to introduce the main notion of this chapter.

Assume \mathcal{C} is locally small. So we can talk about $\mathcal{C}(A, B)$, the hom-set of \mathcal{C} -arrows from A to B . Keep A fixed. Then as we vary X through the objects in \mathcal{C} , we get varying $\mathcal{C}(A, X)$.

So: consider the resulting function which sends an object X in \mathcal{C} to the set $\mathcal{C}(A, X)$, a set which we are following standard practice in taking as living in **Set**.

Can we now treat this function on \mathcal{C} -objects as the first component of a functor, call it $\mathcal{C}(A, -)$, from \mathcal{C} to **Set**? Well, how could we find a component of the functor to deal with the \mathcal{C} -arrows? Such a component is going to need to send an arrow $f: X \rightarrow Y$ in \mathcal{C} to a **Set**-function from $\mathcal{C}(A, X)$ to $\mathcal{C}(A, Y)$. The obvious candidate for the latter function is the one we can notate as $f \circ -$ that maps any $g: A \rightarrow X$ in $\mathcal{C}(A, X)$ to $f \circ g: A \rightarrow Y$ in $\mathcal{C}(A, Y)$. (Note, $f \circ g: A \rightarrow Y$ has to be in $\mathcal{C}(A, Y)$ because \mathcal{C} is a category which by hypothesis contains $g: A \rightarrow X$ and $f: X \rightarrow Y$ and hence must contain their composition.)

It is easy to check that these components add up to a genuine covariant functor – in fact the functoriality in this case just reduces to the associativity of composition for arrows in a category and the basic laws for identity arrows.

Now, start again from the hom-set $\mathcal{C}(A, B)$ but this time keep B fixed: then as we vary X through the objects in \mathcal{C} , we again get varying hom-sets $\mathcal{C}(X, B)$. Which generates a function which sends an object X in \mathcal{C} to an object $\mathcal{C}(X, B)$ in **Set**. To turn *this* into a functor $\mathcal{C}(-, B)$, we need again to add a component to deal with \mathcal{C} -arrows. That will need to send $f: X \rightarrow Y$ in \mathcal{C} to some function between $\mathcal{C}(X, B)$ to $\mathcal{C}(Y, B)$. But this time, to get functions to compose properly, things will have to go the other way about, i.e. the associated functor will

Hom-functors

have to send a function $g: Y \rightarrow B$ in $\mathcal{C}(Y, B)$ to $g \circ f: X \rightarrow B$ in $\mathcal{C}(X, B)$. So this means that the resulting functor $\mathcal{C}(-, B)$ is a *contravariant* hom-functor.

(b) So, to summarize, we will say:

Definition 96. Given a locally small category \mathcal{C} , then the associated *covariant hom-functor* $\mathcal{C}(A, -): \mathcal{C} \rightarrow \mathbf{Set}$ is the functor with the following data:

- (1) A mapping $\mathcal{C}(A, -)_{ob}$ whose value at the object X in \mathcal{C} is the hom-set $\mathcal{C}(A, X)$.
- (2) A mapping $\mathcal{C}(A, -)_{arw}$, whose value at the \mathcal{C} -arrow $f: X \rightarrow Y$ is the set function $f \circ -$ from $\mathcal{C}(A, X)$ to $\mathcal{C}(A, Y)$ which sends an element $g: A \rightarrow X$ to $f \circ g: A \rightarrow Y$.

And the associated *contravariant hom-functor* $\mathcal{C}(-, B): \mathcal{C} \rightarrow \mathbf{Set}$ is the functor with the following data:

- (3) A mapping $\mathcal{C}(-, B)_{ob}$ whose value at the object X in \mathcal{C} is the hom-set $\mathcal{C}(X, B)$.
- (4) A mapping $\mathcal{C}(-, B)_{arw}$, whose value at the \mathcal{C} -arrow $f: Y \rightarrow X$ is the set function $- \circ f$ from $\mathcal{C}(X, B)$ to $\mathcal{C}(Y, B)$ which sends an element $g: X \rightarrow B$ to the map $g \circ f: Y \rightarrow B$.

The use of a blank in the notation ' $\mathcal{C}(A, -)$ ' invites an obvious shorthand: instead of writing ' $\mathcal{C}(A, -)_{arw}(f)$ ' to indicate the result of the component of the functor which acts on arrows applied to the function f , we will write simply ' $\mathcal{C}(A, f)$ '. Similarly for the dual. \triangle

Alternative notations for hom-functors, to along with the alternative notations for hom-sets, are ' $\text{Hom}_{\mathcal{C}}(A, -)$ ' and ' $\text{Hom}_{\mathcal{C}}(-, B)$ '.

(c) For the record, we can also define a related 'bi-functor' $\mathcal{C}(-, -): \mathcal{C}^{op} \times \mathcal{C} \rightarrow \mathbf{Set}$, which we can think of as contravariant in the first place and covariant in the second. This acts on the product category mapping the pair object (A, B) to the hom-set $\mathcal{C}(A, B)$, and the pair of morphisms $(f: X' \rightarrow X, g: Y \rightarrow Y')$ to the morphism between $\mathcal{C}(X, Y)$ and $\mathcal{C}(X', Y')$ that sends $h: X \rightarrow Y$ to $g \circ h \circ f: X' \rightarrow Y'$. We will return to this if/when we need to say more.

18.3 Hom-functors preserve limits

(a) As noted at the outset, hom-functors will play a special role in category theory, and we will meet them repeatedly. But in the rest of this chapter, we just consider how they interact with limits.

We start with a preliminary observation. If some functor F preserves products, it has to be the case that $F(C \times D) \cong FC \times FD$. So if a hom-functor $\mathcal{C}(A, -)$ is to preserve products, we need this to be true:

Theorem 100. *Assuming the product exists, $\mathcal{C}(A, C \times D) \cong \mathcal{C}(A, C) \times \mathcal{C}(A, D)$.*

However, this is easy to show:

Proof. An arrow $f: A \rightarrow C \times D$ factors into two arrows $c: A \rightarrow C$ and $d: A \rightarrow D$ via the projection arrows of the product $C \times D$. And two such arrows c, d form a wedge which factors uniquely through the product via f . This gives us a bijection between arrows f in $\mathcal{C}(A, C \times D)$ and pairs of arrows (c, d) in $\mathcal{C}(A, C) \times \mathcal{C}(A, D)$, an isomorphism in **Set**. \square

This observation can now be turned into a proof that hom-functors preserve any binary product which exists. They also preserve any terminal objects and equalizers. And then using the fact that if there is a limit cone over $D: \mathbf{J} \rightarrow \mathcal{C}$ (with \mathbf{J} a small category), then it can be constructed from suitable products and equalizers (as indicated by the proof of Theorem 60), we can derive

Theorem 101. *Suppose that \mathcal{C} is a small category. Then the covariant hom-functor $\mathcal{C}(A, -): \mathcal{C} \rightarrow \mathbf{Set}$, for any A in the category \mathcal{C} , preserves all small limits that exist in \mathcal{C} .*

However, rather than officially prove this important theorem in the way just sketched, let's instead go for a brute-force just-apply-the-definitions-and-see-what-happens demonstration (for it is quite a useful reality check to run through the details):

Proof. We'll first check that $\mathcal{C}(A, -): \mathcal{C} \rightarrow \mathbf{Set}$ sends a cone over the diagram $D: \mathbf{J} \rightarrow \mathcal{C}$ to a cone over $\mathcal{C}(A, -) \circ D: \mathbf{J} \rightarrow \mathbf{Set}$.

A cone has a vertex C , and arrows $c_J: C \rightarrow DJ$ for each $J \in \mathbf{J}$, where for any $f: J \rightarrow K$ in \mathbf{J} , so for any $Df: DJ \rightarrow DK$, $c_K = Df \circ c_J$.

Now, acting on objects, $\mathcal{C}(A, -)$ sends C to $\mathcal{C}(A, C)$ and sends DJ to $\mathcal{C}(A, DJ)$. And acting on arrows, $\mathcal{C}(A, -)$ sends $c_J: C \rightarrow DJ$ to the set function $c_J \circ -$ which takes $g: A \rightarrow C$ and outputs $c_J \circ g: A \rightarrow DJ$; and it sends $Df: DJ \rightarrow DK$ to the set-function $Df \circ -$ which takes $h: A \rightarrow DJ$ and outputs $Df \circ h: A \rightarrow DK$.

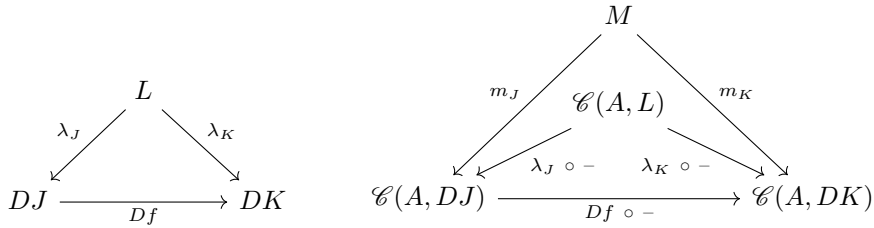
Diagrammatically, then, the functor sends the triangle on the left to the one on the right:

$$\begin{array}{ccc}
 \begin{array}{ccc}
 & C & \\
 c_J \swarrow & & \searrow c_K \\
 DJ & \xrightarrow{Df} & DK
 \end{array} & \Rightarrow & \begin{array}{ccc}
 & \mathcal{C}(A, C) & \\
 c_J \circ - \swarrow & & \searrow c_K \circ - \\
 \mathcal{C}(A, DJ) & \xrightarrow{Df \circ -} & \mathcal{C}(A, DK)
 \end{array}
 \end{array}$$

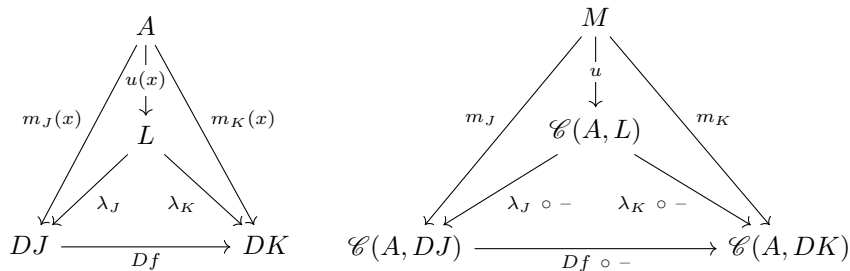
And assuming $c_K = Df \circ c_J$, we have $c_K \circ - = (Df \circ c_J) \circ - = (Df \circ -) \circ (c_J \circ -)$: hence, if the triangle on the left commutes, so does the triangle on the right. Likewise for other such triangles. Which means that if $[C, c_J]$ is a cone over D , then $[\mathcal{C}(A, C), c_J \circ -]$ is indeed a cone over $\mathcal{C}(A, -) \circ D$.

Hom-functors

So far, so good! It remains, then, to show that in particular $\mathcal{C}(A, -)$ sends limit cones to limit cones. So suppose that $[L, \lambda_J]$ is a limit cone in \mathcal{C} over D . The functor $\mathcal{C}(A, -) : \mathcal{C} \rightarrow \mathbf{Set}$ sends the left-hand commuting diagram below to the commuting triangle at the bottom of the right-hand diagram. And we now suppose that $[M, m_j]$ is any other cone over the image of D :



Hence $m_K = (Df \circ -) \circ m_J$. Now remember that M lives in \mathbf{Set} : so take a member x . Then $m_J(x)$ is a particular arrow in $\mathcal{C}(A, DJ)$, in other words $m_J(x): A \rightarrow DJ$. Likewise we have $m_K(x): A \rightarrow DK$. But $m_K(x) = Df \circ m_J(x)$. Which means that for all f the outer triangles on the left below commute and so $[A, m_J(x)]$ is a cone over D . And this must factor uniquely through an arrow $u(x)$ as follows:



Hence $u(x)$ is an arrow from A to L , i.e. an element of $\mathcal{C}(A, L)$. So consider the map $u: M \rightarrow \mathcal{C}(A, L)$ which sends x to $u(x)$. Since $m_J(x) = \lambda_J \circ u(x)$ for each x , $m_J = (\lambda_J \circ -) \circ u$. And since this applies for each J , So $[M, m_j]$ factors through the image of the cone $[L, \lambda_J]$ via u .

Suppose there is another map $v: M \rightarrow \mathcal{C}(A, L)$ such that we also have each $m_J = (\lambda_J \circ -) \circ v$. Then again take an element $x \in M$: then $m_J(x) = \lambda_J \circ v(x)$. So again, $[A, m_J(x)]$ factorizes through $[L, \lambda_J]$ via $v(x)$ – which, by the uniqueness of factorization through limits, means that $v(x) = u(x)$. Since that obtains for all $x \in M$, $v = u$. Hence $[M, m_j]$ factors uniquely through the image of $[L, \lambda_J]$. Since $[M, m_j]$ was an arbitrary cone, we have therefore proved that the image of the limit cone $[L, \lambda_J]$ is also a limit cone. \square

(b) What is the dual of Theorem 101? We have two dualities to play with: limits vs colimits and covariant functors vs contravariant functors.

Two initial observations. First, a covariant hom-functor need not preserve colimits. For example, take the hom-functor $\mathbf{Grp}(A, -)$. In \mathbf{Grp} the initial object 0 is also the terminal object, so for any group A , $\mathbf{Grp}(A, 0)$ is a singleton, which is not initial in \mathbf{Set} . Second, contravariant hom-functors can't preserve either limits or colimits, because contravariant functors reverse arrows.

So the dual result we want is this:

Theorem 102. *Suppose that \mathcal{C} is a small category. Then the contravariant hom-functor $\mathcal{C}(-, A): \mathcal{C} \rightarrow \mathbf{Set}$, for any A in the category \mathcal{C} , sends a colimit of shape J (for small category J) to a limit of that shape.*

Yes, that's right: contravariant functors send colimits to limits (the two reversals of arrows involved in going from covariant to contravariant, and from limit to colimit, cancelling out). We can leave the proof as an exercise in dualizing.

19 Functors and comma categories

We have now introduced the notion of a functor as a map between categories, and seen how functors can e.g. preserve/reflect (or fail to preserve/reflect) various properties of arrows and various limit constructions. And we are about to move on to introduce the next Big Idea, i.e. the notion of maps between functors.

However, before we do that, this chapter pauses to use the notion of a functor to define the idea of a comma category. I'm afraid that this might initially seem to involve a rather contorted construction. But bear with me! We will in fact be repeatedly meeting instances of comma categories, so we ought to get to grips with this idea sooner or later.

19.1 Functors and slice categories

By way of a warm-up exercise, recall the notion of a slice category \mathcal{C}/I (Defn. 18). If \mathcal{C} is a category, and I is a \mathcal{C} -object, then \mathcal{C}/I 's objects, economically defined, are the arrows $f: A \rightarrow I$ (for any \mathcal{C} -object A), while \mathcal{C}/I 's arrows between these objects $f: A \rightarrow I$ and $g: B \rightarrow I$ are the arrows $j: A \rightarrow B$ such that $g \circ j = f$.

Here, then, are a couple of simple examples of functors operating on slice categories:

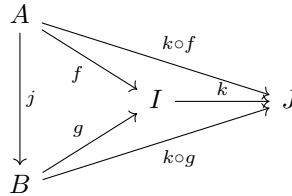
- (1) There is functor, another kind of forgetful functor, $F: \mathcal{C}/I \rightarrow \mathcal{C}$, which sends a \mathcal{C}/I -object $f: A \rightarrow I$ back to A , and sends an arrow j in \mathcal{C}/I back to the original arrow j in \mathcal{C} .

For example, recall the slice category \mathbf{FinSet}/I_n which we met at the end of §4.3, which is the category of finite sets whose members are coloured from a palette of n colours. The forgetful functor $F: \mathbf{FinSet}/I_n \rightarrow \mathbf{FinSet}$ forgets about the colourings of a set S provided by functions $f: S \rightarrow I_n$.

- (2) Next, let's show how we can use an arrow $k: I \rightarrow J$ (for $I, J \in \mathcal{C}$) to generate a corresponding functor $K: \mathcal{C}/I \rightarrow \mathcal{C}/J$.

The functor needs to act on *objects* in \mathcal{C}/I and send them to objects in \mathcal{C}/J . That is to say, K_{ob} needs to send an arrow $f: X \rightarrow I$ to an arrow f' with codomain J . The obvious thing to do is to put $K_{ob}(f) = k \circ f$.

And how will a matching K_{arw} act on *arrows* of \mathcal{C}/I ? Consider:



Here, the \mathcal{C}/I -arrows from $f: A \rightarrow I$ to $g: B \rightarrow I$, by definition, include any j which makes the left-hand inner triangle commute. But then such a j will also make the outer triangle commute, i.e. j is an arrow from $k \circ f: A \rightarrow J$ to $k \circ g: B \rightarrow J$ (which is therefore an arrow from $K(f)$ to $K(g)$).

So we can simply put $K(j)$ (for $j: f \rightarrow g$ in \mathcal{C}/I) to be j (i.e. $j: K(f) \rightarrow K(g)$ in \mathcal{C}/J).

Claim: K is then a functor from \mathcal{C}/I to \mathcal{C}/J .

It is a useful small reality check to confirm that (2) all makes sense, and that K is indeed a functor.

19.2 Comma categories

We have already met various ways of getting new categories from old, including the one we've just reminded ourselves about, namely constructing slice categories. Given that we now have the notion of a functor to hand, in this section we can introduce another way of defining new from old, this time deriving a category from three(!) categories and a pair of functors relating them.

Suppose, then, that we have a pair of functors sharing a target, say $S: \mathcal{A} \rightarrow \mathcal{C}$ and $T: \mathcal{B} \rightarrow \mathcal{C}$. Then we have a way of indirectly connecting an object A in \mathcal{A} to an object B in \mathcal{B} , i.e. by looking at their respective images SA and TB and considering arrows $f: SA \rightarrow TB$ between them.

We are going to define a category of such connections. But if its objects are to comprise an \mathcal{A} -object A , a \mathcal{B} -object B , together with a \mathcal{C} -arrow $f: SA \rightarrow TB$, what could be the arrows in our new category? Suppose we have, then, two triples (A, f, B) , (A', f', B') ; an arrow between them will presumably involve arrows $a: A \rightarrow A'$ and $b: B \rightarrow B'$. But note that these two are sent respectively to arrows $Sa: SA \rightarrow SA'$ and $Tb: TB \rightarrow TB'$ in \mathcal{C} , and we will need these arrows to interact appropriately with the other \mathcal{C} -arrows f and f' .

All that prompts the following – seemingly rather esoteric – definition:

Definition 97. Given functors $S: \mathcal{A} \rightarrow \mathcal{C}$ and $T: \mathcal{B} \rightarrow \mathcal{C}$, then the ‘comma category’ $(S \downarrow T)$ is the category with the following data:

- (1) The objects of $(S \downarrow T)$ are triples (A, f, B) where A is an \mathcal{A} -object, B is a \mathcal{B} -object, and $f: SA \rightarrow TB$ is an arrow in \mathcal{C} .

- (2) An arrow of $(S \downarrow T)$ from (A, f, B) to (A', f', B') is a pair (a, b) , where $a: A \rightarrow A'$ is an \mathcal{A} -arrow, $b: B \rightarrow B'$ is an \mathcal{B} -arrow, and the following diagram commutes:

$$\begin{array}{ccc} SA & \xrightarrow{f} & TB \\ \downarrow Sa & & \downarrow Tb \\ SA' & \xrightarrow{f'} & TB' \end{array}$$

- (3) The identity arrow on the object (A, f, B) is the pair $(1_A, 1_B)$.
 (4) Composition in $(S \downarrow T)$ is induced by the composition laws of \mathcal{A} and \mathcal{B} , thus: $(a', b') \circ (a, b) = (a' \circ_{\mathcal{A}} a, b' \circ_{\mathcal{B}} b)$. \triangle

It is readily seen that, so defined, $(S \downarrow T)$ is indeed a category.

The standard label ‘comma category’ arises from an unhappy earlier notation ‘ (S, T) ’: the notation has long been abandoned but the name has stuck. But why we should be bothering with such a construction? Well, the notion of a comma category in fact nicely generalizes a number of simpler constructions. And indeed, we have already met two comma categories in thin disguise. The next section reveals which they are.

19.3 Two (already familiar) types of comma category

- (a) First take the minimal case where $\mathcal{A} = \mathcal{B} = \mathcal{C}$, and where both S and T are the identity functor on that category, $1_{\mathcal{C}}$.

Then the objects in this category $(1_{\mathcal{C}} \downarrow 1_{\mathcal{C}})$ are triples $(X, X \xrightarrow{f} Y, Y)$ for X, Y both \mathcal{C} -objects. And an arrow from $(X, X \xrightarrow{f} Y, Y)$ to $(X', X' \xrightarrow{f'} Y', Y')$ is a pair of \mathcal{C} -arrows $a: X \rightarrow X'$, $b: Y \rightarrow Y'$ such that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \downarrow a & & \downarrow b \\ X' & \xrightarrow{f'} & Y' \end{array}$$

So the only difference between $(1_{\mathcal{C}} \downarrow 1_{\mathcal{C}})$ and the arrow category $\mathcal{C}^{\rightarrow}$ is that we have now ‘decorated’ the objects of $\mathcal{C}^{\rightarrow}$, i.e. \mathcal{C} -arrows $f: X \rightarrow Y$, with explicit assignments of their sources and targets as \mathcal{C} -arrows, to give triples $(X, X \xrightarrow{f} Y, Y)$. Hence $(1_{\mathcal{C}} \downarrow 1_{\mathcal{C}})$ and $\mathcal{C}^{\rightarrow}$, although not strictly identical, come to the just same.

And of course, we can do better than limply say the two categories ‘come to just the same’. Working in a big enough category CAT , consider the functor $F: \mathcal{C}^{\rightarrow} \rightarrow (1_{\mathcal{C}} \downarrow 1_{\mathcal{C}})$ which sends a $\mathcal{C}^{\rightarrow}$ -object to the corresponding triple, and

19.4 Another (new) type of comma category

sends $\mathcal{C} \rightarrow$ -arrows (pairs of \mathcal{C} -arrows) to themselves. Then, F trivially has an inverse, and so the categories are isomorphic.

(b) Let's take secondly the special case where $\mathcal{A} = \mathcal{C}$ with S the identity functor $1_{\mathcal{C}}$, and where $\mathcal{B} = \mathbf{1}$ (the category with a single object \star and the single arrow 1_{\star}). And take the functor $I: \mathbf{1} \rightarrow \mathcal{C}$ which sends \star to some individual \mathcal{C} -object which we'll also call I – see §15.2, Ex. (F11).

Applying the definition, the objects of the category $(1_{\mathcal{C}} \downarrow I)$ are therefore triples $(A, A \xrightarrow{f} I, \star)$, and an arrow between $(A, A \xrightarrow{f} I, \star)$ and $(B, B \xrightarrow{g} I, \star)$ will be a pair $(j, 1_{\star})$, with $j: A \rightarrow B$ an arrow such the diagram on the left commutes:

$$\begin{array}{ccc}
 A & \xrightarrow{f} & I \\
 \downarrow j & & \downarrow 1_I \\
 B & \xrightarrow{g} & I
 \end{array}
 \qquad
 \begin{array}{ccc}
 A & & I \\
 \downarrow j & \searrow f & \\
 B & \nearrow g & I
 \end{array}$$

The diagram on the left is trivially equivalent to that on the right – which should look very familiar! We've ended up with something tantamount to the slice category \mathcal{C}/I , the only differences being that (i) instead of the slice category's objects, i.e. pairs (A, f) , we now have 'decorated' objects (A, f, \star) which correspond one-to-one with them, and (ii) instead of the slice category's arrows $j: A \rightarrow B$ we have decorated arrows $(j, 1_{\star})$ which correspond one-to-one with them.

Again the categories $(1_{\mathcal{C}} \downarrow I)$ and \mathcal{C}/I are evidently isomorphic categories.

19.4 Another (new) type of comma category

(a) While are we looking at examples of comma categories, let's add for the record a third illustrative case (pretty similar to the case of slice categories). It will turn out to be useful, and we choose notation with an eye to a later application.

Suppose we have a functor $G: \mathcal{C} \rightarrow \mathcal{A}$ and an object $A \in \mathcal{A}$. There is a corresponding functor $A: \mathbf{1} \rightarrow \mathcal{A}$ (which sends the sole object \star in the one-object category $\mathbf{1}$ to the object A in \mathcal{A}). Then what is the comma category $(A \downarrow G)$? Flat-footedly applying the definitions, we get:

- (1) The objects of $(A \downarrow G)$ are triples (\star, f, C) where C is a \mathcal{C} -object, and $f: A \rightarrow GC$ is an arrow in \mathcal{A} .
- (2) An arrow of $(A \downarrow G)$ from (\star, f, C) to (\star, f', C') is a pair of arrows, $(1_{\star}, j)$ with $j: C \rightarrow C'$ such the following square commutes:

$$\begin{array}{ccc} A & \xrightarrow{f} & GC \\ \downarrow 1_A & & \downarrow Gj \\ A & \xrightarrow{f'} & GC' \end{array}$$

However, since the \star -component in all the objects of $(A \downarrow G)$ is doing no real work, our comma category is tantamount to the stripped-down category such that

- (1') the objects are, more simply, pairs (C, f) where C is a \mathcal{C} -object and $f : A \rightarrow GC$ is an arrow in \mathcal{A} ,
- (2') an arrow from (C, f) to (C', f') is, more simply, a \mathcal{C} -arrow $j : C \rightarrow C'$ making this commute:

$$\begin{array}{ccc} & & GC \\ & \nearrow f & \\ A & & \downarrow Gj \\ & \searrow f' & \\ & & GC' \end{array}$$

We add, of course, the obvious definitions for the identity arrows and for composition of arrows. And it is this stripped-down version which is in fact usually referred to by the label ' $(A \downarrow G)$ ' (we can, incidentally, read ' A ' in the label here as just referring to an object, not to the corresponding functor).

- (b) Similarly, there is a category $(G \downarrow A)$. In its stripped down version,
 - (1'') its objects are pairs (C, f) where C is a \mathcal{C} -object and $f : GC \rightarrow A$ is an arrow in \mathcal{A} ,
 - (2'') an arrow from (C, f) to (C', f') is a \mathcal{C} -arrow $j : C \rightarrow C'$ making this commute:

$$\begin{array}{ccc} GC & & \\ \downarrow Gj & \nearrow f & \\ GC' & & A \\ & \searrow f' & \end{array}$$

19.5 An application: free monoids again

We make a connection between the idea of a *free monoid* (which we met in §15.5) and the idea of a certain *comma category* (of the kind we met in the last section).

Take the two categories **Mon** and **Set**; let S be a set living in **Set, and let $F : \mathbf{Mon} \rightarrow \mathbf{Set}$ be the forgetful functor. And now consider the comma category $(S \downarrow F)$. Unthinkingly applying the definition,**

- (1) the objects of this category $(S \downarrow F)$ are pairs (\mathcal{N}, f) where \mathcal{N} is a monoid $(N, \cdot, 1_N)$ and f is a set-function from S to $F(\mathcal{N})$, i.e. $f: S \rightarrow N$;
- (2) an $(S \downarrow F)$ -arrow from (\mathcal{N}, f) to (\mathcal{N}', f') is a monoid homomorphism $j: \mathcal{N} \rightarrow \mathcal{N}'$, which treated as a set-function is $j = Fj: N \rightarrow N'$, such that $\bar{j}' = j \circ f$.

But what does this mean, intuitively? We can think of a function $f: S \rightarrow N$ as *labelling* elements of N by members of S : N -elements can thereby receive zero, one, or many labels. So we can think of a pair (\mathcal{N}, f) as a monoid with some S -labelled elements. And an arrow between these monoids-with- S -labelled-elements is a monoid homomorphism which sends labelled elements to elements with the same label(s).

Now suppose $(S \downarrow F)$ has an initial object (\mathcal{M}, g) . This is a monoid \mathcal{M} with some elements labelled by $g: S \rightarrow M$ such that for *any* monoid \mathcal{N} with S -labelled elements, there is a unique monoid homomorphism from \mathcal{M} to \mathcal{N} that preserves labels.

Since some labelled monoids have no objects with multiple labels, it follows that g also can't give the same object multiple labels. In other words, g is injective. Hence, without loss of generality, simply by swapping objects around, we can in fact choose \mathcal{M} so that g is an inclusion.

So the situation is as follows. We can think of the monoid \mathcal{M} as having objects M including the selected set S . And this monoid is such that, for any other monoid \mathcal{N} and set-function $f: S \rightarrow N$, there is a *unique* homomorphism from \mathcal{M} to \mathcal{N} which sends members of S to their images under f .

A moment's reflection shows that \mathcal{M} must be a free monoid with generators S , in the sense we initially characterized in §15.5. In other words, its objects M include a unit element, the members of S , all their possible products, products of products, etc., with no unnecessary identities between these elements, and with nothing else. Why so? Here's the argument:

1. Just because \mathcal{M} is a monoid, it must contain a unit element, the members of S , all their possible products, products of products, and so on.
2. Suppose there were some unnecessary identity between two of those elements. Then take a monoid \mathcal{M}' with the same generators (and the same labelling function g) but without that identity. Then a homomorphism from \mathcal{M} to \mathcal{M}' respecting labels will send generators to generators, and (being a homomorphism), will send their products to products, so enforcing the same identity to recur in \mathcal{M}' contrary to hypothesis.
3. Suppose there were extra elements in \mathcal{M} not generated from the unit and members of S . Then there could evidently be multiple homomorphisms from \mathcal{M} to other monoids respecting labelled objects and their products but dealing with the 'junk' differently.

Which all goes to motivate an official categorial definition of the notion we previously only informally characterized:

Definition 98. A free monoid over the set S is an initial object of the comma category $(S \downarrow F)$, where $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ is the forgetful functor. \triangle

So here's another notion that we have defined in terms of a universal mapping property.

We should check that this tallies with our discussion back in §15.5:

Theorem 103. Take the monoid $\mathcal{L} = (List(S), \cap, 1)$ and equip it with the function $g: S \rightarrow List(S)$ which sends an element s of S to the list with just that element. Then (\mathcal{L}, g) is a free monoid over S .

Proof. Suppose \mathcal{N} is a monoid $(N, \cdot, 1_N)$ and $f: S \rightarrow N$ is a set function. We need to show that there is a unique monoid homomorphism from \mathcal{L} to \mathcal{N} which sends a list with the single element s to $f(s)$.

Let $j: List(S) \rightarrow N$ send the empty list to 1_N , and send a one-element list $s \in List(S)$ (with the single element $s \in S$) to $f(s)$. Extend the function to all members of $List(S)$ by putting $j(s_1 \cap s_2 \cap \dots \cap s_n) = j(s_1) \cdot j(s_2) \cdot \dots \cdot j(s_n)$. Then j is a monoid homomorphism.

Suppose k is another monoid homomorphism $j: List(S) \rightarrow N$ which sends a list with the single element s to $f(s)$, so j and k agree on unit lists. Hence

$$\begin{aligned} k(s_1 \cap s_2 \cap \dots \cap s_n) &= k(s_1) \cdot k(s_2) \cdot \dots \cdot k(s_n) \\ &= j(s_1) \cdot j(s_2) \cdot \dots \cdot j(s_n) \\ &= j(s_1 \cap s_2 \cap \dots \cap s_n). \end{aligned}$$

Whence j and k must agree on all members of $List(S)$. \square

19.6 A theorem on comma categories and limits

We end this chapter with what you can consider for the moment to be a slightly tricky exercise to test understanding of various definitions: so by all means skip it for now. However, we will appeal to this result later, so we prove it now to avoid breaking up the flow later.

Theorem 104. Suppose we have a functor $G: \mathcal{B} \rightarrow \mathcal{A}$ and an object $A \in \mathcal{A}$. Then if \mathcal{B} has limits of shape J and G preserves them, then $(A \downarrow G)$ also has limits of shape J .

Proof. Take any diagram $D: J \rightarrow (A \downarrow G)$. By definition, for any J -object J , DJ is a pair (D_J, f_J) , where D_J is a object in \mathcal{B} , and $f_J: A \rightarrow GD_J$ is an arrow in \mathcal{A} . And for any $d: J \rightarrow K$ in J , $Dd: D_J \rightarrow D_K$ is a \mathcal{B} -arrow such that $f_K = GDd \circ f_J$. The target is to show that, given our suppositions, D has a limit in $(A \downarrow G)$.

For convenience, we introduce the forgetful functor $U: (A \downarrow G) \rightarrow \mathcal{B}$ which acts in the obvious way, i.e. it sends an $(A \downarrow G)$ -object (B, f) to B , and sends an $(A \downarrow G)$ -arrow $j: B \rightarrow B'$ to itself.

Start with the functor $U \circ D: \mathbf{J} \rightarrow \mathcal{B}$. We know that *this* has a limit (by our hypothesis that \mathcal{B} has all limits of shape \mathbf{J}). Call this limit $[L, \pi_J]$. So L is a \mathcal{B} -object; and the π_J are \mathcal{B} -arrows such that any $d: J \rightarrow K$, $\pi_K = U D d \circ \pi_J$, i.e. $\pi_K = D d \circ \pi_J$. And since G preserves limits, we also know that $[GL, G\pi_J]$ is a limit cone in \mathcal{A} for $GUD: \mathbf{J} \rightarrow \mathcal{A}$.

Now take A and the arrows f_J . These comprise a cone $[A, f_J]$ over GUD in \mathcal{A} . Why? By definition, f_J is an arrow from A to GD_J i.e. to $GUD(J)$. And we know that for each $d: J \rightarrow K$, $f_K = GUD(d) \circ f_J$.

This cone $[A, f_J]$ must therefore factor uniquely through the limit $[GL, G\pi_J]$: i.e. there is a unique $u: A \rightarrow GL$ such that for all J , $f_J = G\pi_J \circ u$. Which, by definition of arrows in the comma category, means that for each J , π_J is an arrow from (L, u) to (D_J, f_J) in $(A \downarrow G)$. And these arrows π_J give us a cone over D in $(A \downarrow G)$ with vertex (L, u) , since as we have already seen, for any $d: J \rightarrow K$, $\pi_K = D d \circ \pi_J$.

If we can show that this cone is indeed a limit cone, we are done. Suppose therefore that there is another cone over D in $(A \downarrow G)$ with vertex (B, v) and arrows $b_J: (B, v) \rightarrow (D_J, f_J)$ in $(A \downarrow G)$ where, given $d: J \rightarrow K$ in \mathbf{J} , $b_K = D d \circ b_J$. We need to show that there is a unique $k: (B, v) \rightarrow (L, u)$ in $(A \downarrow G)$, i.e. a unique $k: B \rightarrow B'$ in \mathcal{B} , such that for each J , $b_J = \pi_J \circ k$. However, our assumptions also make $[B, b_J]$ a cone over $U \circ D$. So $[B, b_J]$ must factor through the limit $[L, \pi_J]$ via a unique $k: B \rightarrow L$: so there is indeed a unique k such that, for each J , $b_J = \pi_J \circ k$. \square

20 Natural isomorphisms

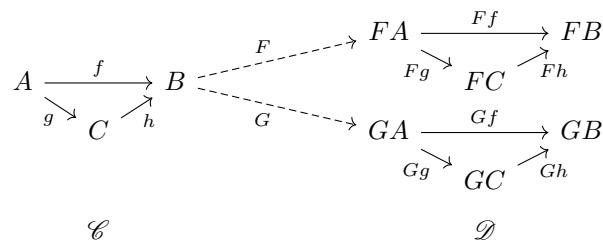
Category theory is an embodiment of Klein’s dictum that it is the maps that count in mathematics. If the dictum is true, then it is the functors between categories that are important, not the categories. And such is the case. Indeed, the notion of category is best excused as that which is necessary in order to have the notion of functor. But the progression does not stop here. There are maps between functors, and they are called natural transformations. (Freyd 1965, quoted in Marquis 2008.)

Natural transformations – and more specifically, natural isomorphisms – were there from the very start. The founding document of category theory is the paper by Samuel Eilenberg and Saunders Mac Lane ‘General theory of natural equivalences’ (Eilenberg and Mac Lane, 1945). But the key idea had already been introduced, three years previously, in a paper on ‘Natural isomorphisms in group theory’, before the categorial framework was invented in order to provide a general setting for the account (Eilenberg and Mac Lane, 1942). Natural isomorphisms and natural transformations are now going to start to take centre stage in our story too.

20.1 Natural isomorphisms between functors defined

Suppose we have a pair of parallel functors $\mathcal{C} \begin{matrix} \xrightarrow{F} \\ \xrightarrow{G} \end{matrix} \mathcal{D}$; when do the two functors ‘come to same’, categorially speaking?

Each of F and G projects the objects and arrows of \mathcal{C} into \mathcal{D} giving two images of \mathcal{C} within \mathcal{D} . Omitting identity arrows, we might have:



In general these images of \mathcal{C} can be significantly different. But at least we can guarantee that the results of applying F and G to objects will be the same (up to isomorphism) if there is a suite ψ of \mathcal{D} -isomorphisms $\psi_A: FA \xrightarrow{\sim} GA$, $\psi_B: FB \xrightarrow{\sim} GB$, etc., thus ensuring that $FA \cong GA$, $FB \cong GB$, etc.

Now, given such a suite of isomorphisms ψ and an arrow $f: A \rightarrow B$, there will be the following arrows from $FA \rightarrow FB$: Ff , of course, but also $\psi_B^{-1} \circ Gf \circ \psi_A$. If things are to fit together nicely, we should require these arrows to be the same (i.e. require that $\psi_B \circ Ff = Gf \circ \psi_A$). This ensures that when F and G are both applied to arrows $f, f', f'', \dots: A \rightarrow B$, there is a tidy one-to-one correspondence between the arrows Ff, Ff', Ff'', \dots and Gf, Gf', Gf'', \dots , so the results of applying F and G to arrows also stay in step.

Which all goes to motivate the following standard definition of an appropriate notion of isomorphism between parallel functors (or rather, it's a pair of definitions, one for each flavour of functor):

Definition 99. Let \mathcal{C} and \mathcal{D} be categories, let $\mathcal{C} \begin{matrix} \xrightarrow{F} \\ \xrightarrow{G} \end{matrix} \mathcal{D}$ be covariant functors (respectively, contravariant functors), and suppose that for each \mathcal{C} -object C there is a \mathcal{D} -isomorphism $\psi_C: FC \xrightarrow{\sim} GC$. Then ψ , the family of arrows ψ_C , is said to be a *natural isomorphism* between F and G if for every arrow $f: A \rightarrow B$ (respectively, $f: B \rightarrow A$, note the reversal!) in \mathcal{C} the following *naturality square* commutes in \mathcal{D} :

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow \psi_A & & \downarrow \psi_B \\ GA & \xrightarrow{Gf} & GB \end{array}$$

In this case, we write $\psi: F \xrightarrow{\sim} G$, and the ψ_C are said to be components of ψ . If there is such a natural isomorphism, F and G will be said to be naturally isomorphic, and we write $F \cong G$. △

20.2 Why 'natural'?

But why call this a *natural* isomorphism? There's a back-story which we mentioned in the preamble of the chapter and which we should now pause to explain, using one of Eilenberg and Mac Lane's own examples.

(a) Consider a finite dimensional vector space V over the reals \mathbb{R} , and the corresponding dual space V^* of linear functions $f: V \rightarrow \mathbb{R}$. It is elementary to show that V is isomorphic to V^* (there's a bijective linear map between the spaces).

Proof sketch: Take a basis $B = \{v_1, v_2, \dots, v_n\}$ for V . Define the functions $v_i^*: V \rightarrow \mathbb{R}$ by putting $v_i^*(v_j) = 1$ if $i = j$ and $v_i^*(v_j) = 0$ otherwise. Then

Natural isomorphisms

$B^* = \{v_1^*, v_2^*, \dots, v_n^*\}$ is a basis for V^* , and the linear function $\varphi_B: V \rightarrow V^*$ generated by putting $\varphi_B(v_i) = v_i^*$ is an isomorphism.

Note, however, that the isomorphism we have arrived at here depends on the initial choice of basis B . And no choice of basis B is more ‘natural’ than any other. So no one of the isomorphisms from $\varphi_B: V \rightarrow V^*$ of the kind just defined is to be especially preferred.

To get a sharply contrasting case, now consider V^{**} the double dual of V , i.e. the space of functionals $g: V^* \rightarrow \mathbb{R}$. Suppose we select a basis B for V , define a derived basis B^* for V^* as we just did, and then use this new basis in turn to define a basis B^{**} for V^{**} by repeating the same construction. Then we can construct an isomorphism from V to V^{**} by mapping the elements of B to the corresponding elements of B^{**} . However, *we don't have to go through any such palaver of initially choosing a basis*. Suppose we simply define $\psi_V: V \rightarrow V^{**}$ as acting on an element $v \in V$ to give as output the functional $\psi_V(v): V^* \rightarrow \mathbb{R}$ which sends a function $f: V \rightarrow \mathbb{R}$ to the value $f(v)$: in short, we set $\psi_V(v)(f) = f(v)$. It is readily checked that ψ_V is an isomorphism (we rely on the fact that V is finite-dimensional). And obviously we get *this* isomorphism independently of any arbitrary choice of basis.

Interim summary: it is very natural(!) to say that the isomorphisms of the kind we described between V and V^* are not intrinsic, are not ‘natural’ to the spaces involved. By contrast there *is* a ‘natural’ isomorphism between V and V^{**} , generated by a general procedure that applies to any suitable vector space.

Now, there are many other cases where we might similarly want to contrast intuitively ‘natural’ maps with more arbitrarily cooked-up maps between structured objects. The story goes that such talk was already bandied about quite a bit e.g. by topologists in the 1930s. So a question arises: can we give a clear general account of what makes for naturality here? Eilenberg and Mac Lane were aiming to provide such a story.

(b) To continue with our example, the isomorphism $\psi_V: V \xrightarrow{\sim} V^{**}$ which we constructed might be said to be natural *because the only information about V it relies on is that V is a finite dimensional vector space over the reals*.

That implies that our construction will work in exactly same way for any other such vector space W , so we get a corresponding isomorphism $\psi_W: W \xrightarrow{\sim} W^{**}$. Now, we will expect such naturally constructed isomorphisms to respect the relation between a structure-preserving map f between the spaces V and W and its double-dual correlate map between V^{**} to W^{**} . Putting that more carefully, we want the following informal diagram to commute, whatever vector spaces we take and for any linear map $f: V \rightarrow W$,

$$\begin{array}{ccc} V & \xrightarrow{f} & W \\ \downarrow \psi_V & & \downarrow \psi_W \\ V^{**} & \xrightarrow{DD(f)} & W^{**} \end{array}$$

where $DD(f)$ is the double-dual correlate of f .

Recall, back in §15.8, we saw that the correlate Df of $f: V \rightarrow W$ is the functional $(- \circ f): W^* \rightarrow V^*$; and then moving to the double dual, the correlate DDf will be the functional we can notate $(- \circ (- \circ f)): V^{**} \rightarrow W^{**}$. Our diagram can then indeed be seen to commute, both paths sending an element $v \in V$ to the functional that maps a function $k: W \rightarrow \mathbb{R}$ to the value $k(f(v))$. Think about it!

(c) So far, so good. Now let's pause to consider why there can't be a similarly 'natural' isomorphism from V to V^* . (The isomorphisms based on an arbitrary choice of basis aren't natural: but we want to show that there is no other 'natural' isomorphism either.)

Suppose then that there were a construction which gave us an isomorphism $\varphi_V: V \xrightarrow{\sim} V^*$ which again does not depend on information about V other than that it has the structure of a finite dimensional vector space. So again we will want the construction to work the same way on other such vector spaces, and to be preserved by structure-preserving maps between the spaces. This time, therefore, we will presumably want the following diagram to commute for any structure-preserving f between vector spaces (note, however, that we have to reverse an arrow for things to make any sense, given our definition of the contravariant functor D):

$$\begin{array}{ccc} V & \xrightarrow{f} & W \\ \downarrow \varphi_V & & \downarrow \varphi_W \\ V^* & \xleftarrow{D(f)} & W^* \end{array}$$

Hence $D(f) \circ \varphi_W \circ f = \varphi_V$. But by hypothesis, the φ s are isomorphisms; so in particular φ_V has an inverse. So we have $(\varphi_V^{-1} \circ D(f) \circ \varphi_W) \circ f = 1_V$. Therefore f has a left inverse. But it is obvious that in general, a linear map $f: V \rightarrow W$ need not have a left inverse. Hence there can't in general be isomorphisms $\varphi_V, \varphi_W: V \rightarrow V^*$ making that diagram commute.

(d) We started off by saying that, intuitively, there's a 'natural', intrinsic, isomorphism between a (finite dimensional) vector space and its double dual, one that depends only on their structures as vector spaces. And we've now suggested that this intuitive idea can be reflected by saying that a certain diagram always commutes, for any choice of vector spaces and structure-preserving maps between them.

We have also seen that we can't get analogous always-commuting diagrams for the case of isomorphisms between a vector space and its dual – which chimes with the intuition that the obvious examples are *not* 'natural' isomorphisms.

So this gives us a promising way forward: characterize 'naturalness' here in terms of the availability of a family of isomorphisms which make certain informal

Natural isomorphisms

(non-categorical) diagrams commute. Note next, however, that the claim that the diagram

$$\begin{array}{ccc} V & \xrightarrow{f} & W \\ \downarrow \psi_V & & \downarrow \psi_W \\ V^{**} & \xrightarrow{DD(f)} & W^{**} \end{array}$$

always commutes can be indeed put a slightly different way, using category-speak.

For we have in effect been talking about the category we'll here call simply **Vect** (of finite-dimensional spaces over the reals and the structure-preserving maps between them), and about a functor we can call $DD: \mathbf{Vect} \rightarrow \mathbf{Vect}$ which takes a vector space to its double dual, and maps each arrow between vector spaces to its double-dual correlate as explained. There is also a trivial functor $1: \mathbf{Vect} \rightarrow \mathbf{Vect}$ that maps each vector space to itself and each **Vect**-arrow to itself. So we can re-express the claim that the last diagram commutes as follows. For every arrow $f: V \rightarrow W$ in **Vect**, there are isomorphisms ψ_V and ψ_W in **Vect** such that *this* diagram commutes:

$$\begin{array}{ccc} 1(V) & \xrightarrow{1(f)} & 1(W) \\ \downarrow \psi_V & & \downarrow \psi_W \\ DD(V) & \xrightarrow{DD(f)} & DD(W) \end{array}$$

In other words, in the terms of the previous section, the suite of isomorphisms ψ_V provide a natural isomorphism $\psi: 1 \xrightarrow{\cong} DD$.

(e) In sum: our claim that there is an intuitively ‘natural’ isomorphism between two *spaces*, a vector space and its double dual, now becomes reflected in the claim that there is an isomorphism in our official sense between two *functors*, the identity and the double-dual functors from the category **Vect** to itself. Hence the aptness of calling the latter isomorphism between functors a *natural* isomorphism.

We will return at the end of the chapter to the thought that we can generalize from our example of vector spaces and claim that in many (most? all?) cases, intuitively ‘natural’ isomorphisms between widgets and wombats can be treated officially as natural isomorphisms between suitable functors.

20.3 More examples of natural isomorphisms

We now have one case to hand. Let's next give some more simple examples of natural isomorphisms:

20.3 More examples of natural isomorphisms

- (1) We quickly mention the trivial case. Given any functor $F: \mathcal{C} \rightarrow \mathcal{D}$, then the following diagram of course commutes for every $f: A \rightarrow B$ in \mathcal{C} :

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow 1_{FA} & & \downarrow 1_{FB} \\ FA & \xrightarrow{Ff} & FB \end{array}$$

So we have a natural isomorphism $1_F: F \xrightarrow{\cong} F$, where the components $(1_F)_A$ of the isomorphism are the identity arrows $1_{(FA)}$.

- (2) Given a group $G = (G, *, e)$ we can define its opposite $G^{op} = (G, *^{op}, e)$, where $a *^{op} b = b * a$.

We can also define a functor $Op: \mathbf{Grp} \rightarrow \mathbf{Grp}$ which sends a group G to its opposite G^{op} , and sends an arrow f in the category, i.e. a group homomorphism $f: G \rightarrow H$, to $f^{op}: G^{op} \rightarrow H^{op}$ where $f^{op}(a) = f(a)$ for all a in G . f^{op} so defined is indeed a group homomorphism, since

$$f^{op}(a *^{op} a') = f(a' * a) = f(a') * f(a) = f^{op}(a) *^{op} f^{op}(a')$$

Claim: there is a natural isomorphism $\psi: 1 \xrightarrow{\cong} Op$ (where 1 is the trivial identity functor in \mathbf{Grp}).

Proof. We need to find a family of isomorphisms ψ_G, ψ_H, \dots in \mathbf{Grp} such that the following diagram always commutes for any homomorphism $f: G \rightarrow H$:

$$\begin{array}{ccc} G & \xrightarrow{f} & H \\ \downarrow \psi_G & & \downarrow \psi_H \\ G^{op} & \xrightarrow{f^{op}} & H^{op} \end{array}$$

(Careful: G, H are groups here, not functors!) Now, since taking the opposite *between* groups involves reversing the order of multiplication and taking inverses *inside* a group in effect does the same, let's put $\psi_G(a) = a^{-1}$ for any G -element a , and likewise for ψ_H , etc. It is easy to check that with this choice of components, ψ is a natural isomorphism. \square

- (3) Recall from §15.2 the functor $List: \mathbf{Set} \rightarrow \mathbf{Set}$ which sends a set X to the set of finite lists of members of X . One natural isomorphism from this functor to itself is the identity isomorphism $1: List \xrightarrow{\cong} List$. But there is also another natural isomorphism $\rho: List \xrightarrow{\cong} List$, whose component $\rho_X: List(X) \rightarrow List(X)$ acts on a list of X -elements to reverse their order.

- (4) Now for an example involving contravariant functors from \mathbf{Set} to \mathbf{Set} .

First, recall the contravariant powerset functor $\bar{P}: \mathbf{Set} \rightarrow \mathbf{Set}$ which maps a set X to its powerset $\mathcal{P}(X)$, and maps a set-function $f: Y \rightarrow X$ to the function $Inv(f)$ which sends $U \subseteq X$ to its inverse image $f^{-1}[U] \subseteq Y$.

Natural isomorphisms

And let C be the hom-functor $\text{Set}(-, 2)$, where 2 is some nice two-element set such as $\{\{\emptyset\}, \emptyset\}$ which we can think of as $\{\text{true}, \text{false}\}$. So C sends a set X to $\text{Set}(X, 2)$, i.e. the set of functions from X to 2 : and C sends an arrow $f: Y \rightarrow X$ to the function $- \circ f: \text{Set}(X, 2) \rightarrow \text{Set}(Y, 2)$ (i.e. the function which sends an arrow $g: X \rightarrow 2$ to the arrow $g \circ f: Y \rightarrow 2$).

Claim: $\bar{P} \cong C$.

Proof. We need to find a family of isomorphisms ψ_X, ψ_Y, \dots in Set such that the following diagram always commutes:

$$\begin{array}{ccc} \bar{P}X & \xrightarrow{\bar{P}f} & \bar{P}Y \\ \downarrow \psi_X & & \downarrow \psi_Y \\ CX & \xrightarrow{Cf} & CY \end{array} \quad \text{equivalently} \quad \begin{array}{ccc} \mathcal{P}(X) & \xrightarrow{\text{Inv}(f)} & \mathcal{P}(Y) \\ \downarrow \psi_X & & \downarrow \psi_Y \\ \text{Set}(X, 2) & \xrightarrow{- \circ f} & \text{Set}(Y, 2) \end{array}$$

Take any ψ_X to be the isomorphism which associates a set $U \subseteq X$ with its characteristic function (i.e. the function which sends an element of X to *true* iff it is in U). Then it is easy to see that the diagram will always commute. Both routes sends a set $U \subseteq X$ to the function which sends y in Y to *true* iff $fy \in U$. \square

- (5) This time we take a certain pair of (covariant) functors $\text{Grp} \xrightarrow[U]{V} \text{Set}$.

Here U is simply the forgetful functor which sends a group G to its underlying set, and sends homomorphisms to themselves. While V is the hom-functor $\text{Grp}(Z, -)$, where Z is the group of integers under addition. So, by definition, V sends an object, i.e. a group G , to the set of group homomorphisms from Z to G . And V sends an arrow $f: G \rightarrow G'$ to the function we notate $f \circ -$, i.e. the function which sends a homomorphism $h: Z \rightarrow G$ to the homomorphism $f \circ h: Z \rightarrow G'$. Claim: $U \cong V$.

Proof. Note first that a group homomorphism from $Z = (\mathbb{Z}, 0, +)$ to $G = (G, e, \cdot)$ is entirely fixed by fixing where 1 goes. For 0 has to go to the identity element e ; and if 1 goes to the element a , every sum $1+1+1+\dots+1$ has to go to the corresponding $a \cdot a \cdot a \cdot \dots \cdot a$, with inverses going to inverses. Which means that there is a set-bijection ψ_G from elements of G to members of $\text{Grp}(Z, -)$.

It is then immediate that the required naturality square commutes for any $f: G \rightarrow G'$:

$$\begin{array}{ccc} UG & \xrightarrow{f} & UG' \\ \downarrow \psi_G & & \downarrow \psi_{G'} \\ VG & \xrightarrow{f \circ -} & VG' \end{array}$$

20.3 More examples of natural isomorphisms

with either route round the square taking us from an element $a \in G$ to the unique homomorphism from Z to G' which sends 1 to fa . \square

Our next examples also involve hom-functors. For motivation, reflect on the natural one-to-one bijection between two-place set functions from A and B to C , and one-place functions from A to functions-from- B -to- C (see §13.1). Categorically, that gives us an isomorphism between the hom-sets $\mathbf{Set}(A \times B, C)$ and $\mathbf{Set}(A, C^B)$. And the intuitive naturality of the bijection means that this doesn't depend on particular choices of A , B or C . So we will expect, inter alia, that the hom-functors $\mathbf{Set}(A \times B, -)$ and $\mathbf{Set}(A, (-)^B)$ are isomorphic. Moreover, this should apply not just to the category \mathbf{Set} but, generalizing,

- (6) If \mathcal{C} is a locally small category with exponentials, then $\mathcal{C}(A \times B, -) \cong \mathcal{C}(A, (-)^B)$.

Proof. Here $\mathcal{C}(A, (-)^B) = \mathcal{C}(A, -) \circ (-)^B$, where $(-)^B$ is the functor that we met in §15.6. Now, $(-)^B$ sends an arrow $f: C \rightarrow C'$ to $f^B = \overline{f \circ ev}$. Hence $\mathcal{C}(A, (-)^B)$ sends f to $\overline{f \circ ev} \circ -$.

To provide the announced natural isomorphism, we need to find a family of isomorphisms ψ_C such that for every $f: C \rightarrow C'$ in \mathcal{C} , the following diagram commutes in \mathbf{Set} :

$$\begin{array}{ccc} \mathcal{C}(A \times B, C) & \xrightarrow{\mathcal{C}(A \times B, f) = f \circ -} & \mathcal{C}(A \times B, C') \\ \downarrow \psi_C & & \downarrow \psi_{C'} \\ \mathcal{C}(A, C^B) & \xrightarrow{\mathcal{C}(A, f^B) = \overline{(f \circ ev)} \circ -} & \mathcal{C}(A, C'^B) \end{array}$$

Suppose then that we take the component ψ_C to be the isomorphism which sends an arrow g in $\mathcal{C}(A \times B, C)$ to its exponential transpose \bar{g} in $\mathcal{C}(A, C^B)$. Will that make the diagram commute?

Chase an arrow g in $\mathcal{C}(A \times B, C)$ round the diagram both ways. Then the diagram will commute if $\overline{f \circ ev} \circ \bar{g} = \overline{f \circ g}$. But consider:

$$\begin{array}{ccccc} & & A \times B & & \\ & & \downarrow \bar{g} \times 1_B & \searrow g & \\ & & C^B \times B & \xrightarrow{ev} & C \\ \overline{f \circ g} \times 1_B & \curvearrowright & \downarrow \overline{f \circ ev} \times 1_B & & \downarrow f \\ & & C'^B \times B & \xrightarrow{ev'} & C' \end{array}$$

Note the composite $f \circ g: A \times B \rightarrow C'$. By the definition of $[C'^B, ev']$ as an exponential, there is a unique arrow $\overline{f \circ g}$ such that

$$ev' \circ \overline{f \circ g} \times 1_B = f \circ g.$$

Natural isomorphisms

But since the top triangle and the bottom square also commute, we have

$$f \circ g = ev' \circ (\overline{f \circ ev} \times 1_B) \circ (\overline{g} \times 1_B) = ev' \circ (\overline{f \circ ev \circ \overline{g}}) \times 1_B.$$

Hence, by the uniqueness requirement, we get $\overline{f \circ ev} \circ \overline{g} = \overline{f \circ g}$, and we are done. \square

- (7) Similarly motivated, we see that if \mathcal{C} is a locally small category with exponentials, then $\mathcal{C}(- \times B, C) \cong \mathcal{C}(-, C^B)$

Proof. Here, $\mathcal{C}(- \times B, C) = \mathcal{C}(-, C) \circ (- \times B)$, where the first is a contravariant hom-functor, and $- \times B$ is another functor that we met in §15.6. Now, $- \times B$ sends an arrow $f: A' \rightarrow A$ to $f \times 1_B$. Hence $\mathcal{C}(- \times B, C)$ sends f to $- \circ (f \times 1_B): (A \times B, C) \rightarrow (A' \times B, C)$.

To provide the announced natural isomorphism, we need to find a family of isomorphisms ψ_A such that for every $f: A' \rightarrow A$ in \mathcal{C} , the following diagram commutes in **Set**:

$$\begin{array}{ccc} \mathcal{C}(A \times B, C) & \xrightarrow{\mathcal{C}(f \times 1_B, C) = - \circ (f \times 1_B)} & \mathcal{C}(A' \times B, C) \\ \downarrow \psi_A & & \downarrow \psi_{A'} \\ \mathcal{C}(A, C^B) & \xrightarrow{\mathcal{C}(f, C^B) = - \circ f} & \mathcal{C}(A', C^B) \end{array}$$

As before, take the component ψ_A to be the isomorphism which sends an arrow g in $\mathcal{C}(A \times B, C)$ to its transpose \overline{g} in $\mathcal{C}(A, C^B)$.

Chase an arrow g in $\mathcal{C}(A \times B, C)$ round the diagram both ways. Then the diagram will commute if $\overline{g} \circ f = \overline{g \circ (f \times 1_B)}$.

But now consider this further diagram:

$$\begin{array}{ccccc} A' \times B & \xrightarrow{f \times 1_B} & A \times B & & \\ & \searrow & \downarrow \overline{g} \times 1_B & \searrow g & \\ & \overline{g \circ (f \times 1_B)} \times 1_B & C^B \times B & \xrightarrow{ev} & C \end{array}$$

By definition, $\overline{g \circ (f \times 1_B)}: A' \rightarrow C^B$ is the unique arrow that when plugged into $- \times 1_B$ makes the rhombus commute.

But the right-hand triangle commutes, so it follows that $(\overline{g} \times 1_B) \circ (f \times 1_B)$ is another arrow from $A' \times B$ to $C^B \times B$ which makes the rhombus commute. However, by Theorem 37, $(\overline{g} \times 1_B) \circ (f \times 1_B) = (\overline{g} \circ f) \times 1_B$. Hence $\overline{g} \circ f$ plugged into $- \times 1_B$ also makes the rhombus commute. Which proves that $\overline{g} \circ f = \overline{g \circ (f \times 1_B)}$. \square

These last two proofs show how confirming that two functors are indeed naturally isomorphic (even in simple cases where the result is entirely expected) can be fiddly. We will encounter this sort of annoyance again.

20.4 Natural/unnatural isomorphisms between objects

(a) Suppose we have functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$; and let A, A', A'', \dots be objects in \mathcal{C} . Then there will be objects $FA, FA', FA'' \dots$ and $GA, GA', GA'' \dots$ in \mathcal{D} . And in some cases these will be pairwise isomorphic, so that we have $FA \cong GA$, $FA' \cong GA'$, $FA'' \cong GA'' \dots$.

One way this can happen, as we have seen, is that there is a natural isomorphism between the functors F and G . But it is important to emphasize that it can happen in other, ‘unnatural’, ways. We’ve met unnaturalness before, but still let’s have a couple more examples, one a toy example to make again the point of principle, then a standard illustrative case which is worth thinking through:

- (1) Suppose \mathcal{C} is a category with exactly one object A , and two arrows, the identity arrow 1_A , and distinct arrow f , where $f \circ f = f$. And now consider two functors, the identity functor $1_{\mathcal{C}}: \mathcal{C} \rightarrow \mathcal{C}$, and the functor $F: \mathcal{C} \rightarrow \mathcal{C}$ which sends the only object to itself, and sends both arrows to the identity arrow. Then, quite trivially, we have $1_{\mathcal{C}}(A) \cong F(A)$ for the one and only object in \mathcal{C} . But there isn’t a natural isomorphism between the functors, because by hypothesis $1_A \neq f$, and hence the square

$$\begin{array}{ccc}
 F(A) & \xrightarrow{F(f)} & F(A) \\
 \downarrow 1_A & & \downarrow 1_A \\
 1_{\mathcal{C}}(A) & \xrightarrow{1_{\mathcal{C}}(f)} & 1_{\mathcal{C}}(A)
 \end{array}
 , \text{ which is simply }
 \begin{array}{ccc}
 A & \xrightarrow{1_A} & A \\
 \downarrow 1_A & & \downarrow 1_A \\
 A & \xrightarrow{f} & A
 \end{array}$$

cannot commute.

- (2) We’ll work in the category \mathcal{F} of finite sets and *bijections* between them.

There is a functor $Sym: \mathcal{F} \rightarrow \mathcal{F}$ which (i) sends a set A in \mathcal{F} to the set of permutations on A (treating permutation functions as sets, this is a finite set), and (ii) sends a bijection $f: A \rightarrow B$ in \mathcal{F} to the bijection that sends the permutation p on A to the permutation $f \circ p \circ f^{-1}$ on B . Note: if A has n members, there are $n!$ members of the set of permutations on A .

There is also a functor $Ord: \mathcal{F} \rightarrow \mathcal{F}$ which (i) sends a set A in \mathcal{F} to the set of total linear orderings on A (you can identify an order-relation with a set, so we can think of this too as a finite set), and (ii) sends a bijection $f: A \rightarrow B$ in \mathcal{F} to the bijection $Ord(f)$ which sends a total order on A to the total order on B where $x <_A y$ iff $f(x) <_B f(y)$. Again, if A has n members, there are also $n!$ members of the set of linear orderings on A .

Now, for any object A of \mathcal{F} , $Sym(A) \cong Ord(A)$ (since they are equinumerous finite sets). But there cannot be a natural isomorphism ψ between the functors Sym and Ord . For suppose otherwise, and consider the functors acting on a bijection $f: A \rightarrow A$. Then the following naturality square would have to commute:

Natural isomorphisms

$$\begin{array}{ccc}
 \text{Sym}(A) & \xrightarrow{\text{Sym}(f)} & \text{Sym}(A) \\
 \downarrow \psi_A & & \downarrow \psi_A \\
 \text{Ord}(A) & \xrightarrow{\text{Ord}(f)} & \text{Ord}(A)
 \end{array}$$

Consider then what happens to the identity permutation i in $\text{Sym}(A)$: it gets sent by $\text{Sym}(f)$ to $f \circ i \circ f^{-1} = i$. So the naturality square would tell us that $\psi_A(i) = \text{Ord}(f)(\psi_A(i))$. But that in general won't be so – suppose f swaps around elements, so $\text{Ord}(f)$ is not the ‘do nothing’ identity map.

In a summary slogan, then: pointwise isomorphism doesn't entail natural isomorphism.

(b) We are, however, going mostly to be interested in cases where $FA \cong GA$ (and $FA' \cong GA'$, $FA'' \cong GA'' \dots$) as a result of a natural isomorphism. There is standard terminology for such cases:

Definition 100. Given functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$ and A an object in \mathcal{C} , we say that $FA \cong GA$ *naturally in A* (or *naturally in A in \mathcal{C}*) just if F and G are *naturally isomorphic*.

The definition mentions just a specific object A in \mathcal{C} ; but there is an implicit generality here. For if $FA \cong GA$ naturally in A , then for some ψ we have $\psi: F \xrightarrow{\cong} G$. So as well as an isomorphism $\psi_A: FA \xrightarrow{\cong} GA$, there are other isomorphisms $\psi_{A'}: FA' \xrightarrow{\cong} GA'$, $\psi_{A''}: FA'' \xrightarrow{\cong} GA''$, etc., for other objects A', A'', \dots , making $FA' \cong GA'$ (naturally in A'), $FA'' \cong GA''$ (naturally in A''), etc.

To help fix ideas, let's note a useful little result about this notion of an isomorphism between objects holding naturally:

Theorem 105. *Given functors $F, G, H: \mathcal{C} \rightarrow \mathcal{D}$, an object A in \mathcal{C} , and a functor $K: \mathcal{B} \rightarrow \mathcal{C}$, then*

- (1) *if $FA \cong GA$ naturally in A , then for all A' in \mathcal{C} , $FA' \cong GA'$ naturally in A' .*
- (2) *if $FA \cong GA$ and $GA \cong HA$, both naturally in A , then $FA \cong HA$ naturally in A .*
- (3) *if $FA \cong GA$ naturally in A , then $FKB \cong GKB$ naturally in B in \mathcal{B} .*

Proof. (1) is immediate, for if $FA \cong GA$ naturally in A , F is naturally isomorphic to G , so there is a component of the natural isomorphism at A' making $FA' \cong GA'$.

For (2), just note that natural isomorphisms vertically compose.

For (3), just note that, if there is a natural isomorphism α between F and G , then (by ‘whiskering’) there is a natural isomorphism between FK and GK , whose component at B is α_{KB} . \square

(c) Let’s mention just a few examples. We have seen that $V \cong DDV$ naturally in V in **Vect**: that was the message of §20.2.

Likewise, $UG \cong \text{Grp}(Z, G)$ naturally in G in **Grp**: that was the message of §20.3 (5).

And from §20.3 (6) and (7) we get the following, which we will highlight as a theorem:

Theorem 106. *Given a category \mathcal{C} with exponentials, $\mathcal{C}(A \times B, C) \cong \mathcal{C}(A, C^B)$ both naturally in A and naturally in C .*

20.5 An ‘Eilenberg/Mac Lane Thesis’?

Let’s return to the question we raised before. Can we generalize from e.g. our example of a vector space and its double dual, and say that whenever we have a ‘natural’ isomorphism between widgets and wombats (i.e. one that doesn’t depend on arbitrary choices of co-ordinates, or the like), this can be regimented as a natural isomorphism between suitable associated functors? Let’s call the claim that we *can* generalize like this the ‘Eilenberg/Mac Lane Thesis’.

I choose the label to be reminiscent of the Church/Turing Thesis that we all know and love, which asserts that every algorithmically computable function (in an informally characterized sense) is in fact recursive/Turing computable/lambda computable. A certain intuitive concept, this Thesis claims, in fact picks out the same functions as certain (provably equivalent) sharply defined concepts.

What kind of evidence do we have for this thesis? Two sorts: (1) ‘quasi-empirical’, i.e. no unarguable clear exceptions have been found, and (2) conceptual, as in for example Turing’s own efforts to show that when we reflect on what we mean by algorithmic computation we get down to the sort of operations that a Turing machine can emulate, so morally a computable function just ought to be Turing computable. The evidence in this case is so overwhelming that in fact we are allowed to appeal to the Church/Turing Thesis as a labour-saving device: if we can give an arm-waving sketch of an argument that a certain function is algorithmically computable, we are allowed to assume that it is indeed recursive/Turing computable/lambda computable without doing the hard work of e.g. defining a Turing machine to compute it.

We now seem to have on the table another Thesis of the same general type: an informal intuitive concept, the Eilenberg/Mac Lane Thesis claims, in fact picks out the same isomorphisms as a certain sharply defined categorial concept.

Natural isomorphisms

Evidence? We would expect two sorts. (1*) ‘quasi-empirical’, a lack of clear exceptions, and maybe (2*) conceptual, an explanation of why the Thesis just ought to be true.

It is, however, not clear exactly how things stand evidentially here, and the usual textbook discussions of natural isomorphisms oddly don’t pause to do much more than give a few examples. More really needs to be said. We therefore can’t suppose that the new Eilenberg/Mac Lane Thesis is so secure that we can cheerfully appeal to it in the same labour-saving way as the old Church/Turing Thesis. In other words, even if (i) intuitively an isomorphism between objects seems to be set up in a very ‘natural’ way, without appeal to arbitrary choices, and (ii) we can readily massage the claim of an isomorphism into a claim about at least pointwise isomorphism of relevant functors, we really need to pause to work through a proof if we are to conclude that in fact (iii) there is a natural isomorphism here in the official categorial sense. Annoying, as we said. For as we have already seen, such proofs can be a bit tedious.

21 Natural transformations

We think of isomorphisms categorially as special cases of some wider class of morphisms, namely those of the morphisms which have inverses. Thus isomorphisms inside categories are particular arrows, those with inverses; isomorphisms between categories are particular functors, those with inverses. And now natural isomorphisms between functors are special cases of What?

21.1 Natural transformations

(a) The generalized notion of morphisms between functors that we want is obvious enough. In fact, as before, the definition gives us two notions for the price of one:

Definition 101. Let \mathcal{C} and \mathcal{D} be categories, let $\mathcal{C} \begin{smallmatrix} F \\ \rightrightarrows \\ G \end{smallmatrix} \mathcal{D}$ be covariant functors (respectively, contravariant functors), and suppose that for each \mathcal{C} -object C there is a \mathcal{D} -arrow $\alpha_C: FC \rightarrow GC$. Then α , the family of arrows α_C , is a *natural transformation* between F and G if for every $f: A \rightarrow B$ (respectively $f: B \rightarrow A$, note the reversal!) in \mathcal{C} the following *naturality square* commutes in \mathcal{D} :

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ GA & \xrightarrow{Gf} & GB \end{array}$$

In this case, we write $\alpha: F \Rightarrow G$. (A natural isomorphism is thus a natural transformation each of whose components is an isomorphism.) \triangle

Note that while different styles of arrows can be found in use, Greek letters are almost universally used for names of natural transformations.

(b) In sum, a natural transformation between functors $\mathcal{C} \begin{smallmatrix} F \\ \rightrightarrows \\ G \end{smallmatrix} \mathcal{D}$ sends an F -image of (some or all of) \mathcal{C} to its G -image in a way which respects the internal structure of the original at least to the extent of preserving composition

Natural transformations

of arrows. Let's have a couple of initial toy examples of natural transformations which aren't isomorphisms:

- (1) Suppose \mathcal{D} has a terminal object 1 , and let $F: \mathcal{C} \rightarrow \mathcal{D}$ be any functor. Then there is also a parallel functor $T: \mathcal{C} \rightarrow \mathcal{D}$ which sends every \mathcal{C} -object to the terminal object 1 , and every \mathcal{C} -arrow to the identity arrow on the terminal object. Claim: there is a natural transformation $\alpha: F \Rightarrow T$.

Proof. We need a suite of \mathcal{D} -arrows α_A (one for each A in \mathcal{C}) which make the following commute for any $f: A \rightarrow B$ in \mathcal{C} :

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ 1 & \xrightarrow{1_1} & 1 \end{array}$$

Put each component of α to be the unique arrow from its source to the terminal object: and the diagram must commute because all arrows from FA to 1 are equal. \square

- (2) Recall the functor $List: \mathbf{Set} \rightarrow \mathbf{Set}$ where $List_{ob}$ sends a set A to the set of all finite lists of members of A and $List_{arw}$ sends a set-function $f: A \rightarrow B$ to the map that sends a list $a_0 \wedge a_1 \wedge a_2 \wedge \dots \wedge a_n$ to $fa_0 \wedge fa_1 \wedge fa_2 \wedge \dots \wedge fa_n$. Claim: there is a natural transformation $\alpha: 1 \Rightarrow List$, where 1 is the trivial identity functor $1: \mathbf{Set} \rightarrow \mathbf{Set}$.

Proof. We need a suite of functions α_A which make the following commute for any $f: A \rightarrow B$ in \mathcal{C} :

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ List(A) & \xrightarrow{List(f)} & List(B) \end{array}$$

For any A , put α_A to be the function which sends an element of A to the length-one list containing just that element, and we are immediately done. \square

Note, by the way, that we can think of $List$ as the composite functor GF where F is the 'free' functor from \mathbf{Set} to \mathbf{Mon} which we met in §15.5 and G is the forgetful functor in the other direction, from \mathbf{Mon} to \mathbf{Set} . We will find later that there are many important natural transformations which are significantly of the form $\alpha: 1_{\mathcal{C}} \Rightarrow GF$ (where $1_{\mathcal{C}}$ is the identity functor from \mathcal{C} to itself, and for some \mathcal{D} , $\mathcal{C} \xrightleftharpoons[G]{F} \mathcal{D}$) and also many of the form $\alpha: FG \Rightarrow 1_{\mathcal{D}}$.

(c) Now for two cases of natural transformations which aren't isomorphisms and which have rather more mathematical significance (though we will only sketch them here):

- (3) For those who know just a bit more group theory, consider the abelianization of a group G . Officially, this is the quotient of a group by its commutator subgroup $[G, G]$ (but you can think of it as the 'biggest' Abelian group A for which there is a surjective homomorphism from G onto A). There is then a functor Ab which sends a group G to its abelianization $Ab(G)$, and sends an arrow $f: G \rightarrow H$ to the arrow $Ab(f): Ab(G) \rightarrow Ab(H)$ defined in a fairly obvious way.

We therefore have a pair of functors, $\text{Grp} \xrightarrow[Ab]{1} \text{Grp}$, and we can then check that the following diagram always commutes,

$$\begin{array}{ccc} G & \xrightarrow{f} & H \\ \downarrow \alpha_G & & \downarrow \alpha_H \\ Ab(G) & \xrightarrow{Ab(f)} & Ab(H) \end{array}$$

where $\alpha_G = G/[G, G]$. So we have a natural transformation, but not usually a natural isomorphism, between the functors 1 and Ab .

- (4) For those who know rather more topology, we can mention two important functors from topological spaces to groups. One we've met before in §15.7, namely the functor $\pi_1: \text{Top}_* \rightarrow \text{Grp}$ which sends a space with a basepoint to its fundamental group at the base point. The other functor $H_1: \text{Top} \rightarrow \text{AbGrp}$ sends a space to the abelian group which is its first homology group (we aren't going to try to explain that here!). Now these functors aren't yet parallel functors between the same categories. But we can define a functor $H'_1: \text{Top}_* \rightarrow \text{Grp}$ which first forgets base points of spaces, then applies H_1 , and then forgets that the relevant groups are abelian. We simply record that it is a very important fact of topology that, in our categorical terms, there is natural transformation from π_1 to H'_1 .

(d) A natural transformation is a suite of arrows from various sources, with each pair of arrows making certain diagrams commute. A cone is essentially a suite of arrows all from the same source, the apex of the cone, with each pair of arrows making certain diagrams commute. Which suggests that we should be able to treat cones as special cases of natural transformations. And we can.

- (5) Suppose we have a diagram-as-functor $D: \mathbf{J} \rightarrow \mathcal{C}$ and also a collapse-to- C functor $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$, i.e. a constant functor which sends every \mathbf{J} -object to C in \mathcal{C} and every \mathbf{J} -arrow to 1_C (see §15.2 (F10)). Let's ask: what does it take for there to be a natural transformation $\alpha: \Delta_C \rightarrow D$?

Natural transformations

Given such an α , the following diagram must commute for any J-arrow $j: K \rightarrow L$:

$$\begin{array}{ccc}
 \Delta_C K & \xrightarrow{\Delta_C j} & \Delta_C L \\
 \alpha_K \downarrow & & \downarrow \alpha_L \\
 DK & \xrightarrow{Dj} & DL
 \end{array}
 =
 \begin{array}{ccc}
 C & \xrightarrow{1_C} & C \\
 \alpha_K \downarrow & & \downarrow \alpha_L \\
 DK & \xrightarrow{Dj} & DL
 \end{array}
 =$$

$$\begin{array}{ccc}
 & C & \\
 \alpha_K \swarrow & & \searrow \alpha_L \\
 DK & \xrightarrow{Dj} & DL
 \end{array}$$

Which makes the α_J (where J runs over objects in \mathcal{J}) the legs of a cone over D with a vertex C . Conversely, the legs of any cone over D with a vertex C can be assembled into a natural transformation $\alpha: \Delta_C \rightarrow D$. So that means that cones (thought of the austere way, as simply suites of arrows) are indeed certain natural transformations.

21.2 Vertical composition of natural transformations

Before continuing, a further bit of notation will prove useful. When we have functors $F: \mathcal{C} \rightarrow \mathcal{D}, G: \mathcal{C} \rightarrow \mathcal{D}$, together with a natural transformation $\alpha: F \Rightarrow G$, we can neatly represent the whole situation thus:

$$\begin{array}{ccc}
 & F & \\
 \curvearrowright & & \curvearrowleft \\
 \mathcal{C} & \Downarrow \alpha & \mathcal{D} \\
 \curvearrowleft & & \curvearrowright \\
 & G &
 \end{array}$$

Now, arrows in a category can be composed to form new arrows (when targets and sources suitably mesh). Functors between categories can be composed to form new functors. Now we see that natural transformations between functors can be composed, in more than one way, to form new natural transformations. We'll run the discussion entirely in terms of transformations between covariant functors: but there will be parallel results about contravariant functors.

Suppose first that we have three functors $F: \mathcal{C} \rightarrow \mathcal{D}, G: \mathcal{C} \rightarrow \mathcal{D}, H: \mathcal{C} \rightarrow \mathcal{D}$, together with two natural transformations $\alpha: F \Rightarrow G$, and $\beta: G \Rightarrow H$.

We can evidently compose these two transformations to get a natural transformation $\beta \circ \alpha: F \Rightarrow H$, defined componentwise by putting $(\beta \circ \alpha)_A = \beta_A \circ \alpha_A$ for all objects A in \mathcal{C} . Vertically gluing together two commuting naturality squares which share a side gives us a bigger commuting square, meaning that for any $f: A \rightarrow B$ in \mathcal{C} , the following commutes in \mathcal{D} :

21.3 Horizontal composition of natural transformations

$$\begin{array}{ccc}
 FA & \xrightarrow{Ff} & FB \\
 \downarrow \alpha_A & & \downarrow \alpha_B \\
 GA & \xrightarrow{Gf} & GB \\
 \downarrow \beta_A & & \downarrow \beta_B \\
 HA & \xrightarrow{Hf} & HB
 \end{array}$$

$\beta_{A \circ \alpha_A}$ (left brace) $\beta_{B \circ \alpha_B}$ (right brace)

Composing two transformations as in $\mathcal{C} \xrightarrow{G} \mathcal{D}$ to get $\mathcal{C} \xrightarrow{H} \mathcal{D}$ is rather predictably called *vertical composition*.

21.3 Horizontal composition of natural transformations

We can, however, also put things together *horizontally* in various ways. First, there is so-called *whiskering* (!) where we combine a functor with a natural transformation between functors to get a new natural transformation. Thus, what happens when we ‘add a whisker’ on the left of a diagram for a natural transformation?

$$\mathcal{C} \xrightarrow{F} \mathcal{D} \xrightarrow{J} \mathcal{E} \quad \text{gives rise to} \quad \mathcal{C} \xrightarrow{J \circ F} \mathcal{E}$$

where the component of βF at A is the component of β at FA – i.e. $(\beta F)_A = \beta_{FA}$ (which is why the suggestive notation ‘ β_F ’ is quite often preferred to ‘ βF ’). Why does this hold? Consider the function $Ff: FA \rightarrow FB$ in \mathcal{D} (where $f: A \rightarrow B$ is in \mathcal{C}). Now apply the functors J and K , and since β is a natural transformation we get the commutative ‘naturality square’

$$\begin{array}{ccc}
 J(FA) & \xrightarrow{J(Ff)} & J(FB) \\
 \downarrow \beta_{FA} & & \downarrow \beta_{FB} \\
 K(FA) & \xrightarrow{K(Ff)} & K(FB)
 \end{array}$$

and we can read that as giving a natural transformation between $J \circ F$ and $K \circ F$.

Likewise, adding a whisker on the right,

Natural transformations

the situation $\mathcal{C} \begin{array}{c} \xrightarrow{F} \\ \Downarrow \alpha \\ \xrightarrow{G} \end{array} \mathcal{D} \xrightarrow{J} \mathcal{E}$ gives rise to $\mathcal{C} \begin{array}{c} \xrightarrow{J \circ F} \\ \Downarrow J\alpha \\ \xrightarrow{J \circ G} \end{array} \mathcal{E}$

where the component of $J\alpha$ at X is $J(\alpha_X)$.

For future use, by the way, we should note the following mini-result:

Theorem 107. *Whiskering a natural isomorphism yields a natural isomorphism.*

Proof. Retaining the same notation as above, but now taking α and β to be isomorphisms, we saw that ‘post-whiskering’ α by J to get $J\alpha$ yields a transformation whose components are $J\alpha_X$, and since functors preserve isomorphisms, these components are all isomorphisms, hence so is $J\alpha$. ‘Pre-whiskering’ β by F to get βF yields a transformation whose components are (some of the) components of β and therefore are isomorphisms, hence again so is βF . \square

(a) Second, we can *horizontally compose* two natural transformations in the following way:

We take $\mathcal{C} \begin{array}{c} \xrightarrow{F} \\ \Downarrow \alpha \\ \xrightarrow{G} \end{array} \mathcal{D} \begin{array}{c} \xrightarrow{J} \\ \Downarrow \beta \\ \xrightarrow{K} \end{array} \mathcal{E}$ and get $\mathcal{C} \begin{array}{c} \xrightarrow{J \circ F} \\ \Downarrow \beta * \alpha \\ \xrightarrow{K \circ G} \end{array} \mathcal{E}$.

How do we define $\beta * \alpha$? Take an arrow $f: A \rightarrow B$ and form this naturality square

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ GA & \xrightarrow{Gf} & GB \end{array} \quad \text{Applying the functor } J, \quad \begin{array}{ccc} J(FA) & \xrightarrow{J(Ff)} & J(FB) \\ \downarrow J(\alpha_A) & & \downarrow J(\alpha_B) \\ J(GA) & \xrightarrow{J(Gf)} & J(GB) \end{array}$$

also commutes. And since $Gf: GA \rightarrow GB$ is a map in \mathcal{D} , and β is a natural transformation between $\mathcal{D} \xrightarrow{J} \mathcal{E}$, we have

$$\begin{array}{ccc} J(GA) & \xrightarrow{J(Gf)} & J(GB) \\ \downarrow \beta_{GA} & & \downarrow \beta_{GB} \\ K(GA) & \xrightarrow{K(Gf)} & K(GB) \end{array}$$

commutes. Gluing together those last two commutative diagrams one above the other gives a natural transformation from $J \circ F$ to $K \circ G$, if we set the component of $\beta * \alpha$ at X to be $\beta_{GX} \circ J\alpha_X$.

Three remarks:

- (1) That definition for $\beta * \alpha$ looks surprisingly asymmetric. But note that applying J to the initial naturality square for α and then pasting the result above a naturality square for β , we could have similarly applied K to the

21.3 Horizontal composition of natural transformations

initial naturality square and pasted the result below another naturality square for β , thus showing that we can alternatively define the natural transformation $J \circ F$ to $K \circ G$ as having the components $K\alpha_X \circ \beta_{FX}$. So symmetry is restored: we get equivalent accounts which mirror each other.

- (2) We can think of whiskering as a special case of the horizontal composition of two natural transformations where one of them is the identity

natural transformation. For example $\mathcal{C} \begin{array}{c} \xrightarrow{F} \\ \Downarrow \alpha \\ \xrightarrow{G} \end{array} \mathcal{D} \begin{array}{c} \xrightarrow{J} \\ \Downarrow 1_J \\ \xrightarrow{J} \end{array} \mathcal{E}$ produces

$\mathcal{C} \begin{array}{c} \xrightarrow{J \circ F} \\ \Downarrow 1_J * \alpha \\ \xrightarrow{J \circ G} \end{array} \mathcal{E}$, and the component of $1_J * \alpha$ at X is an identity com-

posed with $J\alpha_X$. So this is the same as taking the left-hand natural transformation and simply whiskering with J on the right.

- (3) We could now go on to consider the case of horizontally composing a couple of pairs of vertical compositions – and show that it comes to the same if we construe the resulting diagram as the result of vertically composing a couple of horizontal compositions. But we won't now pause over this, but return to the point if and when we ever need the construction. (Or see Leinster 2014, p. 38.)

22 Functor categories

In this chapter, we highlight the observation that the functors between two categories together with the natural transformations between the functors together give us the data for another sort of category!

22.1 Functor categories defined

We saw in §20.3 that for any functor $F: \mathcal{C} \rightarrow \mathcal{D}$, there is an identity natural transformation $1_F: F \Rightarrow F$.

We saw in §21.3 that given parallel functors $F, G, H: \mathcal{C} \rightarrow \mathcal{D}$, then if there are natural transformations $\alpha: F \Rightarrow G$ and $\beta: G \Rightarrow H$ then there is a composite natural transformation $\beta \circ \alpha: F \Rightarrow H$. Moreover, it is immediate from the definition of this ‘vertical’ composition of parallel functors, that composition is associative (that’s because the composition of the arrows which are components of a transformation is associative).

So, lo and behold, the following definition must be in good order!

Definition 102. The *functor category* from \mathcal{C} to \mathcal{D} , denoted $[\mathcal{C}, \mathcal{D}]$ is the category whose objects are all the (covariant) functors $F: \mathcal{C} \rightarrow \mathcal{D}$, with the natural transformations between them as arrows. \triangle

The laconic notation here ‘ $[\mathcal{C}, \mathcal{D}]$ ’ is standard. An alternative is ‘ $\mathcal{D}^{\mathcal{C}}$ ’. (We needn’t worry about a category of contravariant functors as we can always talk about a category $[\mathcal{C}^{op}, \mathcal{D}]$ instead.)

We will see many instances of functor categories at work later. But let’s pause now for a pair of simple examples:

- (1) Recall the discrete category $\bar{2}$, which comprises just two objects \bullet and \star together with their identity arrows. Ask: what is the functor category $[\bar{2}, \mathcal{C}]$?

An object in this category is a functor $F: \bar{2} \rightarrow \mathcal{C}$, where (i) F_{ob} will send \bullet to some \mathcal{C} -object X and send \star to an object Y , and (ii) F_{arw} will map the identity arrows on \bullet and \star to the identity arrows on this X and Y . So (A) there is a simple bijection between such functors F , the objects of $[\bar{2}, \mathcal{C}]$, and pairs of \mathcal{C} -objects (X, Y) .

22.2 Functor categories and natural isomorphisms

What about the arrows of our functor category? By definition, each component of a natural transformation from F to the parallel functor F' will be a \mathcal{C} -arrow between the F -image and the F' -image of some object in $\bar{2}$. And since there are no arrows between those objects in $\bar{2}$ there is no naturality square to impose additional constraints. Therefore (B) a natural transformation from F to F' , an arrow of $[\bar{2}, \mathcal{C}]$, is simply any pair of \mathcal{C} -arrows $(j: X \rightarrow X', k: Y \rightarrow Y')$.

So in sum, by (A) and (B), our new category is (or strictly speaking, is isomorphic to) the product category $\mathcal{C} \times \mathcal{C}$ which we met in §4.2.

- (2) Recall now the category 2 . Omitting identity arrows, we can diagram this as $\bullet \longrightarrow \star$. Ask: what is the functor category $[2, \mathcal{C}]$?

An object in this category is a functor $F: 2 \rightarrow \mathcal{C}$, where (i) F_{ob} will send \bullet to some \mathcal{C} -object X and send \star to an object Y , and (ii) F_{are} will map identity arrows to identity arrows and send the unique arrow from \bullet to \star to some \mathcal{C} -arrow $f: X \rightarrow Y$. This time, (A) there is therefore a simple bijection between the objects of $[2, \mathcal{C}]$ and \mathcal{C} -arrows.

And what about the arrows in our new category? A natural transformation from F to the parallel functor F' will have as components any two \mathcal{C} -arrows, j, k , which makes this a commutative square:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \downarrow j & & \downarrow k \\ X' & \xrightarrow{f'} & Y' \end{array}$$

Thus (B) the arrows of the new category are exactly pairs of \mathcal{C} -arrows which make our relevant diagram commute.

So in sum, by (A) and (B), $[2, \mathcal{C}]$ is (or strictly speaking, is isomorphic to) the arrow category $\mathcal{C}^{\rightarrow}$ we met in §4.3).

22.2 Functor categories and natural isomorphisms

Suppose $[\mathcal{C}, \mathcal{D}]$ is a functor category. Then there will be isomorphisms in this category, in the usual categorial sense of ‘isomorphism’ – i.e. arrows which have inverses. Now, how do these isomorphisms in $[\mathcal{C}, \mathcal{D}]$ relate to the natural isomorphisms we defined between \mathcal{C} and \mathcal{D} as we defined them before?

Theorem 108. *The isomorphisms in the functor category $[\mathcal{C}, \mathcal{D}]$ are exactly the natural isomorphisms $\psi: F \Rightarrow G$, where $\mathcal{C} \xrightarrow[F]{G} \mathcal{D}$.*

Proof. Suppose $\psi: F \xRightarrow{\sim} G$ is a natural isomorphism between the parallel functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$, in the sense of Defn. 99. So for any $f: A \rightarrow B$, the naturality square

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow \psi_A & & \downarrow \psi_B \\ GA & \xrightarrow{Gf} & GB \end{array}$$

commutes. But if $\psi_B \circ Ff = Gf \circ \psi_A$, then $Ff \circ \psi_A^{-1} = \psi_B^{-1} \circ Gf$ (relying on the fact that the components of ψ have inverses). Which makes this always commute for any $f: A \rightarrow B$:

$$\begin{array}{ccc} GA & \xrightarrow{Gf} & GB \\ \downarrow \psi_A^{-1} & & \downarrow \psi_B^{-1} \\ FA & \xrightarrow{Ff} & FB \end{array}$$

Whence $\psi^{-1}: G \xrightarrow{\cong} F$ (where ψ^{-1} is assembled from the components ψ_A^{-1} etc. And trivially $\psi^{-1} \circ \psi = 1_F$ and $\psi \circ \psi^{-1} = 1_G$. Which makes ψ an isomorphism (an arrow with an inverse) in the functor category $[\mathcal{C}, \mathcal{D}]$.

Conversely, suppose the natural transformation $\psi: F \Rightarrow G$ has an inverse ψ^{-1} in the category $[\mathcal{C}, \mathcal{D}]$, i.e. $\psi^{-1} \circ \psi = 1_F$, and $\psi \circ \psi^{-1} = 1_G$. But vertical composition of natural transformations is defined component-wise, so this requires for each component that $\psi_X^{-1} \circ \psi_X = 1_{FX}$, $\psi_X \circ \psi_X^{-1} = 1_{GX}$. Therefore each component of ψ has an inverse, so is an isomorphism, and hence ψ is a natural isomorphism. \square

22.3 Hom-functors from functor categories

We have now introduced a new kind of category – namely, functor categories $[\mathcal{C}, \mathcal{D}]$ whose objects are the functors from \mathcal{C} to \mathcal{D} , and whose arrows are the natural transformations between those functors. As with any other category, there can be functors mapping to and from such categories to other categories. Some of these will later turn out to be of central importance in category theory. We start exploring in the rest of this chapter.

Suppose we have a functor category $[\mathcal{C}, \mathcal{D}]$. Its arrows, by definition, are natural transformations. And the collection of natural transformations from the functor $F: \mathcal{C} \rightarrow \mathcal{D}$ to $G: \mathcal{C} \rightarrow \mathcal{D}$, assuming it is set-sized, will be the hom-set $[\mathcal{C}, \mathcal{D}](F, G)$. We will repeatedly meet such hom-sets: it will therefore be handy to have a slightly more memorable alternative notation for them:

Definition 103. ‘ $Nat(F, G)$ ’ will denote the set of natural transformations from F to G (assuming it exists). \triangle

Now, where there are hom-sets, there are hom-functors. Again we introduce some snappier notation for future use:

Definition 104. ‘ $Nat(-, G)$ ’ denotes the contravariant hom-functor $[\mathcal{C}, \mathcal{D}](-, G): [\mathcal{C}, \mathcal{D}] \rightarrow \mathbf{Set}$; ‘ $Nat(F, -)$ ’ denotes the covariant hom-functor $[\mathcal{C}, \mathcal{D}](F, -)$. \triangle

Let’s pause to consider how such functors work. Take the first of them, for example. We simply apply the definition of a contravariant hom-functor. So $Nat(-, G)$ sends an object in the functor category $[\mathcal{C}, \mathcal{D}]$, i.e. a functor F , to the set $Nat(F, G)$. And it sends an arrow in the functor category, i.e. a natural transformation $\alpha: F' \Rightarrow F$, to a set-function from $Nat(F, G)$ to $Nat(F', G)$ – i.e. to the function that sends a natural transformation $\beta: F \Rightarrow G$ to $\beta \circ \alpha: F' \Rightarrow G$. (Note, if that latter function is indeed to live happily in \mathbf{Set} , we must be officially thinking of natural transformations, defined as families of arrows, as themselves properly speaking sets.)

22.4 Evaluation and diagonal functors

(a) Start again with the functor category $[\mathcal{C}, \mathcal{D}]$ and this time also pick an object A in \mathcal{C} . Then there is a functor that looks at what is in $[\mathcal{C}, \mathcal{D}]$ and evaluates it at A :

Definition 105. The functor $ev_A: [\mathcal{C}, \mathcal{D}] \rightarrow \mathcal{D}$ sends a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ to FA and sends a natural transformation $\alpha: F \Rightarrow G$ to $\alpha_A: FA \rightarrow GA$. \triangle

It is trivial to check that ev_A really is functorial.

(b) Now let’s consider a functor which goes in the opposite direction, i.e. one that maps *to* a functor category. We will suppose then that \mathcal{C} is a category, and \mathbf{J} is a small category. Then

Definition 106. The functor $\Delta_{\mathbf{J}}: \mathcal{C} \rightarrow [\mathbf{J}, \mathcal{C}]$ sends an object C to the functor $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$ and sends an arrow $f: C \rightarrow C'$ to the natural transformation from Δ_C to $\Delta_{C'}$ whose every component is simply f again. \triangle

Recall, Δ_C is the constant collapse-to- C functor we first met in §15.2 (F10). To check that $\Delta_{\mathbf{J}}$ is indeed a functor, the crucial thing is to show the last part of our definition does indeed characterize a natural transformation from Δ_C to $\Delta_{C'}$. For this, we just note that for every $d: K \rightarrow L$ in \mathbf{J} , the required naturality square on the left is in fact none other than the trivially commuting square on the right:

$$\begin{array}{ccc}
 \Delta_C K & \xrightarrow{\Delta_C d} & \Delta_C L \\
 \downarrow f & & \downarrow f \\
 \Delta_{C'} K & \xrightarrow{\Delta_{C'} d} & \Delta_{C'} L
 \end{array}
 \qquad
 \begin{array}{ccc}
 C & \xrightarrow{1_C} & C \\
 \downarrow f & & \downarrow f \\
 C' & \xrightarrow{1_{C'}} & C'
 \end{array}$$

Such a functor $\Delta_{\mathbf{J}}$ is often called a diagonal functor. Why? We are generalizing on the case where \mathbf{J} is the discrete two-object category $\bar{2}$ with objects $0, 1$. Here, $\Delta_{\bar{2}}$ sends an object C in \mathcal{C} to a functor that sends 0 to C and sends 1 to C . If we think of that latter functor as therefore representing a pair of outcomes (C, C) , then the functor $\Delta_{\bar{2}}$ in effect sends C to (C, C) . In other words, values of $\Delta_{\bar{2}}$ lie down the diagonal of pairs of \mathcal{C} -objects.

(c) Given the functor $\Delta_{\mathbf{J}}: \mathcal{C} \rightarrow [\mathbf{J}, \mathcal{C}]$, and an object D in $[\mathbf{J}, \mathcal{C}]$ (i.e. a diagram $D: \mathbf{J} \rightarrow \mathcal{C}$), there will be a comma category $(\Delta_{\mathbf{J}} \downarrow D)$. Applying the definition of such a category at the end of §19.4, we get the following:

- (1) An object of $(\Delta_{\mathbf{J}} \downarrow D)$ is a pair of an object C in \mathcal{C} , and an arrow $c: \Delta_{\mathbf{J}}C \rightarrow D$ in $[\mathbf{J}, \mathcal{C}]$, i.e. a natural transformation from Δ_C to D . But the components of such a natural transformation we saw in §21.1 109 are just the legs c_J of a cone over D with vertex C . So an object (C, c) of our category are in effect just a cone $[C, c_J]$ over D , i.e. an object in the category of cones over D .
- (2) An arrow of $(\Delta_{\mathbf{J}} \downarrow D)$ from (C, c) to (C', c') is a \mathcal{C} -arrow $f: C \rightarrow C'$ such that $c = c' \circ \Delta_{\mathbf{J}}f$, which says that for each J , $c_J = c'_J \circ f$. Which is just the condition for f to be an arrow between cones $[C, c_J]$ and $[C', c'_J]$ in the category of cones over D in Defn. 57.

Hence $(\Delta_{\mathbf{J}} \downarrow D)$ is just the category of cones over D ! Which is neat. We can then say that a cone over D is just an object of the category $(\Delta_{\mathbf{J}} \downarrow D)$; and a limit over D is a terminal object of this category.

It will be no additional surprise to learn that $(D \downarrow \Delta_{\mathbf{J}})$ is the category of cocones under D .

22.5 Cones as natural transformations

(a) Fix on some small category \mathbf{J} . Consider the functor category $[\mathbf{J}, \mathcal{C}]$ whose objects are diagrams-as-functors $D: \mathbf{J} \rightarrow \mathcal{C}$ and whose arrows are natural transformations between such functors.

One particular kind of object in $[\mathbf{J}, \mathcal{C}]$ is a trivial constant functor such as $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$, i.e. the functor that sends every object in \mathbf{J} to the object C and every arrow in \mathbf{J} to 1_C .

Now, what would be a natural transformation from Δ_C to another diagram-as-functor D ? Applying the definition, it would be a family α of \mathbf{J} arrows $\alpha_J: \Delta_C(J) \rightarrow D(J)$ indexed by $J \in \mathbf{J}$, i.e. arrows $\alpha_J: C \rightarrow D(J)$, such that for every $d: K \rightarrow L$ in \mathbf{J} , the square below always commutes in \mathcal{C} . Hence, trivially, so does the triangle:

$$\begin{array}{ccc}
 C & \xrightarrow{1_C} & C \\
 \alpha_K \downarrow & & \downarrow \alpha_L \\
 D(K) & \xrightarrow{D(d)} & D(L)
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 & C & \\
 \alpha_K \swarrow & & \searrow \alpha_L \\
 D(K) & \xrightarrow{D(d)} & D(L)
 \end{array}$$

But we recognize that! It means that C together with the α_J form a cone over D . And conversely, of course, the arrows α_J in any cone over D with vertex C form a natural transformation $\alpha: \Delta_C \Rightarrow D$. So in sum:

Theorem 109. *A cone over $D: \mathbf{J} \rightarrow \mathcal{C}$ with vertex C comprises C together with a natural transformation from the trivial functor $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$ to D .*

If we think of cones the more austere way (i.e. just as a family of arrows – see §10.1), then we can take $\text{Cone}(C, D)$, the collection of cones over D with vertex C , to be simply $[\mathbf{J}, \mathcal{C}](\Delta_C, D)$, i.e. the hom-set of arrows in the functor category $[\mathbf{J}, \mathcal{C}]$ from Δ_C to D .

(b) We can think of the functor Δ_C (living as an object in the functor category $[\mathbf{J}, \mathcal{C}]$) as itself the value at the object C of a functor $\Delta: \mathcal{C} \rightarrow [\mathbf{J}, \mathcal{C}]$.

To be functorial, how must Δ act on an arrow $f: C \rightarrow D$ in \mathcal{C} ? It must send f to an arrow, i.e. a natural transformation, from $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$ to $\Delta_D: \mathbf{J} \rightarrow \mathcal{C}$. But just by definition, a natural transformation α from Δ_C to Δ_D is a suite of arrows α_J indexed by objects $J \in \mathbf{J}$ such that for any $j: K \rightarrow L$ in \mathbf{J} , these diagrams commute

$$\begin{array}{ccc}
 \Delta_C(K) & \xrightarrow{\Delta_C(j)} & \Delta_C(L) \\
 \downarrow \alpha_K & & \downarrow \alpha_L \\
 \Delta_D(K) & \xrightarrow{\Delta_D(j)} & \Delta_D(L)
 \end{array}
 \quad \text{i.e.} \quad
 \begin{array}{ccc}
 C & \xrightarrow{1_C} & C \\
 \downarrow \alpha_K & & \downarrow \alpha_L \\
 D & \xrightarrow{1_D} & D
 \end{array}$$

Therefore every component of α must be equal and we'll have to put all of them equal to f (the only arrow from C to D we are given!). The resulting action of Δ on f is easily checked to be functorial.

22.6 Limit functors

(a) Suppose every diagram D of shape \mathbf{J} has a limit in \mathcal{C} . Then we can define a functor $\text{Lim}: [\mathbf{J}, \mathcal{C}] \rightarrow \mathcal{C}$ which sends a diagram D living in the functor category $[\mathbf{J}, \mathcal{C}]$ to the vertex $\text{Lim}_{\leftarrow \mathbf{J}} D$ for some chosen limit cone over D in \mathcal{C} .

But note however that we do have to do some choosing here! This functor is not entirely 'naturally' or canonically defined: for recall, in the general case, limits over D are only unique up to isomorphism, so we will indeed have to select a particular limit object $\text{Lim}_{\leftarrow \mathbf{J}} D$ to be the value of our functor for input D .

Functor categories

And we need to say more. To get a functor, we now need suitably to define $\lim_{\leftarrow J}$'s action on arrows. This must send an arrow in $[J, \mathcal{C}]$, i.e. a natural transformation $\alpha: D \Rightarrow D'$ to an arrow in \mathcal{C} from $\lim_{\leftarrow J} D$ to $\lim_{\leftarrow J} D'$. How can it do this in a, well, natural way? By hypothesis there are limit cones over D and D' , respectively $[\lim_{\leftarrow J} D, \pi_J]$ and $[\lim_{\leftarrow J} D', \pi'_J]$. So now take any arrow $d: K \rightarrow L$ living in J and consider the following diagram:

$$\begin{array}{ccccc}
 & & \lim_{\leftarrow J} D & & \\
 & \swarrow \pi_K & \downarrow u_\alpha & \searrow \pi_L & \\
 D(K) & \xrightarrow{D(d)} & & \xrightarrow{\quad} & D(L) \\
 \downarrow \alpha_K & & \downarrow & & \downarrow \alpha_L \\
 & \swarrow \pi'_K & \lim_{\leftarrow J} D' & \searrow \pi'_L & \\
 D'(K) & \xrightarrow{D'(d)} & & \xrightarrow{\quad} & D'(L)
 \end{array}$$

The top triangle commutes (because $[\lim_{\leftarrow J} D, \pi_J]$ is a limit). The lower square commutes by the naturality of α . Therefore the outer pentangle commutes and so, generalizing over objects J in J , $[\lim_{\leftarrow J} D, \alpha_J \circ \pi_J]$ is a cone over D' . But then *this* cone must factor uniquely through D' 's limit cone $[\lim_{\leftarrow J} D', \pi'_J]$ via some unique $u_\alpha: \lim_{\leftarrow J} D \rightarrow \lim_{\leftarrow J} D'$. The map $\alpha \mapsto u_\alpha$ is then a plausible candidate for $\lim_{\leftarrow J}$'s action on arrows; and indeed this assignment is fairly easily checked to yield a functor.

In summary then:

Definition 107. Assuming every diagram D of shape J has a limit in \mathcal{C} , the functor $\lim_{\leftarrow J}: [J, \mathcal{C}] \rightarrow \mathcal{C}$ (or \lim for brief)

- i. sends an object D in $[J, \mathcal{C}]$ to the vertex $\lim D$ of some chosen limit cone $[\lim D, \pi_J]$ over D
- ii. sends an arrow $\alpha: D \Rightarrow D'$ in $[J, \mathcal{C}]$ to the arrow $u_\alpha: \lim D \rightarrow \lim D'$ where for all J in J , $\pi'_J \circ u_\alpha = \alpha_J \circ \pi_J$. \triangle

The diagram above can be recycled, by the way, to show

Theorem 110. Assuming limits of the relevant shape exist then, if we have a natural isomorphism $D \cong D'$, $\lim D \cong \lim D'$.

Proof. Because we now have a natural isomorphism $D \cong D'$, we can show as above both that there is a unique $u: \lim D \rightarrow \lim D'$ and symmetrically that there is a unique $u': \lim D' \rightarrow \lim D$. These compose to give us map

$u' \circ u: \text{Lim } D \rightarrow \text{Lim } D$ which must be $1_{\text{Lim } D}$ by the now familiar argument (the limit cone with vertex $\text{Lim } D$ can factor through itself by both $u' \circ u$ and $1_{\text{Lim } D}$, but there is by hypothesis only one way for the limit cone to factor through itself). Likewise, $u \circ u' = 1_{\text{Lim } D'}$. So u is an isomorphism. \square

(b) We now remark on the following simple theorem:

Theorem 111. *Suppose that \mathcal{C} has all limits of shape J . Then for any $D: J \rightarrow \mathcal{C}$ which the functor $F: \mathcal{C} \rightarrow \mathcal{D}$ preserves,*

$$(*) \quad F(\underset{\leftarrow J}{\text{Lim}} D) \cong \underset{\leftarrow J}{\text{Lim}}(F \circ D).$$

In brief: F commutes with $\underset{\leftarrow J}{\text{Lim}}$.

Proof. Since \mathcal{C} has all limits of shape J , the limit functor Lim (for short) is well-defined.

Now, if F preserves a limit cone over $D: J \rightarrow \mathcal{C}$ with vertex $\text{Lim } D$, then F sends that limit cone to a limit cone over $F \circ D$ with vertex $F(\text{Lim } D)$. But that vertex must be isomorphic to the vertex of any other limit cone over $F \circ D$. So in particular it must be isomorphic to whatever has been chosen to be $\underset{\leftarrow J}{\text{Lim}}(F \circ D)$. \square

We will have occasion to return to consider the behaviour of limit functors at greater length. For the moment, however, we just recall a slogan from elementary analysis; ‘continuous functions commute with limits’. Which explains a bit of standard terminology you might come across:

Definition 108. A functor which commutes with limits of shape J for all small categories J is said to be *continuous*. \triangle

23 Equivalent categories

We defined what it is for categories to be isomorphic in §16.5, and gave a number of examples. However, as we announced at the time, there are cases of categories that surely ‘come to just the same’ (in some good intuitive sense) but which are not isomorphic. A weaker notion of equivalence of categories turns out to be more useful. It is defined using the notion of a natural transformation, which explains why we have had to wait to now to talk about equivalence.

23.1 The categories Pfn and Set_* are ‘equivalent’

In the general theory of computation, there is no getting away from the central importance of the notion of a partial function from \mathbb{N} to \mathbb{N} (for example, the function φ_e computed by the e -th Turing machine in a standard enumeration is typically partial).

But how should we treat partial functions in logic? Suppose the partial computable function $\varphi: \mathbb{N} \rightarrow \mathbb{N}$ takes no value for n (the algorithm defining φ doesn’t terminate gracefully for input n). Then the term ‘ $\varphi(n)$ ’ apparently lacks a denotation. *But in standard first-order logic, all terms are assumed to denote.* Two-valued logic requires every sentence to be determinately either true or false and truth-value gaps are not allowed: but a sentence with a non-denoting term, on the standard semantics, will lack a truth-value. What to do?

Historically, there are a number of options on the market for dealing with empty terms in a regimented logical language, and hence for dealing with the partial functions which give rise to them. Here we mention just two. One strategy – due to the greatest nineteenth century logician, Gottlob Frege – is to stipulate that apparently empty terms are in fact not empty at all but denote some special object. Then there are no empty terms and no truth-value gaps, hence we can preserve standard logic. An alternative, less artificial, route forward is to bite the bullet and change our logic to allow non-denoting terms and then cope with the truth-value gaps which come along with them.

In just a bit more detail:

- (1) *Frege’s proposal* The idea, to repeat, is to provide apparently empty terms a default ‘rogue’ object for them to denote. Apparently empty terms

23.1 The categories Pfn and Set_\star are ‘equivalent’

are only superficially so: they are still genuine referring terms, but with a deviant denotation.

How does this work for functional terms? Well, given what we naively think of as a partial function $\varphi: \mathbb{N} \rightarrow \mathbb{N}$, we now treat this as officially being a *total* function $f: \mathbb{N} \cup \{\star\} \rightarrow \mathbb{N} \cup \{\star\}$, where \star is any convenient non-number, and where $f(n) = \varphi(n)$ when $\varphi(n)$ takes a numerical value, and f takes the value \star otherwise. If you like, you can think of \star as coding ‘not numerically defined’.

So, on this approach our functions are all total. What we *really* meet in a formalized theory of computation are total functions which are only partially numerical (not all their values are numbers). Because these total functions don’t generate non-denoting terms, we can preserve our standard logic without truth-value gaps.

- (2) *Logical revisionism* Alternatively, we can bite the bullet and live with truth-value gaps, as we surely already do in informal reasoning.

That means, when we come to adopt an official formalized logic, we’ll want one which is free from the assumption that all terms denote; we will adopt a *free logic* for short. We will then have to give new accounts for the logical operators to tell us how they behave when they encounter truth-value gaps – for example, if P is truth-valueless because it contains a non-denoting term, is *not- P* also truth-valueless or is it true because P isn’t true?

The details can get a little messy, and this logical revisionism has its costs and complications. But at least in a formalism with a free logic we can take at face value both partial functions and the apparently empty terms they give rise to.

There is a lot more to be said: and we could, for example, consider a third proposal due to Bertrand Russell which eliminates empty terms in a different way to Frege. But we won’t continue the story any further now: the debate about the best logical treatment of partial functions is the sort of thing that might grip some philosophically-minded logicians but really seems of very little general mathematical interest.

And that’s exactly the point of this section! From a mathematical point of view there surely isn’t anything much to choose between logical revisionism and Frege’s more artificial but more conservative proposal.

On the small scale, we can think of a world of genuinely *partial* numerical functions $\varphi: \mathbb{N} \rightarrow \mathbb{N}$ (genuinely partial because not everywhere defined, and hence giving rise to empty terms), or we can equally think of a corresponding world of *total* functions $f: \mathbb{N} \cup \{\star\} \rightarrow \mathbb{N} \cup \{\star\}$, with $\star \notin \mathbb{N}$, and $f(\star) = \star$. Take your pick! More generally, on the large scale, we can think of sets with partial functions between them, or of corresponding pointed sets (sets with a distinguished object as base point) and base-point preserving total functions between *them*. What’s to choose, apart from familiarity? Mathematically, surely

both approaches come to the same.

And so back to categories! There is a category \mathbf{Pfn} whose objects are sets and whose arrows are (possibly) *partial* functions between them. And there is also the category \mathbf{Set}_* of pointed sets whose objects are sets with a distinguished base point, and whose arrows are (total) set-functions which preserve base points. We can work equally well in either category. So, putting the upshot of our reflections in this section in categorial terms, we get the following attractive

Desideratum An account of what it is for two categories to be equivalent should surely count \mathbf{Set}_* and \mathbf{Pfn} as being so, for mathematically they come to the same.

23.2 \mathbf{Pfn} and \mathbf{Set}_* are not isomorphic

In §16.5 we saw that some examples of categories which ‘come to just the same’ are in fact isomorphic. However, we can now show:

Theorem 112. \mathbf{Set}_* is not isomorphic to \mathbf{Pfn} .

We can remark that there *is* an obvious functor $F: \mathbf{Set}_* \rightarrow \mathbf{Pfn}$. F sends a pointed set (X, x) to the set $X \setminus \{x\}$, and sends a base-point preserving total function $f: (X, x) \rightarrow (Y, y)$ to the partial function $\varphi: X \setminus \{x\} \rightarrow Y \setminus \{y\}$, where $\varphi(x) = f(x)$ if $f(x) \in Y \setminus \{y\}$, and is undefined otherwise. But, nice though this is, F isn’t an isomorphism (it could send distinct (X, x) and (X', x') to the same target object).

Again, there is a whole family of functors from \mathbf{Pfn} to \mathbf{Set}_* which take any set X and add an element not yet in X to give as an expanded set with the new object as a basepoint. Here’s a way of doing this in a uniform way without making arbitrary choices for each X . Define $G: \mathbf{Pfn} \rightarrow \mathbf{Set}_*$ as sending a X to the pointed set $X_* =_{\text{def}} (X \cup \{X\}, X)$, remembering that in standard set theories $X \notin X$! And then let G send a partial function $\varphi: X \rightarrow Y$ to the total basepoint-preserving function $f: X_* \rightarrow Y_*$, where $f(x) = \varphi(x)$ if $\varphi(x)$ is defined and $f(x) = \{Y\}$ otherwise. G is a natural choice, but isn’t an isomorphism (it isn’t surjective on objects).

Still, those observations don’t yet rule out there being *some* pair of functors between \mathbf{Set}_* and \mathbf{Pfn} which are mutually inverse. But there can’t be any such pair.

Proof. A functor which is an isomorphism from \mathbf{Pfn} to \mathbf{Set}_* must send objects in \mathbf{Pfn} one-to-one to objects in \mathbf{Set}_* , and must send isomorphisms to isomorphisms, so should preserve the cardinality of isomorphism classes. But the isomorphism class of the empty set in \mathbf{Pfn} has just one member, while there is no one-membered isomorphism class in \mathbf{Set}_* . So there can’t be an isomorphism between the categories. \square

23.3 Equivalent categories

(a) The last two sections have together shown that there are categories \mathbf{Pfn} and \mathbf{Set}_* which to all intents and purposes are mathematically equivalent but which aren't isomorphic (according to the natural definition of isomorphism for categories).

We did, however, note an obvious choice of functors $F: \mathbf{Set}_* \rightarrow \mathbf{Pfn}$ and $G: \mathbf{Pfn} \rightarrow \mathbf{Set}_*$. And while GF isn't the identity on \mathbf{Set}_* it *does* map \mathbf{Set}_* to itself in a rather natural way (without arbitrary choices).

Reflection on this case suggests the following weakening of the definition of isomorphism between categories:

Definition 109. Categories \mathcal{C} and \mathcal{D} are *equivalent*, in symbols $\mathcal{C} \simeq \mathcal{D}$, iff there are functors $F: \mathcal{C} \rightarrow \mathcal{D}$ and $G: \mathcal{D} \rightarrow \mathcal{C}$, together with a pair of natural isomorphisms $\alpha: 1_{\mathcal{C}} \Rightarrow GF$ and $\beta: FG \Rightarrow 1_{\mathcal{D}}$.

We can now give a direct proof that \mathbf{Pfn} and \mathbf{Set}_* are indeed equivalent in this way (try it!).

But in fact we won't do this. Rather, we'll first prove a result which yields an alternative characterization of equivalence which is often much easier to apply:

Theorem 113. *Assuming a sufficiently strong choice principle, a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ is part of an equivalence between \mathcal{C} and \mathcal{D} iff F is faithful, full and essentially surjective on objects.*

Proof. First suppose F is part of an equivalence between \mathcal{C} and \mathcal{D} , so that there is a functor $G: \mathcal{D} \rightarrow \mathcal{C}$, where $GF \cong 1_{\mathcal{C}}$ and $FG \cong 1_{\mathcal{D}}$. Then:

- (i) Given an arrows $f, g: A \rightarrow B$ in \mathcal{C} , then by hypothesis, the following square commutes for f (α is the required natural isomorphism between the identity functor and the composite GF),

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ GFA & \xrightarrow{GFf} & GFB \end{array}$$

and hence $\alpha_B^{-1} \circ GFf \circ \alpha_A = f$. And of course $\alpha_B^{-1} \circ GFg \circ \alpha_A = g$. It immediately follows that if $Ff = Fg$ then $f = g$, i.e. F is faithful. A companion argument, interchanging the roles of \mathcal{C} and \mathcal{D} , shows that G too is faithful.

- (ii) Suppose we are given an arrow $h: FA \rightarrow FB$, then put $f = \alpha_B^{-1} \circ Gh \circ \alpha_A$. But we know that $f = \alpha_B^{-1} \circ GFf \circ \alpha_A$. So it follows that $GFf = Gh$, and since G is faithful, $h = Ff$. So every such h in \mathcal{D} is the image under F of some arrow in \mathcal{C} . So F is full.

Equivalent categories

- (iii) Recall, $F: \mathcal{C} \rightarrow \mathcal{D}$ is e.s.o. iff for any $D \in \mathcal{D}$ we can find some isomorphic object FC , for $C \in \mathcal{C}$. But we know that our natural isomorphism between $1_{\mathcal{D}}$ and FG means that there is an isomorphism from D to FGD , so putting $C = GD$ gives the desired result that F is e.s.o.

Now for the argument in the other direction. Suppose, then, that $F: \mathcal{C} \rightarrow \mathcal{D}$ is faithful, full and e.s.o. We need to construct (iv) a corresponding functor $G: \mathcal{D} \rightarrow \mathcal{C}$, and then a pair of natural isomorphisms (v) $\beta: FG \Rightarrow 1_{\mathcal{D}}$ and (vi) $\alpha: 1_{\mathcal{C}} \Rightarrow GF$:

- (iv) By hypothesis, F is e.s.o., so by definition every object $D \in \mathcal{D}$ is isomorphic in \mathcal{D} to some object FC , for $C \in \mathcal{C}$. Hence – and here we are invoking an appropriate choice principle – for any given $D \in \mathcal{D}$, we can choose a pair (C, β_D) , with $C \in \mathcal{C}$ and $\beta_D: FC \rightarrow D$ an isomorphism in \mathcal{D} . Now define G_{ob} as sending an object $D \in \mathcal{D}$ to the chosen $C \in \mathcal{C}$ (so $GD = C$, and $\beta_D: FGD \rightarrow D$).

To get a functor, we need the component G_{arw} to act suitably on an arrow $g: D \rightarrow E$. Now, note

$$FGD \xrightarrow{\beta_D} D \xrightarrow{g} E \xrightarrow{\beta_E^{-1}} FGE$$

and since F is full and faithful, there must be some unique $f: GD \rightarrow GE$ which F sends to the composite $\beta_E^{-1} \circ g \circ \beta_D$. Put $G_{arw}g = f$.

Claim: G , with components G_{ob} , G_{arw} , is indeed a functor. We need to show that G (a) preserves identities and (b) respects composition:

For (a), note that $G1_D = e$ where e is the unique arrow from GD to GD such that $Fe = \beta_D^{-1} \circ 1_D \circ \beta_D = 1_{FGD}$. So $e = 1_{GD}$.

For (b) we need to show that, given \mathcal{D} -arrows $g: D \rightarrow E$ and $h: E \rightarrow F$, $G(h \circ g) = Gh \circ Gg$. But note that

$$\begin{aligned} FG(h \circ g) &= \beta_F^{-1} \circ h \circ g \circ \beta_D &= (\beta_F^{-1} \circ h \circ \beta_E) \circ (\beta_E^{-1} \circ g \circ \beta_D) \\ &= FG(h) \circ FG(g) = F(G(h) \circ G(g)) \end{aligned}$$

Hence, since $FG(h \circ g) = F(G(h) \circ G(g))$ and F is faithful, $G(h \circ g) = G(h) \circ G(g)$, so G is indeed a functor.

- (v) By construction, β is natural isomorphism from FG to $1_{\mathcal{D}}$.
- (vi) Note next that we have an isomorphism $\beta_{FA}^{-1}: FA \rightarrow FGFA$. As F is full and faithful, $\beta_{FA}^{-1} = F(\alpha_A)$ for some unique $\alpha_A: A \rightarrow GFA$. Since F is fully faithful it is conservative, i.e. reflects isomorphisms (by Theorem 80), hence α_A is also an isomorphism. Also, the naturality diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow \alpha_A & & \downarrow \alpha_B \\ GFA & \xrightarrow{GFf} & GFB \end{array}$$

always commutes for any arrow $f: A \rightarrow B$ in \mathcal{C} . Why? Because

$$\begin{aligned} F(\alpha_B \circ f) &= F\alpha_B \circ Ff = \beta_{FB}^{-1} \circ Ff = \\ &FGFf \circ \beta_{FA}^{-1} = FGFf \circ F\alpha_A = F(GFf \circ F\alpha_A) \end{aligned}$$

relying on the naturality of β^{-1} . But if $F(\alpha_B \circ f) = F(GFf \circ F\alpha_A)$ then since F is faithful, $\alpha_B \circ f = GFf \circ F\alpha_A$. Hence the α_A are the components of our desired natural isomorphism $\alpha: 1_{\mathcal{C}} \Rightarrow GF$.

So we are done! □

Our theorem enables us now to very quickly prove the following equivalence claim without any more hard work:

Theorem 114. $\text{Pfn} \simeq \text{Set}_*$

Proof. Define the functor $G: \text{Pfn} \rightarrow \text{Set}_*$ as before. It sends a set X to a set $X_* =_{\text{def}} X \cup \{X\}$ with basepoint X , and sends a partial function $f: X \rightarrow Y$ to the total function $f_*: X_* \rightarrow Y_*$, where for $f_*(x) = f(x)$ if $f(x)$ is defined and $f_*(x) = Y$ otherwise.

G is faithful, as it is easily checked that it sends distinct functions to distinct functions. And it is equally easy to check that G is full, i.e. given any basepoint preserving function between sets X_* and Y_* , there is a partial function f which G sends to it.

But G is essentially surjective on objects. For every pointed set in Set_* – i.e. every set which can be thought of as the union of a set X with $\{*\}$ where $*$ is an additional basepoint element (not in X) – is isomorphic in Set_* to the set $X \cup \{X\}$ with X as basepoint. Hence G is part of an equivalence between Pfn and Set_* . □

(b) Now for another example. Recall FinSet is the category of finite sets and functions between them. Let FinOrdn be its full subcategory containing the empty set and all sets of the form $\{0, 1, 2, \dots, n-1\}$ and all functions between them. It doesn't really matter for present purposes how you think of the natural numbers; but to fix ideas, think of them set-theoretically as von Neumann ordinals, so the objects of FinOrdn are then the finite ordinals – hence the label for the category. We then have:

Theorem 115. $\text{FinOrdn} \simeq \text{FinSet}$

Proof. FinOrdn is a full subcategory of FinSet , so the inclusion functor F is fully faithful. F is also essentially surjective on objects: for take any object in FinSet , which is some n -membered set: that is in bijective correspondence (and hence isomorphic in FinSet) with the finite ordinal n . Hence F is part of an equivalence, and $\text{FinOrdn} \simeq \text{FinSet}$. □

Equivalent categories

How should we regard this last result? We saw that defining equivalence of categories in terms of isomorphism would be too strong, as it rules out our treating \mathbf{Pfn} and \mathbf{Set}_* as in effect equivalent. But now we've seen that defining equivalence of categories as in Defn. 23.3 makes the seemingly very sparse category \mathbf{FinOrd} equivalent to the seemingly much more abundant \mathbf{FinSet} . Is that a strike against the definition of equivalence, showing it to be too weak?

It might help to think of a toy example. Consider the two categories which we can diagram respectively as

$$\bullet \curvearrowright \qquad \curvearrowleft \bullet \rightleftarrows \star \curvearrowright$$

On the left, we have the category $\mathbf{1}$; on the right we have a two-object category $\mathbf{2}!$ with arrows in *both* directions between the objects (in addition, of course, to the required identity arrows). These two categories are also equivalent. For the obvious inclusion functor $\mathbf{1} \hookrightarrow \mathbf{2}!$ is full and faithful, and it is trivially essentially surjective on objects as each object in the two-object category is isomorphic to the other.

What this toy example highlights is that our equivalence criterion counts categories as amounting to the same when (putting it very roughly) one is just the same as the other padded out with new objects and just enough arrows to make the new objects isomorphic to some old objects.

But on reflection that's fine. Taking a little bit of the mathematical world and bulking it out with copies of the structures it already contains and isomorphisms between the copies won't, for many (most? nearly all?) purposes, give us a real enrichment. Therefore a criterion of equivalence of categories-as-mathematical-universes that doesn't care about surplus isomorphic copies is what we typically need. Hence the results that $\mathbf{1} \simeq \mathbf{2}!$ and $\mathbf{FinOrd} \simeq \mathbf{FinSet}$ are arguably welcome features, not bugs, of our account of equivalence.

23.4 Skeletons and evil

(a) Even categories are regarded as being equivalent in an important sense even if one is bulked out with isomorphic extras, shouldn't the usual sort of concern for Bauhaus elegance and lack of redundancy lead us to privilege categories which are as skeletal as possible? Let's say:

Definition 110. The category \mathcal{S} is a *skeleton* of the category \mathcal{C} if \mathcal{S} is a full subcategory of \mathcal{C} which contains exactly one object from each class of isomorphic objects of \mathcal{C} . A category is *skeletal* if it is a skeleton of some category.

For a toy example, suppose \mathcal{C} is a category arising from a pre-order – as in §3.3 (C4). Then any skeleton of \mathcal{C} will be a poset category. (Check that!)

Theorem 116. *If \mathcal{S} is a skeleton of the category \mathcal{C} then $\mathcal{S} \simeq \mathcal{C}$.*

Proof. The inclusion functor $\mathcal{S} \hookrightarrow \mathcal{C}$ is fully faithful, and by the definition of \mathcal{S} is essentially surjective on objects. So we can apply Theorem 113. \square

So how about this for a programme? Take the usual universe of categories. But now slim it down by taking skeletons. Then work with these. And we can now forget bloated non-skeletal categories (and forget too about the notion of equivalence and revert to using the simpler notion of isomorphism, because equivalent skeletal categories are in fact isomorphic). What's not to like?

The trouble is that hardly any categories that occur in the wild (so to speak) are skeletal. And slimming down has to be done by appeal to an axiom of choice. Indeed the following statements are each equivalent to a version of the axiom of choice:

- (1) Any category has a skeleton.
- (2) A category is equivalent to any of its skeletons
- (3) Any two skeletons of a given category are isomorphic.

The choice of a skeleton is usually quite artificial – there typically won't be a canonical choice. So any gain in simplicity from concentrating on skeletal categories would be bought at the cost of having to adopt 'unnatural', non-canonical, choices of skeletons. Given that category theory is supposed to be all about natural patterns already occurring in mathematics, this perhaps isn't going to be such a good trade-off after all.

(b) Categorical notions that are not invariant under equivalence are sometimes said to be 'evil'. So being skeletal is evil. So too is being small:

Theorem 117. *Smallness is not preserved by categorial equivalence.*

In other words, we can have \mathcal{C} a small category, $\mathcal{C} \simeq \mathcal{D}$, yet \mathcal{D} not small. This is a simple corollary of our observation in §23.3 that if we take a category, inflate it by adding lots of objects and just enough arrows to ensure that these objects are isomorphic to the original objects, then the augmented category is equivalent to the one we started with. For an extreme example, start with the one-object category $\mathbf{1}$, i.e. $\bullet \curvearrowright$ (that's small)! Now add as new objects every set, and as new arrows an identity arrow for each set, and also for every set X a pair of arrows $\bullet \rightleftarrows X$ which composed to give identities. Then we get a new pumped-up category $\mathbf{1}^+$ (which is certainly not small). But $\mathbf{1}^+ \simeq \mathbf{1}$.

If you fuss about evil, you can highlight a neighbouring notion to smallness which evidently is virtuous:

Definition 111. A category is *essentially small* if it is equivalent to category with a set's worth of arrows.

But we aren't going to fuss here.

There is, by the way, a companion positive result

Equivalent categories

Theorem 118. *Local smallness is preserved by categorical equivalence.*

Proof. An equivalence $\mathcal{C} \xrightleftharpoons[G]{F} \mathcal{D}$ requires F and G to be full and faithful functors. So in particular, for any \mathcal{D} -objects D, D' , there are the same number of arrows between them as between the \mathcal{C} -objects GD, GD' . So that ensures that if \mathcal{C} has only a set's worth of arrows between any pair of objects, the same goes for \mathcal{D} . \square

24 The Yoneda embedding

We met hom-functors in Chapter 18: they have nice properties like preserving limits. We introduced natural transformations in Chapter 21. We now put things together and start talking about natural transformations between hom-functors.

This will quickly lead on to a proof of a preliminary, restricted, version of the important Yoneda Lemma, and we discover the related Yoneda embedding. These tell us how to find a category built from functors-into-Set-and-arrows-between-them which looks just like the category we start off with. This seems closely analogous to some classical representation theorems like e.g. Cayley's Theorem which tells us how, starting from any group, we can find a group built specifically from permutations-of-a-set which looks just the given group. So we will say something about the parallel.

24.1 Natural transformations between hom-functors

(a) Take a locally small category \mathcal{C} : in fact, in this chapter, we assume all the relevant categories are locally small, so that we can unproblematically talk about the relevant hom-sets and hom-functors. Fix on a \mathcal{C} -arrow $f: B \rightarrow A$, noting the direction of the arrow here. And we now describe how to construct from f a corresponding natural transformation α from the hom-functor $\mathcal{C}(A, -)$ to the hom-functor $\mathcal{C}(B, -)$.

By definition, if α is to be a natural transformation, its components must be such that the following diagram commutes, given any arrow $j: X \rightarrow Y$:

$$\begin{array}{ccc} \mathcal{C}(A, X) & \xrightarrow{\mathcal{C}(A, j)} & \mathcal{C}(A, Y) \\ \downarrow \alpha_X & & \downarrow \alpha_Y \\ \mathcal{C}(B, X) & \xrightarrow{\mathcal{C}(B, j)} & \mathcal{C}(B, Y) \end{array}$$

where $\mathcal{C}(C, j)$, you will recall, is the map $j \circ -$ which sends an arrow $h: C \rightarrow X$ to the arrow $j \circ h: C \rightarrow Y$.

Suppose then that we set a component $\alpha_Z: \mathcal{C}(A, Z) \rightarrow \mathcal{C}(B, Z)$ to be the function $- \circ f$ that sends an arrow $k: A \rightarrow Z$ to the composite $k \circ f: B \rightarrow Z$ (the only obvious way to use f).

The Yoneda embedding

Then our diagram will indeed commute. For going round the top-route takes us from $g: A \rightarrow X$ to $j \circ g: A \rightarrow Y$ to $(j \circ g) \circ f: B \rightarrow Y$; and going round the bottom route takes us from $g: A \rightarrow X$ to $g \circ f: A \rightarrow Y$ to $j \circ (g \circ f): B \rightarrow Y$.

So in sum, if there is a morphism $f: B \rightarrow A$, then there is a corresponding natural transformation $\alpha: \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$ with components α_Z as defined.

And note too: if f is an isomorphism, then each component α_Z (i.e. $- \circ f$) has an inverse (i.e. $- \circ f^{-1}$), so is an isomorphism. Therefore the induced transformation α is a natural isomorphism.

To sum up this result and introduce some notation:

Theorem 119. *Suppose \mathcal{C} is a locally small category, and $\mathcal{C}(A, -)$, $\mathcal{C}(B, -)$ are hom-functors (for objects A, B in \mathcal{C}). Then, given an arrow $f: B \rightarrow A$, there exists a corresponding natural transformation $\mathcal{C}(f, -): \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$, where for each Z , the component $\mathcal{C}(f, -)_Z: \mathcal{C}(A, Z) \rightarrow \mathcal{C}(B, Z)$ sends an arrow $k: A \rightarrow Z$ to $k \circ f: B \rightarrow Z$.*

Furthermore, if f is an isomorphism, then $\mathcal{C}(f, -)$ is a natural isomorphism.

(b) Both as a quick reality-check and for future use, let's pause to show:

Theorem 120. *Given a locally small category \mathcal{C} including objects A, B, C , and arrows $f: B \rightarrow A$ and $g: C \rightarrow B$, then*

- (1) $\mathcal{C}(f \circ g, -) = \mathcal{C}(g, -) \circ \mathcal{C}(f, -)$.
- (2) $\mathcal{C}(f, -)_A 1_A = f$.
- (3) $\mathcal{C}(1_A, -) = 1_{\mathcal{C}(A, -)}$.

Proof. (1) $\mathcal{C}(f \circ g, -)_Z$ sends any arrow $e: A \rightarrow Z$ to $e \circ (f \circ g)$. However, $(\mathcal{C}(f, -)_Z(e) = e \circ f$, so $\mathcal{C}(g, -)_Z(\mathcal{C}(f, -)_Z(e)) = (e \circ f) \circ g$. Which means that $\mathcal{C}(f \circ g, -)$ and $\mathcal{C}(g, -) \circ \mathcal{C}(f, -)$ agree on all components, so are identical natural transformations.

(2) $\mathcal{C}(f, -)_A$ sends any arrow $j: A \rightarrow A$ to $j \circ f: B \rightarrow A$. So in particular it sends 1_A to f .

(3) $\mathcal{C}(1_A, -)_Z$ sends any arrow $j: A \rightarrow Z$ to itself. While $1_{\mathcal{C}(A, -)}$ is the identity arrow on the object $\mathcal{C}(A, -)$ in the functor category $[\mathcal{C}, \text{Set}]$. In other words it is natural transformation from $\mathcal{C}(A, -)$ to itself which in particular sends $j: A \rightarrow Z$ to itself. Which shows that $\mathcal{C}(1_A, -)$ and $1_{\mathcal{C}(A, -)}$ agree on all components so are identical. \square

(c) The obvious next question to ask is: are *all* possible natural transformations between the hom-functors $\mathcal{C}(A, -)$ and $\mathcal{C}(B, -)$ generated from arrows $f: B \rightarrow A$ in the way described in Theorem 119?

Start from a natural transformation $\alpha: \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$. If α is indeed of the form $\mathcal{C}(f, -)$ for some $f: B \rightarrow A$, then by the last theorem $\alpha_A 1_A = \mathcal{C}(f, -)_A 1_A = f$. So we already know one candidate for f , and we might naturally conjecture:

24.1 Natural transformations between hom-functors

Theorem 121. *Suppose \mathcal{C} is a locally small category, and consider the hom-functors $\mathcal{C}(A, -)$ and $\mathcal{C}(B, -)$, for objects A, B in \mathcal{C} . Then if there is a natural transformation $\alpha: \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$, there is a unique arrow $f: B \rightarrow A$ such that $\alpha = \mathcal{C}(f, -)$, namely $f = \alpha_A(1_A)$.*

And this indeed is right:

Proof. Since α is a natural transformation, the following diagram in particular must commute, for any Z and any $g: A \rightarrow Z$,

$$\begin{array}{ccc} \mathcal{C}(A, A) & \xrightarrow{\mathcal{C}(A, g)} & \mathcal{C}(A, Z) \\ \downarrow \alpha_A & & \downarrow \alpha_Z \\ \mathcal{C}(B, A) & \xrightarrow{\mathcal{C}(B, g)} & \mathcal{C}(B, Z) \end{array}$$

We start with $\mathcal{C}(A, A)$ at the top left because we know that it is populated, at least by 1_A . Then, recalling the definitions, $\mathcal{C}(A, g)$ is the map that (among other things) sends an arrow $h: A \rightarrow A$ to the arrow $g \circ h: A \rightarrow Z$, and $\mathcal{C}(B, g)$ sends an arrow $k: B \rightarrow A$ to the arrow $g \circ k: B \rightarrow Z$.

Chase that identity arrow 1_A round the diagram from the top left to bottom right nodes. The top route sends it to $\alpha_Z(g)$. The bottom route sends it to $g \circ (\alpha_A(1_A))$, which equals $\mathcal{C}(\alpha_A(1_A), -)_Z(g)$ (check how we set up the notation in Theorem 119). Since our square always commutes we have

$$\text{for all objects } Z \text{ and arrows } g: A \rightarrow Z, \quad \alpha_Z(g) = \mathcal{C}(\alpha_A(1_A), -)_Z(g).$$

Hence, since Z and g were arbitrary,

$$\alpha = \mathcal{C}(\alpha_A(1_A), -).$$

Putting $f: B \rightarrow A =_{\text{def}} \alpha_A(1_A)$ therefore proves the existence part of the theorem.

Now suppose both f and f' are such that $\alpha = \mathcal{C}(f, -) = \mathcal{C}(f', -)$. Then by Theorem 120 (2)

$$f = \mathcal{C}(f, -)_A(1_A) = \mathcal{C}(f', -)_A(1_A) = f'$$

which shows f 's uniqueness. □

(d) The theorems so far in this section have been about covariant hom-functors. We have corresponding duals for contravariant hom-functors. Here's part of the story (proofs are routine exercises in dualization, paying attention to the direction of arrows):

Theorem 122. *Suppose \mathcal{C} is a locally small category, and $\mathcal{C}(-, A)$, $\mathcal{C}(-, B)$ are contravariant hom-functors (for objects A, B in \mathcal{C}). Then (1) if there exists an*

The Yoneda embedding

arrow $f: A \rightarrow B$, there is a natural transformation $\mathcal{C}(-, f): \mathcal{C}(-, A) \Rightarrow \mathcal{C}(-, B)$, where for each Z , the component $\mathcal{C}(-, f)_Z: \mathcal{C}(Z, A) \rightarrow \mathcal{C}(Z, B)$ sends an arrow $k: Z \rightarrow A$ to $f \circ k: Z \rightarrow B$.

And (2) if there is a natural transformation $\alpha: \mathcal{C}(-, A) \Rightarrow \mathcal{C}(-, B)$, there is a unique arrow $f: A \rightarrow B$ such that $\alpha = \mathcal{C}(-, f)$, namely $f = \alpha_A(1_A)$.

(3) $\mathcal{C}(-, g \circ f) = \mathcal{C}(-, g) \circ \mathcal{C}(-, f)$.

24.2 The Restricted Yoneda Lemma

Sticking to the covariant case for the moment, we have been considering pairs of hom-functors such as $\mathcal{C}(A, -): \mathcal{C} \rightarrow \mathbf{Set}$ and $\mathcal{C}(B, -): \mathcal{C} \rightarrow \mathbf{Set}$, and the natural transformations between them. Theorem 121 tells us that there are no more such natural transformations than there are \mathcal{C} -arrows $f: B \rightarrow A$. Since we are assuming all along that \mathcal{C} is locally small, that means there can be a set of such natural transformations. It is a hom-set for the functor category $[\mathcal{C}, \mathbf{Set}]$; in the notation of Defn. 103, we can denote it ‘ $\mathit{Nat}(\mathcal{C}(A, -), \mathcal{C}(B, -))$ ’.

Now, a \mathcal{C} -arrow $f: B \rightarrow A$ is of course a member of the hom-set $\mathcal{C}(B, A)$. So, in the proofs of our Theorems 119 and 121 we have in effect defined two suites of functions \mathcal{X}_{AB} and \mathcal{E}_{AB} in \mathbf{Set} (functions indexed by the \mathcal{C} -objects A, B), where

- i) $\mathcal{X}_{AB}: \mathcal{C}(B, A) \rightarrow \mathit{Nat}(\mathcal{C}(A, -), \mathcal{C}(B, -))$ sends a function $f: B \rightarrow A$ to the natural transformation $\mathcal{C}(f, -)$.
- ii) $\mathcal{E}_{AB}: \mathit{Nat}(\mathcal{C}(A, -), \mathcal{C}(B, -)) \rightarrow \mathcal{C}(B, A)$ sends a natural transformation $\alpha: \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$ to $\alpha_A(1_A)$.

And again, the next thing to do is obvious: we check that \mathcal{X}_{AB} and \mathcal{E}_{AB} are inverses of each other in \mathbf{Set} as they surely ought to be.

Let’s fix on some particular A and B . Then we note:

- (1) Given some $f: B \rightarrow A$,

$$(\mathcal{E}_{AB} \circ \mathcal{X}_{AB})f = \mathcal{E}_{AB}(\mathcal{C}(f, -)) = \mathcal{C}(f, -)_A(1_A) = f$$

with the last identity by Theorem 120 (2). But f was arbitrary. Whence $\mathcal{E}_{AB} \circ \mathcal{X}_{AB} = 1$.

- (2) Given some $\alpha: \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$,

$$(\mathcal{X}_{AB} \circ \mathcal{E}_{AB})\alpha = \mathcal{X}_{AB}(\alpha_A(1_A)) = \mathcal{C}(\alpha_A(1_A), -) = \alpha$$

where the last identity is as shown in the proof of Theorem 121. But α was arbitrary. Whence $\mathcal{X}_{AB} \circ \mathcal{E}_{AB} = 1$.

So \mathcal{X}_{AB} and \mathcal{E}_{AB} are mutual inverses, and hence isomorphisms. Therefore we have in summary:

Theorem 123 (Restricted Yoneda Lemma). *Suppose \mathcal{C} is a locally small category, and A, B are objects of \mathcal{C} . Then $\text{Nat}(\mathcal{C}(A, -), \mathcal{C}(B, -)) \cong \mathcal{C}(B, A)$.*

There is, needless to say, a dual version of all this. For each A, B in \mathcal{C} , there is an isomorphism $\mathcal{Y}_{AB}: \mathcal{C}(A, B) \rightarrow \text{Nat}(\mathcal{C}(-, A), \mathcal{C}(-, B))$ which sends a function $f: A \rightarrow B$ to the natural transformation $\mathcal{C}(-, f)$; and \mathcal{Y}_{AB} has an inverse. Consequently,

Theorem 124 (Restricted Yoneda Lemma, continued). *Suppose \mathcal{C} is a locally small category, and A, B are objects of \mathcal{C} . Then $\text{Nat}(\mathcal{C}(-, A), \mathcal{C}(-, B)) \cong \mathcal{C}(A, B)$.*

The shared label we've given this dual pair of theorems is not standard, but the reason for it will become clear when we meet the full Yoneda Lemma in Ch. 25.

The future full version has a reputation for being the first result in category theory whose proof takes some real effort to understand. Be that as it may, at least the route up to our current cut-down version should seem entirely unproblematic. A simple observation established Theorem 119, that each $f: B \rightarrow A$ generates a natural transformation from $\mathcal{C}(A, -)$ to $\mathcal{C}(B, -)$. It was then very natural to ask if there is a converse result, and we get Theorem 121. In proving those simple theorems, we have set up maps each way between members of $\mathcal{C}(B, A)$ and of $\text{Nat}(\mathcal{C}(A, -), \mathcal{C}(B, -))$. Checking that those maps are indeed mutually inverse as we might expect gives us the Restricted Yoneda Lemma – which is all we need for the main result in this chapter, and for a number of other results which are often said to obtain ‘by Yoneda’.

24.3 The Yoneda embedding

(a) Suppose, as always in this chapter, that the category \mathcal{C} is locally small, then:

- (i) we can define a map – let's call it \mathcal{X}_{ob} – that takes any \mathcal{C} -object A (equivalently, any \mathcal{C}^{op} -object A) and sends it to the corresponding hom-functor $\mathcal{C}(A, -)$.
- (ii) we can define another map – let's call it \mathcal{X}_{arw} – that takes any \mathcal{C} -arrow $f: B \rightarrow A$ (equivalently, any \mathcal{C}^{op} -arrow $f: A \rightarrow B$) and sends it to $\mathcal{X}_{AB}f$, i.e. sends f to the natural transformation $\mathcal{C}(f, -): \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$.

Now, hom-functors like $\mathcal{C}(A, -)$ are objects of the functor category $[\mathcal{C}, \text{Set}]$. And natural transformations like $\mathcal{C}(f, -): \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$ are arrows in that same category. So, we might hope that, as our labels for them prematurely suggest, the maps \mathcal{X}_{ob} and \mathcal{X}_{arw} can be put together as the components of a covariant functor $\mathcal{X}: \mathcal{C}^{op} \rightarrow [\mathcal{C}, \text{Set}]$.

To confirm that they can be, we just need to check the two functorial axioms are indeed satisfied. First, identities are preserved:

$$\mathcal{X}(1_A) = \mathcal{C}(1_A, -) = 1_{\mathcal{C}(A, -)} = 1_{\mathcal{X}(A)}$$

The Yoneda embedding

where the central equation holds by Theorem 120 (3). And secondly, composition is respected. In other words, for any composable f, g in \mathcal{C}^{op} ,

$$\mathcal{X}(g \circ^{\mathcal{C}^{op}} f) = \mathcal{X}(f \circ^{\mathcal{C}} g) = \mathcal{C}(f \circ^{\mathcal{C}} g, -) = \mathcal{C}(g, -) \circ^{\square} \mathcal{C}(f, -) = \mathcal{X}(g) \circ^{\square} \mathcal{X}(f)$$

where ‘ \circ^{\square} ’ indicates composition in the functor category $[\mathcal{C}, \mathbf{Set}]$, and the third equation holds by Theorem 120 (1).

Let’s summarize this important result, again along with its obvious dual companion where we similarly define a functor \mathcal{Y} in terms of the maps \mathcal{Y}_{AB} :

Theorem 125. *For any locally small category \mathcal{C} , there is a functor we’ll label simply $\mathcal{X}: \mathcal{C}^{op} \rightarrow [\mathcal{C}, \mathbf{Set}]$ with components \mathcal{X}_{ob} and \mathcal{X}_{arw} such that*

- (1) for any $A \in ob(\mathcal{C}^{op})$, $\mathcal{X}_{ob}(A) = \mathcal{C}(A, -)$,
- (2) for any arrow $f \in \mathcal{C}^{op}(A, B)$, i.e. arrow $f: B \rightarrow A$ in \mathcal{C} , $\mathcal{X}_{arw}(f) = \mathcal{C}(f, -)$.

And there is similarly a functor $\mathcal{Y}: \mathcal{C} \rightarrow [\mathcal{C}^{op}, \mathbf{Set}]$ with components \mathcal{Y}_{ob} and \mathcal{Y}_{arw} such that

- (3) for any $A \in ob(\mathcal{C})$, $\mathcal{Y}_{ob}(A) = \mathcal{C}(-, A)$.
- (4) For any arrow $f: A \rightarrow B$ in \mathcal{C} , $\mathcal{Y}_{arw}(f) = \mathcal{C}(-, f)$.

(b) It is immediate that the functors \mathcal{X} and \mathcal{Y} behave nicely in various ways. In particular:

Theorem 126. *$\mathcal{X}: \mathcal{C}^{op} \rightarrow [\mathcal{C}, \mathbf{Set}]$ and $\mathcal{Y}: \mathcal{C} \rightarrow [\mathcal{C}^{op}, \mathbf{Set}]$ are fully faithful functors which are injective on objects.*

Proof. By definition, $\mathcal{X}: \mathcal{C}^{op} \rightarrow [\mathcal{C}, \mathbf{Set}]$ is full just in case, for any \mathcal{C}^{op} -objects A, A' , and any natural transformation $\alpha: \mathcal{C}(A, -) \rightarrow \mathcal{C}(A', -)$ there is an arrow $f: A \rightarrow A'$ in \mathcal{C}^{op} , i.e. an arrow $f: A' \rightarrow A$ in \mathcal{C} , such that $\alpha = \mathcal{X}f = \mathcal{C}(f, -)$. Which we have already proved as the existence claim in Theorem 121.

By definition, $\mathcal{X}: \mathcal{C}^{op} \rightarrow [\mathcal{C}, \mathbf{Set}]$ is faithful just in case, for any \mathcal{C}^{op} -objects A, A' , and any pair of arrows $f, g: A \rightarrow A'$ in \mathcal{C}^{op} , i.e. any pair of arrows $f, g: A' \rightarrow A$ in \mathcal{C} , then if $\mathcal{C}(f, -) = \mathcal{C}(g, -)$ then $f = g$. But that follows immediately from the uniqueness claim in Theorem 121.

So the only new claim is that \mathcal{X} is injective on objects, meaning that if $A \neq B$, then $\mathcal{X}(A) \neq \mathcal{X}(B)$. Suppose $\mathcal{X}(A) = \mathcal{X}(B)$, i.e. $\mathcal{C}(A, -) = \mathcal{C}(B, -)$. Then $\mathcal{C}(A, -)(C) = \mathcal{C}(B, -)(C)$, i.e. $\mathcal{C}(A, C) = \mathcal{C}(B, C)$. But that can’t be so if $A \neq B$, since by our lights hom-sets on different pairs of objects must be disjoint (see the last sentence of §16.6).

The proof for $\mathcal{Y}: \mathcal{C} \rightarrow [\mathcal{C}^{op}, \mathbf{Set}]$ is straightforwardly dual. □

As an important corollary, we now have

Theorem 127. *For any objects A, B in the locally small category \mathcal{C} , $A \cong B$ iff $\mathcal{X}A \cong \mathcal{X}B$, and likewise $A \cong B$ iff $\mathcal{Y}A \cong \mathcal{Y}B$.*

Proof. Suppose $A \cong B$. Then there is an isomorphism $f: B \xrightarrow{\sim} A$. So there is a natural transformation $\mathcal{C}(f, -): \mathcal{C}(A, -) \Rightarrow \mathcal{C}(B, -)$, which by Theorem 119 is an isomorphism. So in our alternative notation, $\mathcal{X}f: \mathcal{X}A \xrightarrow{\cong} \mathcal{X}B$. Hence $\mathcal{X}A \cong \mathcal{X}B$.

Now suppose $\mathcal{X}A \cong \mathcal{X}B$. So there exists a natural isomorphism $\alpha: \mathcal{C}(A, -) \xrightarrow{\cong} \mathcal{C}(B, -)$. By Theorem 121, α is $\mathcal{C}(f, -)$ for some $f: B \rightarrow A$, i.e. is $\mathcal{X}f$. But \mathcal{X} is fully faithful. So Theorem 80 tells us that since $\mathcal{X}f$ is an isomorphism, so is f . Hence $A \cong B$.

That shows $A \cong B$ iff $\mathcal{X}A \cong \mathcal{X}B$. The argument for the functor \mathcal{Y} is dual. \square

(c) So the situation is this. The functor \mathcal{Y} , for example, injects a copy of the \mathcal{C} -objects one-to-one into the objects of the functor category $[\mathcal{C}^{op}, \mathbf{Set}]$; and then it fully and faithfully matches up the arrows between \mathcal{C} -objects with arrows between the corresponding objects in $[\mathcal{C}^{op}, \mathbf{Set}]$. In other words, \mathcal{Y} yields an isomorphic copy of \mathcal{C} sitting inside the functor category as a full sub-category.

So, in a phrase, \mathcal{Y} embeds a copy of \mathcal{C} in $[\mathcal{C}^{op}, \mathbf{Set}]$. Hence the terminology (in honour of its discoverer):

Definition 112. The full and faithful functor $\mathcal{Y}: \mathcal{C} \rightarrow [\mathcal{C}^{op}, \mathbf{Set}]$ is the *Yoneda embedding* of \mathcal{C} . \triangle

There was a reason, then, behind our use of ‘ \mathcal{Y} ’ for this functor! And indeed the ‘ \mathcal{Y} ’ notation – in upper or lower case, in one font or another – is pretty standard for the Yoneda embedding. However, ‘ \mathcal{X} ’ is just our label for the dual embedding, which doesn’t seem to have a standard name or notation, though we can usefully call it a Yoneda embedding too.

24.4 Yoneda meets Cayley

(a) Take any locally small category you like. Then the Yoneda embedding tells us how to find a category built from functors-into-Set-and-arrows-between-them which looks just like the category we started off with. Now, as we remarked in the preamble at the beginning of this chapter, this is surely reminiscent of some classical representation theorems which tell us how, given a mathematical structure of a certain type, we can find another structure which lives in the universe of sets and is isomorphic to it. At the simple end of the spectrum there is an observation that we can attribute to Dedekind: any given partially ordered objects are isomorphic to certain corresponding sets ordered by set-inclusion. A significantly more sophisticated result of the same flavour is the Stone Representation Theorem: any Boolean algebra is isomorphic to a field of sets (where a field of sets is a sub-algebra of a canonical power-set algebra $(\mathcal{P}(X), \overline{}, \cap, \cup, \emptyset, X)$,

The Yoneda embedding

where X is some set and of course \bar{A} is $X - A$). Here, though, we'll concentrate on just one such classical representation theorem, namely Cayley's Theorem:

Theorem 128. *Any group (G, \cdot, e) is isomorphic to a subgroup of the group $Sym(G)$, i.e. the group of permutations on the set G .*

Proof. (The usual one, rehearsed here in case you haven't seen it before, and to fix notation). Given any object $g \in G$, we define the set-function $\underline{g}: G \rightarrow G$ by setting $\underline{g}(x) = g \cdot x$ (i.e. $\underline{g} = \{(x, y) \mid x, y \in G \wedge y = g \cdot x\}$).

Evidently any such \underline{g} is surjective: for any $x \in G$, there's an object which \underline{g} sends to x , namely $g^{-1} \cdot x$. And if $\underline{g}(x) = \underline{g}(y)$, then $g \cdot x = g \cdot y$ whence $g^{-1} \cdot g \cdot x = g^{-1} \cdot g \cdot y$, therefore $x = y$. Hence \underline{g} is also injective and is therefore a bijection on G , i.e. is a permutation of the group objects.

Put $K = \{\underline{g} \mid g \in G\}$. It is now routine to confirm $(K, \circ, \underline{e})$ is a group, and hence a subgroup of $Sym(G)$, where the group operation is composition of functions:

- i. Any two functions $\underline{f}, \underline{g}$ have a product $\underline{f} \circ \underline{g}$, where $(\underline{f} \circ \underline{g})(x) = f \cdot g \cdot x$.
- ii. The function \underline{e} is a group identity.
- iii. $\underline{f} \circ (\underline{g} \circ \underline{h}) = (\underline{f} \circ \underline{g}) \circ \underline{h}$ because $f \cdot (g \cdot h) = (f \cdot g) \cdot h$.
- iv. We note that $(\underline{g^{-1}} \circ \underline{g})(x) = \underline{g^{-1}}(g \cdot x) = g^{-1} \cdot g \cdot x = x = \underline{e}(x)$. So $\underline{g^{-1}} \circ \underline{g} = \underline{e}$, and similarly $\underline{g} \circ \underline{g^{-1}} = \underline{e}$. So each \underline{g} has an inverse.

It remains to check that the map F defined by $g \mapsto \underline{g}$ is a group isomorphism from $(G, \cdot, e) \rightarrow (K, \circ, \underline{e})$. F is injective. For if $\underline{f} = \underline{g}$, then $\underline{f}(e) = \underline{g}(e)$, so $f \cdot e = g \cdot e$, so $f = g$. Since F is also a surjection just by the definition of K , F (as a map on the carrier sets) is an isomorphism.

Also, for any x , $F(f \cdot g)(x) = \underline{(f \cdot g)}(x) = f \cdot g \cdot x = \underline{f}(g \cdot x) = \underline{f}(\underline{g}(x)) = (\underline{f} \circ \underline{g})(x) = (Ff \circ Fg)(x)$, so F indeed respects group structure. \square

(b) Now, the modern way is – at least officially – to think of a group (G, \cdot, e) as a set-theoretic structure from the outset; so Cayley's theorem might seem just to tell us that, given one set theoretic structure, we can find another isomorphic one. Big deal! However, that rather disguises what's actually going on.

For various reasons – some good, some rather disreputable – it has become absolutely standard in mathematics to trade in a lot of plural talk (referring to many objects at once) for singular talk (referring to a set of those many objects). For example, we've learnt to slide easily e.g. from talk of the natural numbers (plural) to talk of the set \mathbb{N} (singular). So instead of stating the Least Number Principle as e.g. 'Given any natural numbers, one of them will be the least' we say 'Any set S , where $S \subseteq \mathbb{N}$, has a least member'. But note that the singular talk about a set here is not yet doing any real work. And indeed, quite a lot of informal set talk is in fact similarly low-level, non-committal stuff which can however be readily translated away, most naturally into a plural idiom. That applies here, to part of the statement of Cayley's Theorem. Instead of starting 'Any group

$(G, \cdot, e), \dots$ and thinking of this as already referring to a set-theoretic object (e.g. an ordered triple of a set, a set-function and a set-member), we can capture the core of the theorem like this:

Suppose we are given some objects and a group operation on them with a unit for that operation. Then there will always also be some *sets* (in particular, some set-functions) with a group structure on *them* which form a group isomorphic to the one we started with.

Put this way, stripped of one layer of unnecessary set-idiom, we have (in an intuitive sense) a ‘cross-category’ result which says that objects with a group structure on them (whatever objects they are) can always be represented by an isomorphic structure living in the world of sets.

(c) Recall from §5.2 that a group can be considered as a category in its own right, a one-object category all of whose arrows are isomorphisms. If we take a group (G, \cdot, e) then the corresponding category \mathcal{G} has the following data:

- (i) the sole object of \mathcal{G} : choose whatever object you like, and dub it ‘ \star ’.
- (ii) the arrows of \mathcal{G} are the *elements* of the group (G, \cdot, e) .
- (iii) the identity arrow 1_\star of \mathcal{G} is the identity element e of the group G .
- (iv) the composite $g \circ f: \star \rightarrow \star$ of the two arrows $g, f: \star \rightarrow \star$ is just $g \cdot f$.

Moreover, \mathcal{G} is locally small since its sole potential hom-set $\mathcal{G}(\star, \star)$ is none other than G , which we assume is indeed set-sized.

We can therefore apply the Restricted Yoneda Lemma in one version or the other. And there’s only one possible application of each version. Consider then the version which tells us that

$$\text{Nat}(\mathcal{G}(-, \star), \mathcal{G}(-, \star)) \cong \mathcal{G}(\star, \star).$$

So: what are the natural transformations $\alpha: \mathcal{G}(-, \star) \Rightarrow \mathcal{G}(-, \star)$? We can apply Theorem 122: every such α is $\mathcal{G}(-, g)$ for some arrow g in \mathcal{G} .

Now, by definition, $\mathcal{G}(-, g)$ sends an arrow $x: \star \rightarrow \star$ to $g \circ x: \star \rightarrow \star$. But $\mathcal{G}(\star, \star)$ is just G , and arrows are G -elements, so $\mathcal{G}(-, g)$ acts on G by sending an element x to the element $g \cdot x$. Hence $\mathcal{G}(-, g)$ is the function we earlier called \underline{g} . As before, that’s a bijective map on G , i.e. a permutation on G .

Therefore the Restricted Yoneda Lemma tells us that some set of permutations on the set G is in bijection with the members of G .

Moreover, our proof of the Lemma gives us the isomorphism \mathcal{Y} , which sends the arrow $g: \star \rightarrow \star$ to $\mathcal{G}(-, g)$. By Theorem 122,

$$\mathcal{Y}(g \cdot g') = \mathcal{G}(-, g \cdot g') = \mathcal{G}(-, g) \circ \mathcal{G}(-, g') = \mathcal{Y}(g) \circ \mathcal{Y}(g').$$

So if as before we put a group structure on the natural transformations $\mathcal{G}(-, g)$, i.e. the functions \underline{g} , by again defining multiplication as composition, our isomorphism \mathcal{Y} preserves group structure.

The Yoneda embedding

So in short, we can more or less immediately read off from the proof of the Restricted Yoneda Lemma that a group (G, \cdot, e) is isomorphic to a group of permutations on G with composition as the group operation.

Which is why it is often said that the Yoneda Lemma is a generalization of Cayley's Theorem.

25 The Yoneda Lemma

In Chapter 24 we showed that a couple of easy preliminary theorems were enough to establish what we called the Restricted Yoneda Lemma, and also that the Yoneda embedding is indeed an embedding. For many purposes, this is all we need to know about Yoneda. Still, talking about a Restricted Lemma invites an obvious question: what's the full-power *unrestricted* Yoneda Lemma? This chapter explains.

25.1 Towards the full Yoneda Lemma

Let F be the functor $\mathcal{C}(B, -)$. Then one half of the Restricted version of the Yoneda Lemma, Theorem 123, tells us that there is an isomorphism between $\text{Nat}(\mathcal{C}(A, -), F)$ and FA . The other half of the Restricted Lemma is of course the dual, but for the moment we'll let it look after itself.

Now, to get from where we are to the Yoneda Lemma proper we need two steps:

- (1) We look again at the ingredients of the proof of the restricted version and ask 'Where did we essentially depend on the fact that the second functor, now notated simply ' F ', actually was a hom-functor $\mathcal{C}(B, -)$ for some B ?' Close inspection reveals that we didn't. So we in fact have the more general result that for any locally small category \mathcal{C} , *any* functor $F: \mathcal{C} \rightarrow \mathbf{Set}$, and any \mathcal{C} -object A , there is an isomorphism \mathcal{E} between $\text{Nat}(\mathcal{C}(A, -), F)$ and FA .
- (2) Next we note that our proof of this generalization (like the proof of the original Restricted Lemma) provides a *general recipe* for constructing the required isomorphism. Take a locally small category \mathcal{C} and any \mathcal{C} -object A , then, without having to invoke any arbitrary choices, our proof fixes inverse isomorphisms \mathcal{X}_{AF} and \mathcal{E}_{AF} between $\text{Nat}(\mathcal{C}(A, -), F)$ and FA . In an intuitive sense, we've constructed a *natural* isomorphism. And so we should be able to show that there is a *natural isomorphism* in the official, categorical, sense between some relevant functors.

In sum, we will get from the Restricted Yoneda Lemma to the full-dress Yoneda Lemma by generalizing a construction, and then recasting in category-theoretic

terms an intuitive judgement of the naturality of our construction. Neither step involves anything conceptually very difficult: we just need to nail down all the details. (Some of these proof details are fiddly. By all means skim over them on a first reading, since they are just a matter of checking that the announced steps do go through.)

25.2 The generalizing move

We continue working in a locally small category \mathcal{C} . Let's restate some of what we already know, still using ' F ' to abbreviate ' $\mathcal{C}(B, -)$ ':

- (i) There is a bijection between arrows in FA and natural transformations $\mathcal{C}(A, -) \Rightarrow F$, which sends f in FA to the transformation whose Z -component maps an arrow $g: A \rightarrow Z$ to $g \circ f: B \rightarrow Z$.
- (ii) By definition, the functor F maps an arrow $g: A \rightarrow Z$ to a function Fg which sends an arrow $f: B \rightarrow A$ to the arrow $g \circ f: B \rightarrow Z$. In other words, $Fg(f) = g \circ f$.
- (iii) Hence, putting (i) and (ii) together, we have: there's a bijection which sends an element f in FA to the natural transformation whose Z -component maps $g: A \rightarrow Z$ to $Fg(f)$.

We next want to redeploy this last idea to prove the following generalization of the Restricted Lemma (where we now free up the interpretation of F to allow it to be any functor from \mathcal{C} to \mathbf{Set}):

Theorem 129. *For any locally small category \mathcal{C} , object $A \in \mathcal{C}$ and functor $F: \mathcal{C} \rightarrow \mathbf{Set}$, $\text{Nat}(\mathcal{C}(A, -), F) \cong FA$.*

Proof. Following the constructions in the proof leading up to Restricted Lemma, Theorem 123, first we generalize on \mathcal{X}_{AB} and we'll introduce a map we'll call \mathcal{X}_{AF} :

- (1) \mathcal{X}_{AF} sends f in FA to a natural transformation $\chi = \mathcal{X}_{AF}f: \mathcal{C}(A, -) \Rightarrow F$. We define χ by requiring its Z -component to be the map which takes $g: A \rightarrow Z$ to $Fg(f)$.

We had better pause to check that this definition indeed defines a natural transformation. But that's easy. For χ is a natural transformation if the following square commutes for any $u: Z \rightarrow Z'$:

$$\begin{array}{ccc}
 \mathcal{C}(A, Z) & \xrightarrow{\mathcal{C}(A, u)} & \mathcal{C}(A, Z') \\
 \downarrow \chi_Z & & \downarrow \chi_{Z'} \\
 FZ & \xrightarrow{Fu} & FZ'
 \end{array}$$

The upper route takes some $j: A \rightarrow Z$ to $u \circ j$ to $F(u \circ j)(f)$. The lower route takes j to $Fj(f)$ to $Fu \circ Fj(f)$. The functoriality of F ensures these are equal.

Now, to prove our theorem, we show that \mathcal{X}_{AF} is an isomorphism by providing it with a two-sided inverse. Again, we follow the pattern in the proof of the Restricted Lemma, this time generalizing on \mathcal{E}_{AB} . So we introduce a map we'll call \mathcal{E}_{AF} :

- (2) \mathcal{E}_{AF} sends a natural transformation $\alpha: \mathcal{C}(A, -) \Rightarrow F$ to the element $\alpha_A(1_A)$ in FA .

And now we check that \mathcal{E}_{AF} is indeed a two-sided inverse of \mathcal{X}_{AF} .

First, given an arbitrary element f in FA ,

$$\mathcal{E}_{AF} \circ \mathcal{X}_{AF}(f) = \mathcal{E}_{AF} \circ \chi = \chi_A(1_A) = F1_A(f) = 1_{FA}(f)$$

and therefore $\mathcal{E}_{AF} \circ \mathcal{X}_{AF} = 1$.

Secondly, for $\alpha: \mathcal{C}(A, -) \Rightarrow F$, we have $\mathcal{X}_{AF} \circ \mathcal{E}_{AF}(\alpha) = \mathcal{X}_{AF}(\alpha_A(1_A))$. The Z -component of that sends a map $g: A \rightarrow Z$ to $Fg(\alpha_A(1_A))$. But since α is a natural transformation, this next diagram must commute:

$$\begin{array}{ccc} \mathcal{C}(A, A) & \xrightarrow{\mathcal{C}(A, g)} & \mathcal{C}(A, Z) \\ \downarrow \alpha_A & & \downarrow \alpha_Z \\ FA & \xrightarrow{Fg} & FZ \end{array}$$

So chasing the arrow 1_A round the diagram by each route, we get $Fg(\alpha_A(1_A)) = \alpha_Z(\mathcal{C}(A, g)(1_A)) = \alpha_Z(g)$.

In other words, for any given Z , the Z -component of $\mathcal{X}_{AF} \circ \mathcal{E}_{AF}(\alpha)$ acts on g just like the Z -component of α . Hence $\mathcal{X}_{AF} \circ \mathcal{E}_{AF}(\alpha) = \alpha$ and, since α too was arbitrary, $\mathcal{X}_{AF} \circ \mathcal{E}_{AF} = 1$. (Reality check: what object is that last identity arrow on?). \square

25.3 Making it all natural

One further step takes us to the full Yoneda Lemma. Not only is there an isomorphism \mathcal{E}_{AF} from $\text{Nat}(\mathcal{C}(A, -), F)$ to FA , but \mathcal{E}_{AF} is intuitively 'natural' in the sense of constructed in a uniform way given A and F , without arbitrary choices. We now want to capture this intuitive remark using our official categorical account of a natural isomorphism.

Here's a reminder:

Definition 100 Given functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$, we say that $FA \cong GA$ naturally in A just if F and G are naturally isomorphic.

And what we want to prove first, keeping F fixed, is that $\text{Nat}(\mathcal{C}(A, -), F) \cong FA$ naturally in A . Which, by our definition, means we have to establish that the

The Yoneda Lemma

functor $Nat(\mathcal{C}(\cdot, -), F)$ (using the dot is a place-holder marking where we have abstracted from A) is naturally isomorphic to F . The first functor is in fact just the composite functor

$$\mathcal{C}^{op} \xrightarrow{\mathcal{X}} [\mathcal{C}, \mathbf{Set}] \xrightarrow{Nat(-, F)} \mathbf{Set}$$

where \mathcal{X} is as in Theorem 125, and $Nat(-, F)$ is the sort of contravariant functor we met in Defn. 104. Note, since \mathcal{X} can also be thought of as a contravariant functor from \mathcal{C} and contravariant functors compose to give a covariant functor, we do indeed end up with a covariant functor from \mathcal{C} !

So we want to show the following:

Theorem 130. *Let \mathcal{C} be a locally small category, and F a functor $F: \mathcal{C} \rightarrow \mathbf{Set}$. Then the functors $N = Nat(-, F) \circ \mathcal{X}$ and F are naturally isomorphic.*

Proof. Working through the definition of N

- (i) N sends any \mathcal{C} -object A to the set $Nat(\mathcal{C}(A, -), F)$.
- (ii) N sends any \mathcal{C} -arrow $f: A \rightarrow B$ to an arrow between $Nat(\mathcal{C}(A, -), F)$ and $Nat(\mathcal{C}(B, -), F)$, namely the arrow that sends any $\alpha: \mathcal{C}(A, -) \Rightarrow F$ to the corresponding $\alpha \circ \mathcal{C}(f, -): \mathcal{C}(B, -) \Rightarrow F$.

So now, given any $f: A \rightarrow B$, consider the following diagram,

$$\begin{array}{ccc} Nat(\mathcal{C}(A, -), F) & \xrightarrow{Nf} & Nat(\mathcal{C}(B, -), F) \\ \downarrow \mathcal{E}_{AF} & & \downarrow \mathcal{E}_{BF} \\ FA & \xrightarrow{F(f)} & F(B) \end{array}$$

Take any $\alpha: \mathcal{C}(A, -) \Rightarrow F$. Then we have:

- (1) $\mathcal{E}_{BF} \circ Nf(\alpha) = \mathcal{E}_{BF}(\alpha \circ \mathcal{C}(f, -)) = (\alpha \circ \mathcal{C}(f, -))_B(1_B) = \alpha_B \circ \mathcal{C}(f, -)_B(1_B) = \alpha_B(f)$ (for the last equation, compare the end of the proof of Theorem 121).
- (2) But also $F(f) \circ \mathcal{E}_{AF}(\alpha) = F(f)(\alpha_A(1_A)) = \alpha_B \circ \mathcal{C}(A, f)(1_A) = \alpha_B(f)$ (for the middle equation we note that $F(f) \circ \alpha_A = \alpha_B \circ \mathcal{C}(A, f)$ by a naturality square for α).

So our diagram will always commute, and hence there is a natural isomorphism $\mathcal{E}_F: N \Rightarrow F$ with components $(\mathcal{E}_F)_A = \mathcal{E}_{AF}$ for each $A \in \mathcal{C}$, and our theorem is proved. \square

That captures in categorial terms the intuition that the construction of \mathcal{E}_{AF} depends in a natural way on A ; now for the companion intuition that it depends in a natural way on F too.

Keeping A fixed, we want to prove $Nat(\mathcal{C}(A, -), F) \cong FA$ naturally in F . This means showing the following:

Theorem 131. *Let \mathcal{C} be a locally small category. Then $\text{Nat}(\mathcal{C}(A, -), -)$ and ev_A are naturally isomorphic.*

Here $\text{Nat}(\mathcal{C}(A, -), -)$ is a covariant hom-functor of the kind we met in Defn. 104, and ev_A is the evaluation-at- A functor which sends F to FA and which we met in Defn. 105.

Proof. Given any $\gamma: F \Rightarrow G$, consider the following diagram,

$$\begin{array}{ccc} \text{Nat}(\mathcal{C}(A, -), F) & \xrightarrow{\text{Nat}(\mathcal{C}(A, -), \gamma)} & \text{Nat}(\mathcal{C}(A, -), G) \\ \downarrow \mathcal{E}_{AF} & & \downarrow \mathcal{E}_{AG} \\ ev_A(F) = FA & \xrightarrow{ev_A(\gamma)} & ev_A(G) = GA \end{array}$$

Take any $\alpha: \mathcal{C}(A, -) \Rightarrow F$, and recall that $\text{Nat}(\mathcal{C}(A, -), \gamma)$ sends α to $\gamma \circ \alpha$. Then we have:

- (1) $\mathcal{E}_{AG} \circ \text{Nat}(\mathcal{C}(A, -), \gamma)(\alpha) = \mathcal{E}_{AG}(\gamma \circ \alpha) = (\gamma \circ \alpha)_A(1_A) = \gamma_A(\alpha_A(1_A))$.
- (2) But also $ev_A(\gamma) \circ \mathcal{E}_{AF}(\alpha) = \gamma_A(\alpha_A(1_A))$.

Hence the diagram always commutes. Therefore there is a natural isomorphism $\mathcal{E}_A: K \Rightarrow ev_A$ with components $(\mathcal{E}_A)_F = \mathcal{E}_{AF}$ for each $F \in [\mathcal{C}, \text{Set}]$. So we are done. \square

25.4 Putting everything together

So now combine all the ingredients from the last three theorems ...

Cue drum-roll!

... and we at last have the full-dress result:

Theorem 132 (Yoneda Lemma). *For any locally small category \mathcal{C} , object $A \in \mathcal{C}$, and functor $F: \mathcal{C} \rightarrow \text{Set}$, $\text{Nat}(\mathcal{C}(A, -), F) \cong FA$, both naturally in $A \in \mathcal{C}$ and naturally in $F \in [\mathcal{C}, \text{Set}]$.*

There will evidently be a dual version too (involving contravariant functors in \mathcal{C} , i.e. functors in \mathcal{C}^{op}):

Theorem 133 (Yoneda Lemma). *For any locally small category \mathcal{C} , object $A \in \mathcal{C}$, and functor $F: \mathcal{C}^{op} \rightarrow \text{Set}$, $\text{Nat}(\mathcal{C}(-, A), F) \cong FA$, both naturally in $A \in \mathcal{C}$ and naturally in $F \in [\mathcal{C}^{op}, \text{Set}]$.*

Some authors call only the second version the Yoneda Lemma: we'll use the label for both, talking of the covariant and contravariant versions if we need to mark the distinction.

And having done all this work, we see as an afterword that a further generalization is in principle possible. We've so far been working with locally small

categories, i.e. categories whose classes of arrows between pairs of objects are indeed sets which live in **Set**. Suppose we turn our attention to larger categories whose hom-classes (as we could naturally call them) are some bigger collections which live in a suitably well-behaved category, call it **SET**, which allows bigger collections. Then we can re-run our arguments to show that for a category \mathcal{C} with hom-classes in **SET**, $A \in \mathcal{C}$, and a functor $F: \mathcal{C} \rightarrow \mathbf{SET}$, then the **SET** of natural transformations from $\mathcal{C}(A, -)$ to F is in natural isomorphism with FA .

But we won't delay over this further generalization – indeed, will we have occasion to use it?

25.5 A brief afterword on ‘presheaves’

We pause for a footnote on some jargon that you might well encounter in treatments of the Yoneda Lemma: you ought to know about it, even though we will not adopt it here.

Recall our earlier talk of diagrams. In these terms, a set-valued (covariant) functor $F: \mathcal{C} \rightarrow \mathbf{Set}$ counts as diagram of shape \mathcal{C} in **Set**. Unpredictably, the corresponding term for a set-valued contravariant functor is this:

Definition 113. A contravariant functor from \mathcal{C} to **Set**, i.e. a functor $F: \mathcal{C}^{op} \rightarrow \mathbf{Set}$, is a *presheaf* on \mathcal{C} . △

The terminology ‘presheaf’ comes from an example in topology. But we will have to just take it as an arbitrary, though widely used, label.

Definition 114. The presheaves on \mathcal{C} (as objects) together with the natural transformations between them (as arrows) form *the presheaf category on \mathcal{C}* , denoted $\widehat{\mathcal{C}}$. △

But note, $\widehat{\mathcal{C}}$ is just a relabelling of the functor category we met in §24.3 and called $[\mathcal{C}^{op}, \mathbf{Set}]$. And so the Yoneda embedding \mathcal{Y} we met there is a functor $\mathcal{Y}: \mathcal{C} \rightarrow \widehat{\mathcal{C}}$; and in our new notation we can say that \mathcal{C} is isomorphic to a full subcategory of $\widehat{\mathcal{C}}$.

Recall $\mathcal{Y}A = \mathcal{C}(-, A)$. Hence $\widehat{\mathcal{C}}(\mathcal{Y}A, F)$ is the hom-class of the presheaf category $\widehat{\mathcal{C}}$ which comprises the arrows of that functor category from $\mathcal{C}(-, A)$ to F , i.e. it is $\mathit{Nat}(\mathcal{C}(-, A), F)$. That’s why (one version) of the Yoneda Lemma can also be presented like this: on the usual assumptions, $\widehat{\mathcal{C}}(\mathcal{Y}A, F) \cong FA$, naturally in both $A \in \mathcal{C}$ and F in $\widehat{\mathcal{C}}$.

26 Representables and universal elements

We saw in §18.3 that covariant hom-functors $\mathcal{C}(A, -)$ have the key property of preserving whatever (small) limits exist in \mathcal{C} . We will show in a moment that isomorphic functors preserve the same limits. So we are naturally going to be interested too in the functors which are isomorphic to hom-functors, as they will also preserve limits. These are the *representable* functors.

This chapter, then, discusses representable functors, their so-called representations, and the associated notion of universal elements. The definitions and theorems are easy: but the wider significance of these notions will perhaps only become clear when we discuss them in relation to adjunctions in later chapters.

26.1 Isomorphic functors preserve the same limits

We start with the intuitive thought that naturally isomorphic functors ought to behave in essentially the same way. In particular, we ought to have the following theorem:

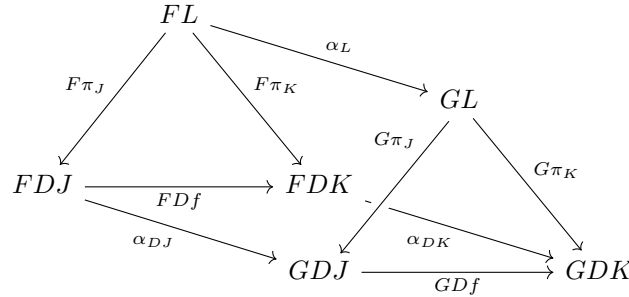
Theorem 134. *Suppose the parallel functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$ are naturally isomorphic. Then if F preserves a given limit so does G .*

We confirm this by a pedestrian apply-the-definitions proof. The argument would look simpler if we could wave our hands at diagrams drawn with different coloured chalks and growing in real time on a blackboard! But in monochrome, we have:

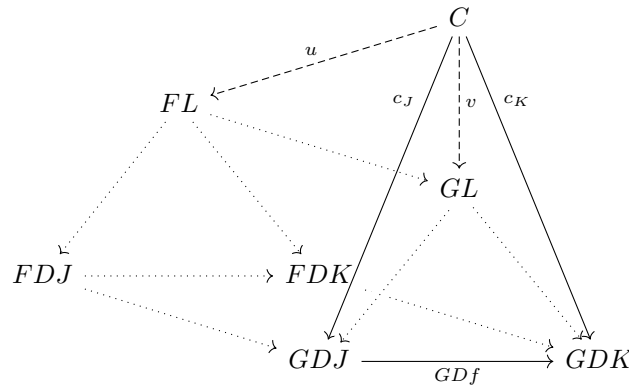
Proof. Let $[L, \pi_J]$ be a limit cone for $D: \mathbf{J} \rightarrow \mathcal{C}$. Then for any $f: J \rightarrow K$ in \mathbf{J} , this diagram commutes in \mathcal{C} :

$$\begin{array}{ccc}
 & L & \\
 \pi_J \swarrow & & \searrow \pi_K \\
 DJ & \xrightarrow{Df} & DK
 \end{array}$$

The actions of F and G now send this triangle to the two commuting triangles in the next diagram, and the assumed natural isomorphism $\alpha: F \xrightarrow{\cong} G$ gives us *three* naturality squares, giving us the sides of a commuting prism in \mathcal{D} :



So now consider any cone $[C, c_J]$ over GD with vertex C . Being part of a cone, each tall triangle such as the one below commutes:



Further, using the commuting base square of the prism, we can extend each leg c_J of the cone by composition with α_{DJ}^{-1} to get a cone $[C, \alpha_{DJ}^{-1} \circ c_J]$ over FD .

Now suppose for the sake of argument that F preserves the limit $[L, \pi_J]$. Then $[FL, F\pi_J]$ must be a limit cone over FD . Which means that our cone $[C, \alpha_{DJ}^{-1} \circ c_J]$ over FD must factor through this limit cone via a unique $u: C \rightarrow FL$.

But it is easy to check – chasing arrows round the diagram, using the sloping sides of the prism – that this implies in turn that $[C, c_J]$ over GD factors through $[GL, G\pi_J]$ via $v = \alpha_L \circ u$.

And $[C, c_J]$ can't factor through a distinct v' : or else there would be a distinct $u' = \alpha_L^{-1} \circ v'$ which makes everything commute, which is impossible by the uniqueness of u .

Hence, in sum, any $[C, c_J]$ factors through $[GL, G\pi_J]$ via a unique v , and therefore $[GL, G\pi_J]$ is a limit cone. So G also preserves the limit $[L, \pi_J]$. \square

26.2 Representable functors

(a) As we remarked at the outset, covariant hom-functors preserve limits. Isomorphisms between functors carry over this property. Similarly contravariant hom-functors preserve colimits as limits of the same shape (see Theorem 102):

and, by duality, isomorphisms between contravariant functors similarly carry over this property.

This makes the following concept an evidently interesting one:

Definition 115. A set-valued functor $F : \mathcal{C} \rightarrow \mathbf{Set}$ which is naturally isomorphic to some hom-functor $\mathcal{C}(A, -) : \mathcal{C} \rightarrow \mathbf{Set}$ is said to be *representable*.

Likewise, a set-valued contravariant functor $F : \mathcal{C} \rightarrow \mathbf{Set}$ which is naturally isomorphic to some hom-functor $\mathcal{C}(-, A) : \mathcal{C} \rightarrow \mathbf{Set}$ is also said to be representable. \triangle

And it is immediate that

Theorem 135. *A covariant representable functor $F : \mathcal{C} \rightarrow \mathbf{Set}$ preserves all (small) limits that exist in \mathcal{C} . Similarly, a contravariant representable functor preserves colimits as limits of the same shape.*

Now, it would perhaps seem most natural to describe the hom-functor that gives us an isomorphic copy of the representable functor $F : \mathcal{C} \rightarrow \mathbf{Set}$ as being a representation of F . But that isn't how the standard jargon goes. Rather:

Definition 116. If there is a natural isomorphism $\psi : \mathcal{C}(A, -) \xrightarrow{\cong} F$, then the object A in \mathcal{C} , is said to be a *representation* of the representable functor F . Similarly for the contravariant case. \triangle

This way of talking does make some claims about representations initially sound slightly odd: we just have to live with that.

Representations need not be strictly unique. However, we do have

Theorem 136. *If the functor $F : \mathcal{C} \rightarrow \mathbf{Set}$ is represented by both A and B , then $A \cong B$.*

Proof. If we have $\mathcal{C}(A, -) \cong F \cong \mathcal{C}(B, -)$ then, in the notation of Theorem 127, $\mathcal{X}A \cong \mathcal{X}B$ and hence $A \cong B$. \square

26.3 A first example

Quite trivially, hom-functors themselves are representables. But are there other kinds of example?

Let's return to the very first functor we met back in §15.2, the forgetful functor $F : \mathbf{Mon} \rightarrow \mathbf{Set}$ which sends any monoid $\mathcal{M} = (M, \cdot, 1_M)$ to its underlying set M , and sends a monoid homomorphism $f : \mathcal{M} \rightarrow \mathcal{M}'$ to the same function thought of as an arrow $f : M \rightarrow M'$ in \mathbf{Set} . And let's ask: is there a representing object, i.e. a monoid \mathcal{R} , such that the hom-functor $\mathbf{Mon}(\mathcal{R}, -)$ is naturally isomorphic to the forgetful F ?

Applying the usual definition, the hom-functor $\mathbf{Mon}(\mathcal{R}, -)$ sends a monoid \mathcal{M} in \mathbf{Mon} to $\mathbf{Mon}(\mathcal{R}, \mathcal{M})$. And it sends a monoid homomorphism $f : \mathcal{M} \rightarrow \mathcal{M}'$ to

Representables and universal elements

the set-function $f \circ -$ which sends an arrow $g: R \rightarrow M$ in $\text{Mon}(\mathcal{R}, \mathcal{M})$ to the arrow $f \circ g: R \rightarrow M'$ in $\text{Mon}(\mathcal{R}, \mathcal{M}')$.

And if this functor $\text{Mon}(\mathcal{R}, -)$ is to be naturally isomorphic with the forgetful functor F , there will have to be an isomorphism ψ with a component at each monoid \mathcal{M} such that, for any $f: \mathcal{M} \rightarrow \mathcal{M}'$ in Mon , the following diagram commutes in Set :

$$\begin{array}{ccc}
 M & \xrightarrow{f} & M' \\
 \downarrow \psi_{\mathcal{M}} & & \downarrow \psi_{\mathcal{M}'} \\
 \text{Mon}(\mathcal{R}, \mathcal{M}) & \xrightarrow{f \circ -} & \text{Mon}(\mathcal{R}, \mathcal{M}')
 \end{array}$$

For this to work, we certainly need to choose a representing monoid \mathcal{R} such that (for any monoid \mathcal{M}) there is a bijection between M and $\text{Mon}(\mathcal{R}, \mathcal{M})$. And presumably, for the needed generality, \mathcal{R} will have to be a monoid without too much distinctive structure. That severely limits the possible options.

First shot: take the simplest such ‘boring’ monoid, the one-element monoid 1 . But a moment’s reflection shows that this can’t work as a candidate for \mathcal{R} (typically M has many members, $\text{Mon}(1, \mathcal{M})$ can have only one, so there won’t be an isomorphism between them).

Second shot: take the next simplest unstructured monoid, the free monoid with a single generator. We can think of this monoid as $\mathcal{N} = (\mathbb{N}, +, 0)$ whose generator is 1 , and whose every element is a sum of 1 s. Now consider a homomorphism from \mathcal{N} to \mathcal{M} . $0 \in \mathbb{N}$ has to be sent to the identity element 1_M in M . And once we also fix that $1 \in \mathbb{N}$ gets sent to some $m \in M$, that determines where every element of \mathbb{N} goes (since every non-zero \mathbb{N} element $1 + 1 + 1 + \dots + 1$ will be sent to a corresponding M -element $m \cdot m \cdot m \cdot \dots \cdot m$).

So consider $\psi_{\mathcal{M}}: M \rightarrow \text{Mon}(\mathcal{N}, \mathcal{M})$ which maps m to the unique homomorphism $\overline{m}: \mathcal{N} \rightarrow \mathcal{M}$ which sends $1 \in \mathbb{N}$ to m . $\psi_{\mathcal{M}}$ is evidently bijective – each homomorphism from \mathcal{N} to \mathcal{M} is some \overline{m} for one and only one m in M . Hence $\psi_{\mathcal{M}}$ is an isomorphism in Set .

And now it is easily seen that our diagram always commutes. Chase an element $m \in M$ round the diagram. The route via the north-east node gives us $m \mapsto fm \mapsto \overline{fm}$, the other route gives us $m \mapsto \overline{m} \mapsto f \circ \overline{m}$. But $f \circ \overline{m} = \overline{fm}$ (consider how each acts e.g. on the number 3).

Since the diagram always commutes, this means in turn that the maps $\psi_{\mathcal{M}}$ assemble into a natural isomorphism $\psi: F \xrightarrow{\cong} \text{Mon}(\mathcal{N}, -)$. Hence, in summary:

Theorem 137. *The forgetful functor $F: \text{Mon} \rightarrow \text{Set}$ is representable, and is represented by \mathcal{N} , the free monoid on one generator.*

Being representable, it follows that the forgetful F preserves limits: but we knew that already.

26.4 More examples of representables

Unsurprisingly, there are analogous representation theorems for other forgetful functors. For instance, although we won't pause over the proofs, we have:

- Theorem 138.** (1) *The forgetful functor $F: \mathbf{Grp} \rightarrow \mathbf{Set}$ is representable, and is represented by \mathcal{Z} , the group of integers under addition.*
- (2) *The forgetful functor $F: \mathbf{Ab} \rightarrow \mathbf{Set}$ is representable, and is also represented by \mathcal{Z} .*
- (3) *The forgetful functor $F: \mathbf{Vect} \rightarrow \mathbf{Set}$ (where \mathbf{Vect} is the category of vector spaces over the reals) is representable, and is represented by \mathcal{R} , the reals treated as a vector-space.*
- (4) *The forgetful functor $F: \mathbf{Top} \rightarrow \mathbf{Set}$ is representable, and is represented by the one-point topological space, call it S_0 .*

To comment on the only last of these, we simply note that a trivial continuous function with domain S_0 into a space S in effect picks out a single point of S , so the set of arrows $\mathbf{Top}(S_0, S)$ is indeed in bijective correspondence with the set of points FS .

Given such examples, you might be tempted to conjecture that *all* such forgetful functors into \mathbf{Set} are representable. But not so. Consider \mathbf{FinGrp} , the category of finite groups. Then

Theorem 139. *The forgetful functor $F: \mathbf{FinGrp} \rightarrow \mathbf{Set}$ is not representable,*

Proof. Suppose a putative representing group \mathcal{R} has r members, and take any group \mathcal{G} with $g > 1$ members, where g is coprime with r . Then it is well known that the only group homomorphism from \mathcal{R} to \mathcal{G} is the trivial one that sends everything to the identity in \mathcal{G} . But then the underlying set of \mathcal{G} can't be in bijective correspondence with $\mathbf{FinGrp}(\mathcal{R}, \mathcal{G})$ as would be required for a naturality square proving that \mathcal{R} represented F . \square

Let's take another pair of examples. We first need to recall definitions from Chapter 15:

- (i) The (covariant) *powerset functor* $P: \mathbf{Set} \rightarrow \mathbf{Set}$ maps a set X to its powerset $\mathcal{P}(X)$ and maps a set-function $f: X \rightarrow Y$ to the function which sends $U \in \mathcal{P}(X)$ to its image $f[U] \in \mathcal{P}(Y)$.
- (ii) The *contravariant powerset functor* $\bar{P}: \mathbf{Set}^{op} \rightarrow \mathbf{Set}$ again maps a set to its powerset, and maps a set-function $f: Y \rightarrow X$ to the function which sends $U \in \mathcal{P}(X)$ to its inverse image $f^{-1}[U] \in \mathcal{P}(Y)$.

Theorem 140. *The contravariant powerset functor \bar{P} is represented by the set $2 = \{0, 1\}$; but the covariant powerset functor P is not representable.*

Representables and universal elements

Proof. As yet, we don't have any general principles about representables and non-representables which we can invoke to prove theorems such as this. So again we just need to labour through by applying definitions and seeing what we get.

If the contravariant functor \bar{P} is to be representable, then there must be a representing set R and a natural isomorphism ψ with components such that, for all set functions $f: Y \rightarrow X$, the following diagram always commutes:

$$\begin{array}{ccc}
 \bar{P}X & \xrightarrow{\bar{P}f} & \bar{P}Y \\
 \downarrow \psi_X & & \downarrow \psi_Y \\
 \text{Set}(X, R) & \xrightarrow{\text{Set}(f, R)} & \text{Set}(Y, R)
 \end{array}$$

Now $\text{Set}(X, R)$ is the set of set-functions from X to R , whose cardinality is $|R|^{|X|}$; and the cardinality of $\bar{P}X$, i.e. $\mathcal{P}(X)$, is $2^{|X|}$. So that forces R to be a two-membered set: so we pick the set $2 = \{0, 1\}$.

$\text{Set}(X, 2)$ is then the set of characteristic functions for subsets of X , i.e. the set of functions $c_U: X \rightarrow \{0, 1\}$ where $c_U(x) = 1$ iff $x \in U \subseteq X$. So the obvious next move is to take $\psi_X: \bar{P}X \rightarrow \text{Set}(X, 2)$ to be the isomorphism that sends a set $U \in \mathcal{P}(X)$ to its characteristic function c_U .

With this choice, the diagram always commutes. Chase the element $U \in \bar{P}X$ around. The route via the north-east node takes us from $U \subseteq X$ to $f^{-1}[U] \subseteq Y$ to its characteristic function, i.e. the function which maps $y \in Y$ to 1 iff $f(y) \in U$. Meanwhile, the route via the south-west node takes us first from $U \subseteq X$ to c_U , and then we apply $\text{Set}(f, 2)$, which maps $c_U: X \rightarrow 2$ to $c_U \circ f: Y \rightarrow 2$, which again is the function which maps $y \in Y$ to 1 iff $f(y) \in U$. Which establishes the first half of the theorem.

For the second half of the theorem, we just note that if we try to run a similar argument for the covariant functor P , we'd need to find a representing set R' such that PX and $\text{Set}(R', X)$ are always in bijective correspondence. But $\text{Set}(R', X)$ is the set of set-functions from R' to X , whose cardinality is $|X|^{|R'|}$, while the cardinality of PX is $2^{|X|}$. And there is no choice of R' which will make these equal for varying X . \square

26.5 Universal elements

Back, though, to the basic idea. Concentrate on the covariant functors (we will mostly do this for a couple of sections, letting duality take care of contravariant cases). We say that a functor $F: \mathcal{C} \rightarrow \text{Set}$ is representable iff there is some hom-functor $\mathcal{C}(A, -): \mathcal{C} \rightarrow \text{Set}$ such that $F \cong \mathcal{C}(A, -)$. And then A is said to be a representation of F .

We might prefer to say, however, that a full certificate for the representability of F comprises not just the object A such that $F \cong \mathcal{C}(A, -)$ but also the required

natural isomorphism $\psi: \mathcal{C}(A, -) \xrightarrow{\cong} F$. In this spirit we might call the pair (A, ψ) the *full* representation of F .

Now, the Yoneda Lemma – or more exactly, Theorem 129 proved en route to the full Lemma – tells us more about natural transformations from $\mathcal{C}(A, -) \rightarrow F$. We can picture the situation like this:

$$\begin{array}{ccc} FA & \xrightarrow{\mathcal{X}_{AF}} & \text{Nat}(\mathcal{C}(A, -), F) \\ a & \longmapsto & \alpha: \mathcal{C}(A, -) \rightarrow F \\ & & \alpha_Z: \mathcal{C}(A, Z) \rightarrow FZ \\ & & g \longmapsto Fg(a) \end{array}$$

That is to say, there is a bijection \mathcal{X}_{AF} between the members of FA and the members of $\text{Nat}(\mathcal{C}(A, -), F)$. This bijection matches up $a \in FA$ with the natural transformation $\alpha = \mathcal{X}_{AF}(a): \mathcal{C}(A, -) \rightarrow F$. And this is the transformation whose Z -component α_Z sends a map $g: A \rightarrow Z$ to $Fg(a)$.

Therefore, instead of saying that a full certificate for the representability of F is a pair (A, ψ) , with $A \in \mathcal{C}$ and $\psi: \mathcal{C}(A, -) \xrightarrow{\cong} F$, we could equivalently invoke the pair (A, a) , with $A \in \mathcal{C}$ and $a \in FA$, where $\mathcal{X}_{AF}(a) = \psi$.

Now note that, since ψ is an isomorphism, each Z -component of $\mathcal{X}_{AF}(a)$ has to be an isomorphism; which means that for each $z \in FZ$ there must be a unique $g: A \rightarrow Z$ such that $Fg(a) = z$.

Which all goes to motivate introducing the following concept (even if it doesn't yet explain the label for the notion):

Definition 117. A *universal element* of the functor $F: \mathcal{C} \rightarrow \mathbf{Set}$ is a pair (A, a) , where $A \in \mathcal{C}$ and $a \in FA$, and where for each $Z \in \mathcal{C}$ and $z \in FZ$, there is a unique map $g: A \rightarrow Z$ such that $Fg(a) = z$. \triangle

The story for contravariant functors, by the way, will be exactly the same, except that the map g will go the other way about, $g: Z \rightarrow A$.

Theorem 141. A functor $F: \mathcal{C} \rightarrow \mathbf{Set}$ is representable by A iff it has a universal element (A, a) .

Proof. Our motivating remarks have already established the 'only if' direction; so we only have to prove the converse.

Suppose, therefore, that (A, a) is a universal element for F . Then, $a \in FA$, and there is a natural transformation $\chi = \mathcal{X}_{AF}(a): \mathcal{C}(A, -) \rightarrow F$ whose component $\chi_Z: \mathcal{C}(A, Z) \rightarrow FZ$ sends a map $g: A \rightarrow Z$ to $Fg(a)$.

We need to show χ_Z has an inverse. But the definition of a universal element tells in effect that there's a function δ_Z which sends $z \in FZ$ to the unique $g: A \rightarrow Z$ in $\mathcal{C}(A, Z)$ where $Fg(a) = z$. And we can immediately see that χ_Z and δ_Z are inverses.

So each component χ_Z is an isomorphism, and hence $\chi: \mathcal{C}(A, -) \xrightarrow{\cong} F$, witnessing that F is representable by A . \square

The proof of Theorem 129 also shows that the bijection \mathcal{X}_{AF} associates $\alpha_A(1_A)$ in FA with the natural transformation $\alpha: \mathcal{C}(A, -) \rightarrow F$. Hence,

Theorem 142. *If the functor $F: \mathcal{C} \rightarrow \mathbf{Set}$ has the full representation (A, α) , then F has the universal element $(A, \alpha_A(1_A))$.*

26.6 Categories of elements

(a) Why ‘universal element’? Because the definition invokes a universal mapping property: (A, a) is a universal element iff for every ... there is a unique map such that ... As in other cases, then, we might expect to be able to define a wider category in which universal elements appear as special cases picked out by this universal mapping property. So here goes:

Definition 118. $\mathbf{Els}_{\mathcal{C}}(F)$, the *category of elements of the functor $F: \mathcal{C} \rightarrow \mathbf{Set}$* , has the following data:

- (1) Objects are the pairs (A, a) , where $A \in \mathcal{C}$ and $a \in FA$.
- (2) An arrow from (A, a) to (B, b) is a \mathcal{C} -arrow $f: A \rightarrow B$ such that $Ff(a) = b$.
- (3) The identity arrow on (A, a) is 1_A .
- (4) Composition of arrows is induced by composition of \mathcal{C} -arrows.

It is easily checked that this *is* a category. (Alternative symbolism for the category includes variations on ‘ $\int_{\mathcal{C}} F$ ’.)

(b) Why ‘category of *elements*’? After all, functors don’t in a straightforward sense have elements. But we can perhaps throw some light on the name as follows.

- (i) Suppose we are given a category \mathcal{C} whose objects *are* sets (perhaps with some additional structure on them) and whose arrows are functions between sets. Then there will be some derived categories whose objects are (or involve) *elements* of \mathcal{C} ’s objects, and whose arrows between these elements are induced by the arrows between the containing sets.

Now such a category can be constructed in more than one way. But if we don’t want the derived category to forget about which elements belong to which sets, then a natural way to go would be to say that the objects of the derived category – which could be called the category of elements of \mathcal{C} – are all the pairs (A, a) for $A \in \mathcal{C}$, $a \in A$. And then given elements $a \in A$, $b \in B$, whenever there is a \mathcal{C} -arrow $f: A \rightarrow B$ such that $f(a) = b$, we’ll say that f is also an arrow from (A, a) to (B, b) in our new category. This derived category of elements in a sense unpacks what’s going on inside the original category \mathcal{C} .

- (ii) However, in the general case, \mathcal{C} ’s objects need not be sets so need not have elements. But a functor $F: \mathcal{C} \rightarrow \mathbf{Set}$ gives us a diagram of \mathcal{C} inside \mathbf{Set} ,

and of course the objects in the resulting diagram of \mathcal{C} *do* have elements. So we can consider the category of elements of F 's-diagram-of- \mathcal{C} , which – following the template in (i) – has as objects all the pairs (FA, a) for $A \in \mathcal{C}$, $a \in FA$. And then given elements $a \in FA$, $b \in FB$, whenever there is a **Set**-arrow $Ff: FA \rightarrow FB$ such that $Ff(a) = b$, we'll say that Ff is also an arrow from (FA, a) to (FB, b) in our new category.

Now, we can streamline that. Instead of taking the objects to be pairs (FA, a) take them simply to be pairs (A, a) (but where, still, $a \in FA$). And instead of talking of the arrow $Ff: FA \rightarrow FB$ we can instead talk more simply of $f: A \rightarrow B$ (but where, still, $Ff(a) = b$). And with that streamlining – lo and behold! – we are back with the category $\text{Elts}_{\mathcal{C}}(F)$, which is isomorphic to category of elements of F 's-diagram-of- \mathcal{C} , and which – as convention has it – we'll call the category of elements of F , for short.

So the construction of $\text{Elts}_{\mathcal{C}}(F)$ is tolerably natural.

(c) Here is another way of thinking of this category. Let 1 be some singleton in **Set**. Then what is the comma category $(1 \downarrow F)$? Applying the definition of such categories given in §19.4, the objects of this category are pairs (A, a) where $A \in \mathcal{C}$ and $a: 1 \rightarrow FA$ is an arrow in **Set**. And the arrows of the category from (A, a) to (B, b) is a \mathcal{C} -arrow $f: A \rightarrow B$ such that $b = Ff \circ a$.

But *that* is just the definition of $\text{Elts}_{\mathcal{C}}(F)$ except that we have traded in the requirement that a is *member* of FA for the requirement that a is an *arrow* $1 \rightarrow FA$. But as we well know by now, members of a set are in bijective correspondence with such arrows from a fixed singleton, and from a categorial perspective we can treat members as such arrows (hence our using the same label ' a ' here for both). Hence

Theorem 143. *For a given functor $F: \mathcal{C} \rightarrow \text{Set}$, the category $\text{Elts}_{\mathcal{C}}(F)$ is (isomorphic to) the comma category $(1 \downarrow F)$ where 1 is terminal in **Set**.*

(d) Having defined a category $\text{Elts}_{\mathcal{C}}(F)$ for universal elements of $F: \mathcal{C} \rightarrow \text{Set}$ to live in, we can finish by asking: how do we distinguish universal elements from other elements categorially? The answer is immediate from Defn. 117, which in our new terminology says:

Theorem 144. *An object $I = (A, a)$ in $\text{Elts}_{\mathcal{C}}(F)$ is a universal element iff, for every object E in $\text{Elts}_{\mathcal{C}}(F)$ there is exactly one morphism $f: I \rightarrow E$, so I is initial in $\text{Elts}_{\mathcal{C}}(F)$.*

But initial objects are unique up to unique isomorphism. Which, recalling what isomorphisms in $\text{Elts}_{\mathcal{C}}(F)$ are, implies

Theorem 145. *If (A, a) and (A', a') are universal elements for $F: \mathcal{C} \rightarrow \text{Set}$, then there is a unique \mathcal{C} -isomorphism $f: A \rightarrow A'$ such that $Ff(a) = a'$.*

26.7 Limits and exponentials as universal elements

(a) Let $\text{Cone}(C, D)$ be the set of cones over some diagram D with vertex C in some given category \mathcal{C} – and we will assume that \mathcal{C} is small enough for $\text{Cone}(C, D)$ indeed to be a set living in \mathbf{Set} .

We can now define a contravariant functor $\text{Cone}(-, D): \mathcal{C}^{op} \rightarrow \mathbf{Set}$ as follows.

- (i) $\text{Cone}(-, D)$ sends an object C to $\text{Cone}(C, D)$.
- (ii) $\text{Cone}(-, D)$ sends an arrow $f: C' \rightarrow C$ to $\text{Cone}(f, D): \text{Cone}(C, D) \rightarrow \text{Cone}(C', D)$, which takes a cone $[C, c_j]$ and sends it to $[C', c_j \circ f]$.

It is easily checked that this is indeed a functor.

We now apply the definition of universal elements, tweaked for the contravariant case. Then a universal element of the functor $\text{Cone}(-, D)$ is a pair $(L, [L, l_J])$, where L is in \mathcal{C} and $[L, l_J]$ is in $\text{Cone}(L, D)$, the set of cones over D with vertex L . And moreover, we require that for each $C \in \mathcal{C}$ and each cone $[C, c_J]$, there is a unique map $f: C \rightarrow L$ such that $\text{Cone}(f)[L, l_J] = [C, c_J]$, which requires $l_J \circ f = c_J$ for each J . But that's just to say that $[L, l_J]$ is a limit cone! Hence

Theorem 146. *In small enough categories, a limit cone over a diagram D is a universal element for $\text{Cone}(-, D)$.*

Since limits are therefore initial objects in an associated category of elements, they have to be unique up to a unique appropriate isomorphism, giving us another proof of Theorem 47.

(b) Consider the contravariant functor $\mathcal{C}(- \times B, C)$ which we met in §20.3 Ex. (7). This sends an object A in \mathcal{C} to the hom-set of arrows from $A \times B$ to C . And it sends an arrow $f: A' \rightarrow A$ to the map $- \circ f \times 1_B$ (i.e. to the map which takes an arrow $j: A \times B \rightarrow C$ and yields the arrow $j \circ f \times 1_B: A' \times B \rightarrow C$).

Now apply the definition of universal element for the contravariant case. Then a universal element of $\mathcal{C}(- \times B, C)$ is a pair (E, ev) , with E in \mathcal{C} and ev in $\mathcal{C}(E \times B, C)$, such that for every A and every $g \in \mathcal{C}(A \times B, C)$, there is a unique $\bar{g}: A \rightarrow E$ such that $\mathcal{C}(- \times B, C)(\bar{g})(ev) = g$, i.e. $ev \circ \bar{g} \times 1_B = g$.

But, trivially squaring up the brackets, a pair $[E, ev]$ with those properties is exactly the exponential $[C^B, ev]$. Hence

Theorem 147. *The exponential $[C^B, ev]$, when it exists in \mathcal{C} , is a universal element of $\mathcal{C}(- \times B, C)$.*

Since exponentials are therefore also initial objects in an associated category of elements, they too have to be unique up to a unique appropriate isomorphism, giving us this time another proof of Theorem 66.

27 Galois connections

We will have quite a lot more to say about functors, limits and representables and about how they interrelate after we have introduced the next really important Big Idea from category theory – namely, the idea of pairs of *adjoint functors* and the *adjunctions* they form.

Now, one option would be to dive straight into the general story about adjoints. But that multi-faceted story can initially seem rather complex, and it is quite easy to get lost in the details. So the plan here is to start by looking first at a very restricted class of cases. These are the so-called Galois connections, which are in effect adjunctions between two categories which are posets. In this chapter, then, we discuss these Galois connections in an elementary way, as a way of introducing us to some key themes. And for the moment, we largely suppress the categorial context.

27.1 (Probably unnecessary) reminders about posets

Recall: The set C equipped with the binary relation \leq , which we denote (C, \leq) , is a poset just in case \leq is a partial order – i.e., for all $x, y, z \in C$, (i) $x \leq x$, (ii) if $x \leq y$ and $y \leq z$ then $x \leq z$, (iii) if $x \leq y$ and $y \leq x$ then $x = y$. (We will, as appropriate, recruit ‘ \sqsubseteq ’, ‘ \preceq ’, ‘ \sqsubset ’ as other symbols for partial orders.)

Reversing a partial order gives us another partial order. Hence reversing the order in a poset $\mathcal{C} = (C, \leq)$ gives us a dual poset $\mathcal{C}^{op} = (C, \geq)$ defined in the obvious way.

There is a related notion of a strict poset defined in terms of a strict partial order $<$, where $x < y$ iff $x \leq y \wedge x \neq y$ for some partial order \leq . It is just a matter of convenience whether we concentrate on the one flavour of poset or the other, and you will already be familiar with a variety of examples of ‘naturally occurring’ posets of both flavours.

The following notions will also be entirely familiar, in one terminology or another:

Definition 119. Suppose that $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$ are two posets. Let the map $F : \mathcal{C} \rightarrow \mathcal{D}$ be a function between the carrier sets C and D . Then

- (1) F is monotone just in case, for all $x, y \in C$, if $x \leq y$ then $Fx \sqsubseteq Fy$;

- (2) F is an order-embedding just in case, for all $x, y \in C$, $x \leq y$ iff $Fx \sqsubseteq Fy$;
- (3) F is an order-isomorphism iff F is a surjective order-embedding. \triangle

Some obvious remarks about these notions:

- i. Monotone maps compose to give monotone maps and composition is associative. Likewise for order-embeddings and order-isomorphisms.
- ii. Order-embeddings are injective. Keeping the same notation, suppose $Fx = Fy$ and hence both $Fx \sqsubseteq Fy$ and $Fy \sqsubseteq Fx$. Then, if F is an embedding, $x \leq y$ and $y \leq x$, and hence $x = y$.
- iii. If $F[C]$ is C 's image under F , an order-embedding $F: (C, \leq) \rightarrow (D, \sqsubseteq)$ is an order-isomorphism from (C, \leq) to $(F[C], \sqsubseteq)$.
- iv. An order-isomorphism is bijective, and therefore is an isomorphism as a set-function. Order-isomorphisms have unique inverses which are also order-isomorphisms.
- v. Posets are deemed isomorphic if there is an order-isomorphism between them.

If (C, \leq) is a poset and $X \subseteq C$, then a maximum of X (with respect to the inherited order \leq) is defined in the obvious way: m is a maximum of X iff $m \in X \wedge (\forall x \in X)x \leq m$. Maxima are unique when they exist – for if $m, m' \in X$ are both maxima, $m' \leq m$ and similarly $m \leq m'$ and hence $m = m'$.

If $X \subseteq C$ we say that (X, \leq) is a sub-poset of (C, \leq) ; and note here that we will not routinely fuss to distinguish a relation defined over C from the restriction of that relation to X .

Definition 120. Suppose Π is a collection of sets. Then Π ordered by inclusion, i.e. (Π, \subseteq) , is an *inclusion poset*. \triangle

Theorem 148. *Every poset is isomorphic to an inclusion poset.*

Proof. Take the poset (C, \leq) . For each $y \in C$, now form the set containing it and its \leq -predecessors $\pi_y = \{x \in C \mid x \leq y\}$. Let Π the set of all π_y for $y \in C$. Then (Π, \subseteq) is an inclusion poset.

Define $F: (C, \leq) \rightarrow (\Pi, \subseteq)$ by putting $Fx = \pi_x$. Then F is very easily seen to be a bijection, and also $x \leq y$ iff $\pi_x \subseteq \pi_y$. So F is an order-isomorphism. \square

27.2 An introductory example

We rather informally describe what will turn out to be an important instance of a Galois connection: we choose notation with an eye to smoothing the transitions to later generalizations.

Suppose, then, that we have a poset $\mathcal{C} = (C, \leq)$ where the members of C are sets of sentences from some suitable formal language \mathcal{L} (the details of \mathcal{L} won't

matter too much), and \leq is simply set-inclusion. We can think of the members of C as *theories* couched in the language \mathcal{L} ; these theories are then partially ordered from less specific (saying less) to more specific (saying more).

There is a corresponding poset $\mathcal{D} = (D, \sqsubseteq)$ where the members of D are collections of \mathcal{L} -structures, i.e. sets of potential models for theories couched in \mathcal{L} ; and we will take \sqsubseteq to be the *converse* of inclusion. A member of D can be thought of as a set of alternative model ‘worlds’ a theory could be true of; these sets of models are then also partially ordered from less specific (more alternatives) to more specific (a narrower range).

There are then two very natural maps between these posets.

- i. $F: \mathcal{C} \rightarrow \mathcal{D}$ sends a theory $c \in C$ to $d \in D$, where d is the set of models of c (i.e. d is the set containing each model on which all the sentences in c are true).
- ii. $G: \mathcal{D} \rightarrow \mathcal{C}$ sends a set of models d to the set c containing each sentence which is true on every model in d .

Put it this way: F is the ‘find the models’ function. It takes a bunch of sentences and returns all its models, the set of structures where the sentences in the bunch are all true. In the other direction, G is the equally natural ‘find all the true sentences’ function. It takes a bunch of structures and returns the set of sentences that are true in all of those structures.

In general F and G will not be inverse to each other. But the mapping functions do interrelate in the following nice ways:

- (1) F and G are monotone.

And for all $c \in C$, $d \in D$,

- (2) $c \leq GFc$ and $FGd \sqsubseteq d$,
- (3) $Fc \sqsubseteq d$ iff $c \leq Gd$.

And further

- (4) $FGF = F$ and $GFG = G$.

Why so? For (1) we note that if the theory c' is more informative than c , then it will be true of a narrow range of possible models. And conversely, if d' is a narrower range of models than d , then more sentences will be true of everything in d' than are true of everything in d .

For the first half of (2) we note that if we start with a bunch of sentences c , look at the models where they are all true together, and then look at the sentences true in all those models together, we’ll get back original sentences in c plus all their consequences (where consequence is defined in the obvious way in terms of preservation of truth in the relevant set of structures).

For the other half of (2) we note that if we start from a collection of models d , find the sentences true in all of them, and then look at the models for those

sentences, we must get back at least the models we started with, maybe more. (Remember, \sqsupseteq is the converse of inclusion!)

For (3) we note that if the models where all the sentences of c are true include all those in d then the theory c must be included in the set of sentences true in all the models in d , and vice versa.

For the first half of (4) we note that the models of a set of sentences c together with their consequences are just the models of the original c . Similarly for the other half.

So in summary: we have here a pair of posets $\mathcal{C} = (C, \leq)$, $\mathcal{D} = (D, \sqsupseteq)$ and a pair of functions $F: \mathcal{C} \rightarrow \mathcal{D}$ and $G: \mathcal{D} \rightarrow \mathcal{C}$ for which conditions (1) to (4) hold. We will see in the next section that this situation is repeatedly realized in different contexts.

27.3 Galois connections defined

We now generalize. However, as we'll see in the next section, conditions (1) to (4) are not independent. The first two together imply the third and fourth, and the third implies the rest. Simply because it is prettier, then, we plump in this section for a general definition just in terms of the third condition (which we relabel):

Definition 121. Suppose that $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsupseteq)$ are two posets, and let $F: \mathcal{C} \rightarrow \mathcal{D}$ and $G: \mathcal{D} \rightarrow \mathcal{C}$ be a pair of functions such that for all $c \in C$, $d \in D$,

$$(G) \quad Fc \sqsupseteq d \text{ iff } c \leq Gd.$$

Then F and G form a *Galois connection* between \mathcal{C} and \mathcal{D} . When this holds, we write $F \dashv G$, and F is said to be the *left adjoint* of G , and G the right adjoint of F .¹ △

The first discussion of a version of such a connection $F \dashv G$ – and hence the name – is to be found in Evariste Galois's work in what has come to be known as Galois theory, a topic beyond our purview here. And there are plenty of other serious mathematical examples (e.g. from number theory, abstract algebra and topology) of two posets with a Galois connection between them. But we really don't want to get bogged down in unnecessary mathematics at this early stage; so for the moment let's just give some simple cases, to add to our informally described motivating example in the last section:

¹Talk of adjoints here seems to have been originally borrowed from the old theory of Hermitian operators, where in e.g. a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ the operators A and A^* are said to be adjoint when we have, generally, $\langle Ax, y \rangle = \langle x, A^*y \rangle$. The formal analogy is evident.

- (1) Suppose F is an order-isomorphism between (C, \leq) and (D, \sqsubseteq) : then F^{-1} is an order-isomorphism in the reverse direction. Take $c \in C, d \in D$: then trivially $Fc \sqsubseteq d$ iff $F^{-1}Fc \leq F^{-1}d$ iff $c \leq F^{-1}d$. Hence $F \dashv F^{-1}$.
- (2) Take $\mathcal{N} = (\mathbb{N}, \leq)$ and $\mathcal{Q}^+ = (\mathbb{Q}^+, \leq)$, i.e. the naturals and the non-negative rationals in their standard orders. Let $I: \mathcal{N} \rightarrow \mathcal{Q}^+$ be the injection function which maps a natural number to the corresponding rational integer, and let $F: \mathcal{Q}^+ \rightarrow \mathcal{N}$ be the ‘floor’ function which maps a rational to the natural corresponding to its integral part. Then $I \dashv F$ is a Galois connection from \mathcal{N} to \mathcal{Q} . Likewise if $C: \mathcal{Q}^+ \rightarrow \mathcal{N}$ is the ‘ceiling’ function which maps a rational to the smallest integer which is at least as big, then $C \dashv I$ is a Galois connection going in the opposite direction.
- (3) Let $f: X \rightarrow Y$ be some function between two sets X and Y . It induces a function $F: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ between their powersets which sends $A \subseteq X$ to $f[A]$, and another function $F^{-1}: \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$ which sends $B \subseteq Y$ to its pre-image under f , $F^{-1}[B] = \{x \in X \mid f(x) \in B\}$. Then $F \dashv F^{-1}$ is a Galois connection between the inclusion posets $(\mathcal{P}(X), \subseteq)$ and $(\mathcal{P}(Y), \subseteq)$.
- (4) Take any poset $\mathcal{C} = (C, \leq)$, and let 1 be a one object poset, i.e. of the form $(\{0\}, =)$. Let $F: \mathcal{C} \rightarrow 1$ be the only possible function, the trivial one which sends everything to 0 . Then F has a right adjoint $G: 1 \rightarrow \mathcal{C}$ just if it is the case that, for any $c \in C$, $Fc = 0$ iff $c \leq G0$. So F has a right adjoint just in case \mathcal{C} has a maximum, and then G sends 1 ’s only element to it. Dually, F has a left adjoint just in case \mathcal{C} has a minimum, and then the left adjoint G' sends 1 ’s only element to *that*.
- (5) Our next example is from elementary logic. Choose a favourite logical proof-system \mathcal{L} – it could be classical or intuitionistic, or indeed any other logic, so long as it has a normally-behaved conjunction and conditional connectives and a sensible deducibility relation. Let $\alpha \vdash \beta$ notate, as usual, that there is a formal \mathcal{L} -proof from premiss α to conclusion β . Then let $|\alpha|$ be the equivalence class of wffs of the system interderivable with α . Take E to be set of all such equivalence classes, and put $|\alpha| \leq |\beta|$ in E iff $\alpha \vdash \beta$. Then it is easily checked that (E, \leq) is a poset.
 Now consider the following two functions between (E, \leq) and itself. Fix γ to be some \mathcal{L} -wff. Then let F send the equivalence class $|\alpha|$ to the class $|(\gamma \wedge \alpha)|$, and let G send $|\alpha|$ to the class $|(\gamma \rightarrow \alpha)|$.
 Given our normality assumption, $\gamma \wedge \alpha \vdash \beta$ if and only if $\alpha \vdash \gamma \rightarrow \beta$. Hence $|\gamma \wedge \alpha| \leq |\beta|$ iff $|\alpha| \leq |\gamma \rightarrow \beta|$. That is to say $F|\alpha| \leq |\beta|$ iff $|\alpha| \leq G|\beta|$. Hence we have a Galois connection $F \dashv G$ between (E, \leq) and itself, and in a slogan, ‘Conjunction is left adjoint to conditionalization’.
- (6) Our last example for the moment is another example from elementary logic. Let \mathcal{L} now be a first-order logic, and consider the set of \mathcal{L} -wffs with at most the variables \vec{x} free.

We will write $\varphi(\vec{x})$ for a formula in this class, $|\varphi(\vec{x})|$ for the class of formulae interderivable with $\varphi(\vec{x})$, and $E_{\vec{x}}$ for the set of such equivalence classes of formulae with at most \vec{x} free. Using \leq as in the last example, $(E_{\vec{x}}, \leq)$ is a poset for any choice of variables \vec{x} .

We now consider two maps between the posets $(E_{\vec{x}}, \leq)$ and $(E_{\vec{x}, y}, \leq)$. In other words, we are going to be moving between (equivalence classes of) formulae with at most \vec{x} free, and (equivalence classes of) formulae with at most \vec{x}, y free – where y is a new variable not among the \vec{x} .

First, since every wff with at most the variables \vec{x} free also has at most the variables \vec{x}, y free, there is a trivial map $F: E_{\vec{x}} \rightarrow E_{\vec{x}, y}$ that sends the class of formulas $|\varphi(\vec{x})|$ in $E_{\vec{x}}$ to the same class of formulas which is also in $E_{\vec{x}, y}$.

Second, we define the companion map $G: E_{\vec{x}, y} \rightarrow E_{\vec{x}}$ that sends $|\varphi(\vec{x}, y)|$ in $E_{\vec{x}, y}$ to $|\forall y \varphi(\vec{x}, y)|$ in $E_{\vec{x}}$.

Then $F \dashv G$, i.e. we have another Galois connection. For that is just to say

$$F(|\varphi(\vec{x})|) \leq |\psi(\vec{x}, y)| \quad \text{iff} \quad |\varphi(\vec{x})| \leq G(|\psi(\vec{x}, y)|).$$

Which just reflects the familiar logical rule that

$$\varphi(\vec{x}) \vdash \psi(\vec{x}, y) \quad \text{iff} \quad \varphi(\vec{x}) \vdash \forall y \psi(\vec{x}, y),$$

so long as y is not free in $\varphi(\vec{x})$. Hence universal quantification is right-adjoint to a certain trivial inclusion operation.

And we can exactly similarly show that existential quantification is left-adjoint to the same operation.

Some morals. Our first example shows that Galois connections are at least as plentiful as order-isomorphisms: and such an isomorphism will have a right adjoint and left adjoint which are the same (i.e. both are the isomorphism's inverse). The second and fourth cases show that posets that aren't order-isomorphic can in fact still be Galois connected. The third case shows that posets can have many Galois connections between them (as any $f: X \rightarrow Y$ generates a connection between the inclusion posets on the powersets of X and Y). The fourth example gives a case where a function has both a left and a right adjoint which are different. The fourth and sixth cases give a couple of illustrations of how a significant construction (taking maxima, forming a universal quantification respectively) can be regarded as adjoint to some quite trivial operation. The fifth example, like the third, shows that even when the Galois-connected posets are isomorphic (in the fifth case trivially so, because they are identical!), there can be a pair of functions which aren't isomorphisms but which also go to make up a connection between the posets. And the fifth and sixth examples, like the motivating example in the previous section, illustrate why Galois connections are of interest to logicians.

27.4 Galois connections re-defined

The following theorem is basic:

Theorem 149. *Suppose that $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$ are posets with maps $F: \mathcal{C} \rightarrow \mathcal{D}$ and $G: \mathcal{D} \rightarrow \mathcal{C}$ between them. Then $F \dashv G$ iff and only if*

- (1) F and G are both monotone, and
- (2) for all $c \in C$, $d \in D$, $c \leq GFc$ and $FGd \sqsubseteq d$, and
- (3) $FGF = F$ and $GFG = G$.

Proof. (If) Assume conditions (1) and (2) both hold. And suppose $Fc \sqsubseteq d$. Since by (1) G is monotone, $GFc \leq Gd$. But by (2) $c \leq GFc$. Hence by transitivity $c \leq Gd$. That establishes one half of the biconditional (G). We don't need (3) here. The proof of the other half is dual.

(Only if) Suppose (G) is true. Then in particular, $Fc \sqsubseteq Fc$ iff $c \leq GFc$. Since \sqsubseteq is reflexive, $c \leq GFc$. Similarly for the other half of (2).

Now, suppose also that $c \leq c'$. Then since we've just shown $c' \leq GFc'$, we have $c \leq GFc'$. But by (G) we have $Fc \sqsubseteq Fc'$ iff $c \leq GFc'$. Whence, $Fc \sqsubseteq Fc'$ and F is monotone. Similarly for the other half of (1).

For (3), since for any $c \in C$, $c \leq GFc$, and also F is monotone, it follows that $Fc \sqsubseteq FGFc$.

But the fundamental condition (G) yields $FGFc \sqsubseteq Fc$ iff $GFc \leq GFc$. The r.h.s. is trivially true, so $FGFc \sqsubseteq Fc$.

By the antisymmetry of \sqsubseteq , then, $FGFc = Fc$. Since c was arbitrary, $FGF = F$. Similarly for the other half of (3). \square

This theorem means that, as already intimated at the end of §27.2, we could equally well have defined a Galois connection like this:

Definition 122 (Alternative). Suppose that $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$ are two posets, and let $F: \mathcal{C} \rightarrow \mathcal{D}$ and $G: \mathcal{D} \rightarrow \mathcal{C}$ be a pair of functions such that for all $c \in C$, $d \in D$,

- (1) F and G are both monotone, and
- (2) for all $c \in C$, $d \in D$, $c \leq GFc$ and $FGd \sqsubseteq d$, and
- (3) $FGF = F$ and $GFG = G$.

Then F and G form a Galois connection between \mathcal{C} and \mathcal{D} . \triangle

Two comments about this. First, our proof of Theorem 149 shows that we needn't have explicitly given clause (3) in our alternative definition as it follows from the other two. We include it because when we move on from the case of Galois connections to discuss adjunctions more generally, again giving two definitions, we will need to explicitly mention the analogue of clause (3).

Second, note that we could replace clause (2) with the equivalent clause

- (2') (i) if $c \leq c'$, then both $c \leq c' \leq GFc'$ and $c \leq GFc \leq GFc'$; and
(ii) if $d \sqsubseteq d'$, then both $FGd \sqsubseteq d \sqsubseteq d'$ and $FGd \sqsubseteq FGd' \sqsubseteq d'$.

For trivially (2') implies (2); conversely (1) and (2) imply (2'). Again, we mention this variant on our alternative definition of Galois connections for later use when we come to generalize.

27.5 Some basic results about Galois connections

(a) We now have a pair of equivalent definitions of Galois connections, and a small range of elementary examples. In this section we start by proving a couple of theorems that show that such connections behave just as you would hope, in two different respects. First, if there is a connection between \mathcal{C} and \mathcal{D} and a connection between \mathcal{D} and \mathcal{E} then they can be composed to give a connection between \mathcal{C} and \mathcal{E} . And second, inside a Galois connection, a left adjoint uniquely fixes its right adjoint, and vice versa. Thus:

Theorem 150. *Suppose there is a Galois connection $F \dashv G$ between the posets $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$, and a connection $H \vdash K$ between the posets \mathcal{D} and $\mathcal{E} = (E, \sqsubseteq)$. Then there is a Galois connection $HF \dashv GK$ between \mathcal{C} and \mathcal{E} .*

Proof. Take any for any $c \in C, e \in E$. Then, using the first connection, we have $Fc \sqsubseteq Ke$ iff $c \leq GKe$. And by the second connection, we have $HFc \sqsubseteq e$ iff $Fc \sqsubseteq Ke$.

Hence $HFc \sqsubseteq e$ iff $c \leq GKe$. Therefore $HF \dashv GK$. \square

Theorem 151. *If we have Galois connections $F \vdash G, F \vdash G'$ between the posets (C, \leq) and (D, \sqsubseteq) , then $G = G'$. Likewise, if $F \vdash G, F' \vdash G$ are both Galois connections between the same posets, then $F = F'$.*

Proof. We prove the first part. $F \vdash G'$ implies, in particular, that for any $d \in D$, $FG'd \sqsubseteq d$ iff $Gd \leq G'd$.

But by Theorem 149, applied to the connection $F \vdash G$, we have $FGd \sqsubseteq d$. So we can infer that, indeed, $Gd \leq G'd$.

By symmetry, $G'd \leq Gd$. But d was arbitrary, so indeed $G = G'$. \square

Careful, though! This second theorem does not say that, for any F which maps between (C, \leq) and (D, \sqsubseteq) , there must actually exist a unique corresponding G in the reverse direction such that $F \dashv G$ (this isn't true as we saw in §27.3 Ex. (4)). Nor does it say that when there is a Galois connection between the posets, it is unique (our toy examples have already shown that that is false too). The claim is only that, if you are given a possible left adjoint – or a possible right adjoint – there can be at most one candidate for its companion to complete a connection.

(b) Given that adjoint functions determine each other, we naturally seek an explicit definition of one in terms of the other. Here it is:

Theorem 152. *If $F \dashv G$ is a Galois connection between the posets (C, \leq) and (D, \sqsubseteq) , then*

- (1) $Gd = \text{the maximum of } \{c \in C \mid Fc \sqsubseteq d\}$,
- (2) $Fc = \text{the minimum of } \{d \in D \mid c \leq Gd\}$.

Proof. We argue for (1), leaving the dual (2) to take care of itself. Fix on an arbitrary $d \in D$ and for brevity, put $\Sigma = \{c \in C \mid Fc \sqsubseteq d\}$.

Theorem 149 tells us that (i) for any $u \in C$, $u \leq GFu$, (ii) $FGd \sqsubseteq d$, and (iii) G is monotone. So by (ii), $Gd \in \Sigma$.

Now suppose $u \in \Sigma \subseteq C$. Then $Fu \sqsubseteq d$. By (iii), $GFu \leq Gd$. Whence from (i), $u \leq Gd$.

That shows Gd is both a member of and an upper bound for Σ , i.e. is a maximum for Σ . \square

Recall the posets $\mathcal{N} = (\mathbb{N}, \leq)$ and $\mathcal{Q}^+ = (\mathbb{Q}^+, \leq)$ with the injection map $I: \mathcal{N} \rightarrow \mathcal{Q}^+$ and floor function $F: \mathcal{Q}^+ \rightarrow \mathcal{N}$ which maps a rational to the natural corresponding to its integral part. Then we remarked before that $I \dashv F$. Now we note that $F \dashv I$ is false. Indeed, there can be no connection of the form $F \dashv G$ from \mathcal{Q}^+ to \mathcal{N} . For $Fq = 1$ iff $1 \leq q < 2$, and hence $\{q \in \mathbb{Q}^+ \mid Fq \leq 1\}$ has no maximum, and so there can be no right adjoint to F .

Generalizing, we have the following:

Theorem 153. *Galois connections are not necessarily symmetric. That is to say, given $F \dashv G$ is a Galois connection between the posets \mathcal{C} and \mathcal{D} , it does not follow that $G \dashv F$ is a connection between \mathcal{D} and \mathcal{C} .*

27.6 Fixed points, isomorphisms, and closures

Theorems 150 and 151 tell us that Galois connections are rather nicely behaved. This section now explores some of the consequences of there being a Galois connection $F \dashv G$ between two posets.

(a) Theorem 149 tells us, in particular, where to find the fixed points of the composite maps GF and FG :

Theorem 154. *Given a Galois connection $F \dashv G$ between the posets (C, \leq) and (D, \sqsubseteq) , then*

- (1) $c \in G[D]$ iff c is a fixed point of GF ; $d \in F[C]$ iff d is a fixed point of FG .
- (2) $G[D] = (GF)[C]$; $F[C] = (FG)[D]$.

Proof. (1) Suppose $c \in G[D]$. Then for some $d \in D$, $c = Gd$ and hence $GFc = GF Gd = Gd = c$, so c is a fixed point of GF . Conversely suppose $GFc = c$. Then c is the value of Gd for $d = Fc$, and therefore $c \in G[D]$.

Hence $c \in G[D]$ iff c is a fixed point of GF . The other half of (1) is dual.

(2) We have just seen that if $c \in G[D]$ then $c = GFc$ so $c \in (GF)[C]$. Therefore $G[D] \subseteq (GF)[C]$. Conversely, suppose $c \in (GF)[C]$, then for some $c' \in C$, $c = GFc'$; but $Fc' \in D$ so $c \in G[D]$. Therefore $(GF)[C] \subseteq G[D]$.

Hence $G[D] = (GF)[C]$. The other half of (2) is dual. \square

(b) We know that a pair of posets which have a Galois connection between them needn't be isomorphic overall. But this next theorem says that they will typically contain an interesting pair of isomorphic sub-posets (alongside the trivially isomorphic one-object posets!).

Definition 123. Suppose $F \dashv G$ is a Galois connection between the posets $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$. Put $C^{-1} = G[D]$ and $D^{-1} = F[C]$. Then we define $\mathcal{C}^{-1} = (C^{-1}, \leq)$ and $\mathcal{D}^{-1} = (D^{-1}, \sqsubseteq)$. \triangle

Theorem 155. If $F \dashv G$ is a Galois connection between the posets $\mathcal{C} = (C, \leq)$ and $\mathcal{D} = (D, \sqsubseteq)$, then \mathcal{C}^{-1} and \mathcal{D}^{-1} are order-isomorphic.

Proof. We show that F restricted to C^{-1} provides the desired order isomorphism.

Note first that if $c \in C^{-1}$, then $Fc \in F[C] = D^{-1}$. So F as required sends elements of C^{-1} to elements of D^{-1} . Moreover every element of D^{-1} is Fu for some $u \in C^{-1}$. For if $d \in F[C]$, then for some c , $d = Fc = FGFc = Fu$ where $u = GFc \in G[D] = C^{-1}$.

So F restricted to C^{-1} is onto D^{-1} . It remains to show that it is an order-embedding. We know that F will be monotone, so what we need to prove is that, if $c, c' \in C^{-1}$ and $Fc \sqsubseteq Fc'$, then $c \leq c'$.

But if $Fc \sqsubseteq Fc'$, then by the monotonicity of G , $GFc \leq GFc'$. Recall, though, that $c, c' \in C^{-1} = G[D]$ are fixed points of GF . Hence $c \leq c'$ as we want. \square

(c) Finally, we want the idea of a closure function K on a poset which, roughly speaking, maps a poset 'upwards' to a subposet which then stays fixed under further applications of K :

Definition 124. Suppose $\mathcal{C} = (C, \leq)$ is a poset; then a *closure function* on \mathcal{C} is a function $K: \mathcal{C} \rightarrow \mathcal{C}$ such that, for all $c, c' \in C$,

- (1) $c \leq Kc$;
- (2) if $c \leq c'$, then $Kc \leq Kc'$, i.e. K is monotone;
- (3) $KKc = Kc$, i.e. K is idempotent. \triangle

Theorem 156. If $F \vdash G$ is a Galois connection between \mathcal{C} and another poset, then GF is a closure function for \mathcal{C} .

Proof. We quickly check that the three conditions for closure apply. (i) is given by Theorem 149. (ii) is immediate as GF is a composition of monotone functions. And for (iii), we know that $FGF = F$, and hence $GFGF = GF$. \square

27.7 One way a Galois connection can arise

The last three sections have been about Galois connections in general, and reveal that they have a perhaps surprisingly rich structure. In this section, we now note one characteristic way in which connections can arise.

Theorem 157. *Let R be a binary relation between members of X and members of Y . We define posets on the powersets, $\mathcal{C} = (\mathcal{P}(X), \subseteq)$, $\mathcal{D} = (\mathcal{P}(Y), \supseteq)$ – note the order reversal.*

Define $F: \mathcal{C} \rightarrow \mathcal{D}$ by putting $FA = \{b \mid (\forall a \in A)aRb\}$ for $A \subseteq X$. Similarly define $G: \mathcal{D} \rightarrow \mathcal{C}$ by putting $GB = \{a \mid (\forall b \in B)aRb\}$ for $B \subseteq Y$.

Then $F \dashv G$.

Proof. We just have to prove that principle (G) holds, i.e. for any $A \subseteq X$, $B \subseteq Y$, $FA \supseteq B$ iff $A \subseteq GB$.

But simply by applying definitions we see $FA \supseteq B$ iff $(\forall b \in B)(\forall a \in A)aRb$ iff $(\forall a \in A)(\forall b \in B)aRb$ iff $A \subseteq GB$. \square

Let's say that Galois connection produced in this way is *relation-generated*. Galois's original classic example was of this kind. And our original motivating example, which we return to in the next section, is relation-generated too.

27.8 Syntax and semantics briefly revisited

(a) In his famous *Dialectica* paper 'Adjointness in foundations' (1969), F. William Lawvere writes of 'the familiar Galois connection between sets of axioms and classes of models, for a fixed [signature]'. This is in fact the motivating example which we presented very informally in §27.2. We will very briefly revisit it.

Let \mathcal{L} be a formal language. Then a set of \mathcal{L} -axioms in the wide sense that Lawvere is using is just any old set of \mathcal{L} -sentences. And by talk of 'models', Lawvere means structures apt for interpreting \mathcal{L} 's. (We'll cheerfully sidestep issues of size by assuming that there's only a set's-worth of such structures.)

We defined two posets. First, $\mathcal{C} = (C, \leq)$, where C is a collection of sets of \mathcal{L} -sentences, and the ordering is set-inclusion. Second, $\mathcal{D} = (D, \supseteq)$, where D is a collection of sets of \mathcal{L} -structures, and the ordering is the inverse of set-inclusion. Then we met two functions which we can define like this (using φ, σ as variables over sentences and structures respectively)

$$(1) F: \mathcal{C} \rightarrow \mathcal{D} \text{ is such that } Fc = \{\sigma \mid (\forall \varphi \in c) \sigma \models \varphi\},$$

Galois connections

(2) $G: \mathcal{D} \rightarrow \mathcal{C}$ is such that $Gd = \{\varphi \mid (\forall \sigma \in d) \sigma \models \varphi\}$,

where $\sigma \models \varphi$ if φ is true interpreted in the structure σ .

Put like that, Theorem 157 (with the generating relation R between a sentence and a structure the converse of \models) immediately gives us

Theorem 158. $F \dashv G$ is a Galois connection between \mathcal{C} and \mathcal{D} .

(b) Now we can just turn the handle, and apply all those general theorems about Galois connections from the preceding sections to our special case of the connection between the ‘syntax’ \mathcal{C} and ‘semantics’ \mathcal{D} , recovering the sorts of results listed at the end of §27.2 and more. Of course, we get no exciting new logical news this way. But that’s not the name of the game. The point rather is this. We take the fundamental *true-of* relation which can obtain between an \mathcal{L} -sentence and an \mathcal{L} -structure: this immediately generates a certain Galois connection $F \dashv G$ between two naturally ordered ‘syntactic’ and ‘semantic’ posets, and this in turn already dictates that e.g. the composite maps GF and FG will have special significance as closure operations. So we come to see some familiar old logical ideas as exemplifying essentially general order-theoretic patterns which recur elsewhere. And that’s illuminating.

28 Adjoints introduced

NB: This chapter, and the next two, are taken, unrevised, from an earlier set of Notes on Category Theory. They continue the story without, I hope, too many jarring discontinuities. These chapters are less gentle than what's gone before and need a great deal of rewriting, not to mention checking for bad errors! However, if you have got this far then they should still be manageable and will hopefully be useful as a Rough Guide to adjunctions.

Recall that quotation from Tom Leinster which we gave at the very outset:

Category theory takes a bird's eye view of mathematics. From high in the sky, details become invisible, but we can spot patterns that were impossible to detect from ground level. (Leinster, 2014, p. 1)

Perhaps the most dramatic patterns that category theory newly reveals are those which involve *adjunctions*. As Mac Lane famously puts it (1997, p. vii) the slogan is “Adjoint functors arise everywhere.” In the last two chapters, we have seen a restricted version of the phenomenon (well known before category theory). But category theory enables us to generalize radically.

28.1 Adjoint functors: a first definition

(a) Let \mathcal{P} now be (not the poset itself but) the category corresponding to the poset (P, \leq) . So the objects of \mathcal{P} are the members of P , and there is a \mathcal{P} -arrow $p \rightarrow p'$ (for $p, p' \in P$), which we can identify with the pair $\langle p, p' \rangle$, if and only if $p \leq p'$. Similarly let \mathcal{Q} be the category corresponding to the poset (Q, \sqsubseteq) .

Now, changing symbolism just a little, a Galois connection between the posets (P, \leq) and (Q, \sqsubseteq) is a pair of functions $f: P \rightarrow Q$ and $g: Q \rightarrow P$ such that

- (i) f and g are monotone, and
- (ii) $f(p) \sqsubseteq q$ iff $p \leq g(q)$ for all $p \in P, q \in Q$.

(Well, we know condition (ii) implies condition (i), but it is helpful now to make it explicit.) However, monotone functions f, g between posets give rise to functors F, G between the corresponding categories – see §15.2, Ex. (F6). Thus the monotone function $f: P \rightarrow Q$ gives rise to the functor $F: \mathcal{P} \rightarrow \mathcal{Q}$ which

Adjoints introduced

sends the object p in \mathcal{P} to $f(p)$ in \mathcal{Q} , and sends an arrow $p \rightarrow p'$ in \mathcal{P} , i.e. the pair $\langle p, p' \rangle$, to the pair $\langle f(p), f(p') \rangle$ which is an arrow in \mathcal{Q} . Similarly, $g: \mathcal{Q} \rightarrow \mathcal{P}$ gives rise to a functor $G: \mathcal{Q} \rightarrow \mathcal{P}$.

So (ii) means that our adjoint *functions*, i.e. the Galois connection (f, g) between the posets (P, \leq) and (Q, \sqsubseteq) , gives rise to a pair of *functors* (F, G) between the poset categories \mathcal{P} and \mathcal{Q} , one in each direction, such that there is a (unique) arrow $Fp \rightarrow q$ in \mathcal{Q} iff there is a corresponding (unique) arrow $p \rightarrow Gq$ in \mathcal{P} . This sets up an isomorphism between the hom-sets $\mathcal{Q}(Fp, q)$ and $\mathcal{P}(p, Gq)$, for each $p \in \mathcal{P}, q \in \mathcal{Q}$.

Of course, for a particular choice of p, q , this will be a rather trivial isomorphism, as the homsets in this case are either both empty or both single-membered. But what isn't trivial is that the existence of the isomorphism arises *systematically* from the Galois connection, in a uniform and natural way. And we now know how to put that informal claim into more formal category-theoretic terms: we have a *natural isomorphism* here, i.e. $\mathcal{Q}(Fp, q) \cong \mathcal{P}(p, Gq)$ *naturally* in $p \in \mathcal{P}, q \in \mathcal{Q}$.

(b) Now we generalize this last idea in the obvious way, and also introduce some absolutely standard notation:

Definition 125. Suppose \mathcal{A} and \mathcal{B} are categories and $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ are functors. Then F is *left adjoint to* G and G is *right adjoint to* F , notated $F \dashv G$, iff

$$\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$$

naturally in $A \in \mathcal{A}, B \in \mathcal{B}$. We also write $\mathcal{A} \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} \mathcal{B}$ when this situation obtains, or $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$, and we say that F and G (together with the associated isomorphism between the relevant hom-sets) form an *adjunction*.

Here, and onwards through our discussions of adjunctions, we'll take it that there is no problem in talking about the relevant hom-sets (either because the categories are small enough, or because we are taking a relaxedly inclusive line on what counts as 'sets').

There is an additional fairly standard bit of notation to indicate the action of the natural isomorphism between the hom-sets in an adjunction:

Definition 126. Given the situation just described, and an arrow $f: F(A) \rightarrow B$, then one direction of the natural bijection between the hom-sets sends that arrow to its *transpose* $\bar{f}: A \rightarrow G(B)$; likewise the inverse bijection associates an arrow $g: A \rightarrow G(B)$ to its transpose $\bar{g}: F(A) \rightarrow B$.

(Another common notation distinguishes f^\flat for our \bar{f} and g^\sharp for our \bar{g} , and this notation might be preferable in principle since transposing by 'sharpening' and 'flattening' are indeed different operations. But the double use of the overlining notation is standard, and is slick.)

Evidently, transposing twice takes us back to where we started: $\overline{\overline{f}} = f$ and $\overline{\overline{g}} = g$.

28.2 Examples

As we'd expect from our discussion of Galois connections, given the existence of an adjoint connection $F \dashv G$ we can deduce a range of additional properties of the adjoint functors and of the operation of transposition. But before exploring this any further in the abstract, let's have some more examples of adjunctions (to add to those generated by Galois connections).

For a warm-up exercise, we start with a particularly easy case:

- (1) Consider any (non-empty!) category \mathcal{A} and the one object category $\mathbf{1}$ (comprising just the object \bullet and its identity arrow). There is a unique functor $F: \mathcal{A} \rightarrow \mathbf{1}$. Questions: when does F have a right adjoint $G: \mathbf{1} \rightarrow \mathcal{A}$? what about a left adjoint?

If G is to be a right adjoint, remembering that $FA = \bullet$ for any $A \in \mathcal{A}$, we require

$$\mathbf{1}(\bullet, \bullet) \cong \mathcal{A}(A, G\bullet),$$

for any A . The hom-set on the left contains just the identity arrow. So that can only be in bijection to the hom-set on the right, for each A , if there is always a *unique* arrow $A \rightarrow G\bullet$, i.e. if $G\bullet$ is terminal in \mathcal{A} .

In sum, F has a right adjoint $G: \mathbf{1} \rightarrow \mathcal{A}$ just in case G sends $\mathbf{1}$'s unique object to \mathcal{A} 's terminal object: no terminal object, no right adjoint.

Dually, F has a left adjoint if and only if \mathcal{A} has an initial object.

This toy example reminds of what we have already seen in the special case of Galois connections, namely that a functor may or may not have a right adjoint, and independently may or may not have a left adjoint, and if both adjoints exist they may be different. But let's also note that we have here a first indication that adjunctions and limits can interact in interesting way: in this case, indeed, we could *define* terminal and initial objects for a category \mathcal{A} in terms of the existence of right and left adjoints to the functor $F: \mathcal{A} \rightarrow \mathbf{1}$. We will return to this theme.

Now for a couple of more substantive examples. And to speed things along, we will proceed informally: we won't in this section actually prove that the relevant hom-sets in our various examples are naturally isomorphic in the official formal sense, but rather we will take it as enough to find a bijection which can be evidently set up in a systematic and intuitively natural way, without arbitrary choices.

- (2) Let's next consider the forgetful functor $U: \mathbf{Top} \rightarrow \mathbf{Set}$ which sends each topological space to its underlying set of points, and sends any continuous

function between topological spaces to the same function thought of as a set-function. Questions: does this have a left adjoint? a right adjoint?

If U is to have a left adjoint $F: \mathbf{Set} \rightarrow \mathbf{Top}$, then for any set S and for any topological space (T, O) – with T a set of points and O a topology (a suitable collection of open sets) – we require

$$\mathbf{Top}(F(S), (T, O)) \cong \mathbf{Set}(S, U(T, O)) = \mathbf{Set}(S, T),$$

where the bijection here needs to be a natural one.

Now, on the right we have the set of *all* functions $f: S \rightarrow T$. So that needs to be in bijection with the set of all *continuous* functions from FS to (T, O) . How can we ensure this holds in a systematic way, for any S and (T, O) ? Well, suppose that for any S , F sends S to the topological space (S, D) which has the discrete topology (i.e. all subsets of S count as open). It is a simple exercise to show that *every* function $f: S \rightarrow T$ then counts as a continuous function $f: (S, D) \rightarrow (T, O)$. So the functor F which assigns a set the discrete topology will indeed be left adjoint to the forgetful functor.

Similarly, the functor $G: \mathbf{Set} \rightarrow \mathbf{Top}$ which assigns a set the indiscrete topology (the only open sets are the empty set and S itself) is right adjoint to the forgetful functor U .

- (3) Let's now take another case of a forgetful functor, this time the functor $U: \mathbf{Mon} \rightarrow \mathbf{Set}$ which forgets about monoidal structure. Does U have a left adjoint $F: \mathbf{Set} \rightarrow \mathbf{Mon}$. If (M, \cdot) is a monoid and S some set, we need

$$\mathbf{Mon}(FS, (M, \cdot)) \cong \mathbf{Set}(S, U(M, \cdot)) = \mathbf{Set}(S, M).$$

The hom-set on the right contains all possible functions $f: S \rightarrow M$. How can these be in one-one correspondence with the monoid homomorphisms from FS to (M, \cdot) ?

Arm-waving for a moment, suppose FS is some monoid with a lot of structure (over and above the minimum required to be a monoid). Then there may be few if any monoid homomorphisms from FS to (M, \cdot) . Therefore, if there are potentially to be *lots* of such monoid homomorphisms, one for each $f: S \rightarrow M$, then FS will surely need to have minimal structure. Which suggests going for broke and considering the limiting case, i.e. the functor F which sends a set S to $(S^*, *)$, the *free* monoid on S which we met back in §15.5, Ex. (F13). Recall, the objects of $(S^*, *)$ are sequences of S -elements (including the null sequence) and its monoid operation is concatenation.

There is an obvious map α which takes an arrow $f: S \rightarrow M$ and sends it to $\bar{f}: (S^*, *) \rightarrow (M, \cdot)$, where \bar{f} sends the empty sequence of S -elements to the unit of M , and sends the finite sequence $x_1 * x_2 * x_3 * \dots * x_n$ to the M -element $f x_1 \cdot f x_2 \cdot f x_3 \cdot \dots \cdot f x_n$. So defined, \bar{f} respects the unit and the monoid operation and so is a monoid homomorphism.

There is an equally obvious map β which takes an arrow $g: (S^*, *) \rightarrow (M, \cdot)$ to the function $\bar{g}: S \rightarrow M$ which sends an element $x \in S$ to $g\langle x \rangle$ (i.e. to g applied to the one-element list containing x).

Evidently α and β are inverses, so form a bijection, and their construction is quite general (i.e. can be applied to any set S and monoid (M, \cdot)). Which establishes that, as required $\mathbf{Mon}(FS, (M, \cdot)) \cong \mathbf{Set}(S, M)$.

So in sum, the free functor F which takes a set to the free monoid on that set is left adjoint to the forgetful functor U which sends a monoid to its underlying set.

Now recall Theorem 151: if a function f has a left or right adjoint to make up a Galois connection, then that adjoint is unique. An analogous uniqueness result applies to adjoints more generally: if a functor has a left adjoint, then it is unique up to isomorphism, and likewise right adjoints (when they exist) are unique up to isomorphism. So we can say that the functor which assigns a set the indiscrete topology is in fact *the* right adjoint to the functor which forgets topological structure, and we can say that the functor sending a set to the free monoid on that set is *the* left adjoint of the forgetful functor on monoids. However, the uniqueness theorem for adjoints takes a bit of work; so we'll delay the proof until the next chapter, §29.4. For the moment, then, we'll officially continue simply to talk of one functor being left (right) adjoint to another without making explicit uniqueness claims.

Our example involving monoids is actually typical of a whole cluster of cases. A left adjoint of the trivial forgetful functor from some class of algebraic structures to their underlying sets is characteristically provided by the non-trivial functor that takes us from a set to a free structure of that algebraic kind. Thus we have, for example,

- (4) The forgetful functor $U: \mathbf{Grp} \rightarrow \mathbf{Set}$ has as a left adjoint the functor $F: \mathbf{Set} \rightarrow \mathbf{Grp}$ which sends a set to the free group on that set (i.e. the group obtained from a set S by adding just enough elements for it to become a group while imposing no constraints other than those required to ensure we indeed have a group).

What about *right* adjoints to our last two forgetful functors?

- (5) We will later show that the forgetful functor $U: \mathbf{Mon} \rightarrow \mathbf{Set}$ has no right adjoint by a neat proof in §30.3. But here's a more arm-waving argument. U would have a right adjoint $G: \mathbf{Set} \rightarrow \mathbf{Mon}$ just in case $\mathbf{Set}(M, S) = \mathbf{Set}(U(M, \cdot), S) \cong \mathbf{Mon}((M, \cdot), GS)$, for all monoids (M, \cdot) and sets S . But this requires the monoid homomorphisms from (M, \cdot) to GS always to be in bijection with the set-functions from M to S . But that's not possible (consider keeping the sets M and S fixed, but changing the possible monoid operations with which M is equipped).

Similarly the forgetful functor $U: \mathbf{Grp} \rightarrow \mathbf{Set}$ has no right adjoint.

- (6) There are however examples of ‘less forgetful’ algebraic functors which have both left and right adjoints. Take the functor $U: \mathbf{Grp} \rightarrow \mathbf{Mon}$ which forgets about group inverses but keeps the monoidal structure. This has a left adjoint $F: \mathbf{Mon} \rightarrow \mathbf{Grp}$ which converts a monoid to a group by adding inverses for elements (and otherwise making no more assumptions that are needed to get a group). U also has a right adjoint $G: \mathbf{Mon} \rightarrow \mathbf{Grp}$ which rather than adding elements subtracts them by mapping a monoid to the submonoid of its invertible elements (which can be interpreted as a group).

Let’s quickly check just the second of those claims. We have $U \dashv G$ so long as

$$\mathbf{Mon}(U(K, \times), (M, \cdot)) \cong \mathbf{Grp}((K, \times), G(M, \cdot)),$$

for any monoid (M, \cdot) and group (K, \times) . Now we just remark that every element of (K, \times) -as-a-monoid is invertible and a monoid homomorphism sends invertible elements to invertible elements. Hence a monoid homomorphism from (K, \times) -as-a-monoid to (M, \cdot) will in fact also be a group homomorphism from (K, \times) to the submonoid-as-a-group $G(M, \cdot)$.

- (7) Recall the functor $F: \mathbf{Set} \rightarrow \mathbf{Rel}$ which ‘forgets’ that arrows are functional (see §15.2, Ex. (F2)). And now we introduce a powerset functor $P: \mathbf{Rel} \rightarrow \mathbf{Set}$ defined as follows:
- a) P sends a set A to its powerset $\mathcal{P}(A)$, and
 - b) P sends a relation R in $A \times B$ to the function $f_R: \mathcal{P}(A) \rightarrow \mathcal{P}(B)$ which sends $X \subseteq A$ to $Y = \{b \mid (\exists x \in X) Rxb\} \subseteq B$.

Claim: $F \dashv P$.

We observe that there is a (natural!) isomorphism which correlates a relation R in $A \times B$ with a function $f: A \rightarrow \mathcal{P}(B)$ where $f(x) = \{y \mid Rxy\}$ and so Rxy iff $y \in f(x)$. This gives us a isomorphism $\mathbf{Rel}(FA, B) \cong \mathbf{Set}(A, PB)$ which can be checked to be natural in $A \in \mathbf{Set}$ and $B \in \mathbf{Rel}$.

And now for some cases not involving forgetful functors:

- (8) Suppose \mathcal{C} is a category with exponentiation (and hence with products). Then, in a slogan, exponentiation by B is right adjoint to taking the product with B .

To see this, we define a pair of functors from \mathcal{C} to itself. First, there is the functor $- \times B: \mathcal{C} \rightarrow \mathcal{C}$ which sends an object A to the product $A \times B$, and sends an arrow $f: A \rightarrow A'$ to $f \times 1_B: A \times B \rightarrow A' \times B$.

Second there is the functor $(-)^B: \mathcal{C} \rightarrow \mathcal{C}$ which sends an object C to C^B , and sends an arrow $f: C \rightarrow C'$ to $f \circ ev: C^B \rightarrow C'^B$ as defined in the proof that (F15) is functor in §15.6. It is easily checked that $(-)^B$ satisfies the conditions for functoriality.

By the theorem just mentioned, $\mathcal{C}(A \times B, C) \cong \mathcal{C}(A, C^B)$ naturally in A and C . Hence $(- \times B) \dashv (-)^B$.

- (9) Recall Defn. 14 which defined the product of two categories. Given a category \mathcal{C} there is a trivial diagonal functor $\Delta: \mathcal{C} \rightarrow \mathcal{C} \times \mathcal{C}$ which sends a \mathcal{C} -object A to the pair $\langle A, A \rangle$, and sends a \mathcal{C} -arrow f to the pair of arrows $\langle f, f \rangle$. What would it take for this functor to have a right adjoint $G: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$? We'd need

$$(\mathcal{C} \times \mathcal{C})(\langle A, A \rangle, \langle B, C \rangle) \cong \mathcal{C}(A, G\langle B, C \rangle)$$

naturally in $A \in \mathcal{C}$ and in $\langle B, C \rangle \in \mathcal{C} \times \mathcal{C}$. But by definition the left hand hom-set is $\mathcal{C}(A, B) \times \mathcal{C}(A, C)$. But then if we can take G to be the product functor that sends $\langle B, C \rangle$ to the product object $B \times C$ in \mathcal{C} we'll get an obvious natural isomorphism

$$\mathcal{C}(A, B) \times \mathcal{C}(A, C) \cong \mathcal{C}(A, B \times C).$$

So in sum, $\Delta: \mathcal{C} \rightarrow \mathcal{C} \times \mathcal{C}$ has a right adjoint if \mathcal{C} has binary products.

- (10) For topologists, let's simply mention another example of a case where the adjoint of a trivial functor is something much more substantial. The inclusion functor from **KHaus**, the category of compact Hausdorff spaces, into **Top** has a left adjoint, namely the Stone-Ćech compactification functor.

28.3 Naturality

We said: $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$ just in case

$$\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$$

holds naturally in $A \in \mathcal{A}$, $B \in \mathcal{B}$. Let's now be more explicit about what the official naturality requirement comes to.

By Defn. 100, the required bijection holds naturally in B (to take that case first) just if the two hom-functors $\mathcal{B}(F(A), -)$ and $\mathcal{A}(A, G(-))$ are naturally isomorphic. By Defn. 99, that means there have to be isomorphisms $\varphi_B: \mathcal{B}(F(A), B) \rightarrow \mathcal{A}(A, G(B))$, one for each B , such that for every $h: B \rightarrow B'$, the usual naturality square always commutes:

$$\begin{array}{ccc} \mathcal{B}(F(A), B) & \xrightarrow{\mathcal{B}(F(A), h)} & \mathcal{B}(F(A), B') \\ \downarrow \varphi_B & & \downarrow \varphi_{B'} \\ \mathcal{A}(A, G(B)) & \xrightarrow{\mathcal{A}(A, G(h))} & \mathcal{A}(A, G(B')) \end{array}$$

But how does the covariant hom-functor $\mathcal{B}(F(A), -)$ operate on $h: B \rightarrow B'$? As we saw in §18.2, it sends h to $h \circ -$, i.e. to that function which composes h with an arrow from $\mathcal{B}(F(A), B)$ to give an arrow in $\mathcal{B}(F(A), B')$. Similarly, $\mathcal{A}(A, G(-))$ will send h to $Gh \circ -$.

So consider an arrow $f: F(A) \rightarrow B$ living in $\mathcal{B}(F(A), B)$. The naturality square now tells us that for any $h: B \rightarrow B'$, $\varphi_{B'}(h \circ f) = Gh \circ \varphi_B(f)$.

But (by the definition of transposition!), the components of φ send an arrow to its transpose. So we have shown the first part of the following theorem. And the second part of this theorem follows by a dual argument, in which some arrows get reversed because the relevant hom-functors in this case are contravariant.

Theorem 159. *Given $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$, then*

- (1) *for any $f: F(A) \rightarrow B$ and $h: B \rightarrow B'$, $\overline{h \circ f} = Gh \circ \overline{f}$,*
- (2) *for any $g: A \rightarrow G(B)$ and $k: A' \rightarrow A$, $\overline{g \circ k} = \overline{g} \circ Fk$, i.e. $\overline{\overline{g} \circ Fk} = g \circ k$.*

Inspecting the proof, we see that there is an obvious converse to this theorem. Given functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ such that there is always a bijection between $\mathcal{B}(F(A), B)$ and $\mathcal{A}(A, G(B))$ then, if conditions (1) and (2) hold, the bijections (for different A s and B s) will assemble into natural transformations, so that $\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$ holds naturally in $A \in \mathcal{A}$, $B \in \mathcal{B}$, and hence $F \dashv G$.

28.4 An alternative definition

We now know what it takes for a pair of functors to be adjoint to each other, and we have given various examples of adjoint pairs (to add to the special cases from the previous two chapters where the adjunctions are Galois connections).

Now, our first definition of adjunctions was inspired by our original definition of Galois connections in §27.3. But we gave an alternative definition of such connections in §27.4. This too can be generalized to give a second definition of adjunctions. In this section we show how, and prove that the new definition is equivalent to our first one. (This alternative definition will turn out to look somewhat more complicated, but it is useful in practice – though for the moment our prime aim is to bring out something of the structural richness of adjunctions.)

A Galois connection between the posets (P, \leq) , (Q, \sqsubseteq) , according to the alternative definition, comprises a pair of functions $f: P \rightarrow Q$ and $g: Q \rightarrow P$ such that

- (i) f and g are monotone,
- (ii) $p \leq g(f(p))$ for all $p \in P$, and
- (iii) $f(g(q)) \sqsubseteq q$ for all $q \in Q$.

Since the composition of monotone functions is monotone, (ii) and (iii) are in fact easily seen to be equivalent to

- (ii') if $p \leq p'$, then $p \leq p' \leq g(f(p'))$ and $p \leq g(f(p)) \leq g(f(p'))$,
- (iii') if $q \sqsubseteq q'$, then $f(g(q)) \sqsubseteq q \sqsubseteq q'$ and $f(g(q)) \sqsubseteq f(g(q')) \sqsubseteq q'$.

As before, let \mathcal{P} be the category corresponding to the poset (P, \leq) , and recall that there is an arrow $p \rightarrow p'$ in \mathcal{P} just when $p \leq p'$ in the poset (P, \leq) . Likewise for \mathcal{Q} corresponding to (Q, \sqsubseteq) . And again as before, note that the monotone functions f, g between the posets give rise to functors F, G between the corresponding categories. Hence, in particular, the composite monotone function $g \circ f$ gives rise to a functor $G \circ F: \mathcal{P} \rightarrow \mathcal{P}$, and likewise $f \circ g$ gives rise to a functor $F \circ G: \mathcal{Q} \rightarrow \mathcal{Q}$.

Now, (ii') corresponds in \mathcal{P} to the claim that the following diagram always commutes:

$$\begin{array}{ccc} p & \longrightarrow & p' \\ \downarrow & & \downarrow \\ (G \circ F)p & \longrightarrow & (G \circ F)p' \end{array}$$

(We needn't label the arrows as in the poset category \mathcal{P} arrows between objects are unique when they exist.)

Dropping the explicit sign for composition of functors for brevity's sake, let's define $\eta_p: 1_{\mathcal{P}} \Rightarrow GF$ to be the arrows $p \rightarrow GFp$, one for each $p \in \mathcal{P}$. Then our commutative diagram version of (ii') can be revealingly redrawn as follows:

$$\begin{array}{ccc} 1_{\mathcal{P}} p & \longrightarrow & 1_{\mathcal{P}} p' \\ \downarrow \eta_p & & \downarrow \eta_{p'} \\ GFp & \longrightarrow & GFp' \end{array}$$

This commutes for all p, p' . So applying Defn. 101, this is just to say that the η_p assemble into a natural transformation $\eta: 1_{\mathcal{P}} \Rightarrow GF$ in \mathcal{P} .

Likewise, (iii) and hence (iii') correspond to the claim that there is a natural transformation $\varepsilon: FG \Rightarrow 1_{\mathcal{Q}}$ in \mathcal{Q} .

(a) So far so good. We have here the initial ingredients for an alternative definition for an adjunction between functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$: we will require there to be a pair of natural transformations $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$.

However, as we'll see, this isn't yet quite enough. But the additional ingredients we want are again suggested by our earlier treatment of Galois connections. Recall from Theorem 149 that if (f, g) is a Galois connection between (P, \leq) and (Q, \sqsubseteq) , then we immediately have the key identities

- (iv) $f \circ g \circ f = f$, and
- (v) $g \circ f \circ g = g$.

By (iv), $fp \leq (f \circ g \circ f)p \leq fp$, for p in (P, \leq) . Hence in \mathcal{P} the following diagram commutes for each p :

$$\begin{array}{ccc}
 Fp & \longrightarrow & FGFp \\
 & \searrow & \downarrow \\
 & & Fp
 \end{array}$$

Here, the diagonal arrow is the identity 1_{Fp} . The downward arrow is ε_{Fp} (the component of ε at Fp). And the horizontal arrow is $F\eta_p$. So we have $\varepsilon_{Fp} \circ F\eta_p = 1_{Fp}$ for each p .

Or what comes to the same, in the functor category $[\mathcal{P}, \mathcal{Q}]$ this diagram commutes¹

$$\begin{array}{ccc}
 F & \xrightarrow{F\eta} & FGF \\
 & \searrow 1_F & \downarrow \varepsilon F \\
 & & F
 \end{array}$$

For remember whiskering(!), discussed in §21.3: the components $F(\eta_p)$ assemble into the natural transformation we symbolized ' $F\eta$ ', and the components ε_{Fp} assemble into the natural transformation we symbolized ' εF '. And then recall from §22.1 that 'vertical' composition of natural transformations between e.g. the functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $FGF: \mathcal{A} \rightarrow \mathcal{B}$ is defined component-wise. So, for each p ,

$$(\varepsilon F \circ F\eta)_p = \varepsilon_{Fp} \circ F\eta_p = 1_{Fp} = (1_F)_p,$$

where 1_F is the natural transformation whose component at p is 1_{Fp} . Since all components are equal, the left-most and right-most natural transformations in that equation are equal and our diagram commutes.

Exactly similarly, from (v) we infer that $G\varepsilon_q \circ \eta_{Gq} = 1_{Gq}$. In other words, the next diagram commutes in $[\mathcal{Q}, \mathcal{P}]$:

$$\begin{array}{ccc}
 G & \xrightarrow{\eta G} & GFG \\
 & \searrow 1_G & \downarrow G\varepsilon \\
 & & G
 \end{array}$$

(b) And *now* we can put everything together to give us our second definition for adjoint functors:

¹Notational fine print: our convention has been to use single arrows to represent arrows inside particular categories, and double arrows to represent natural transformations between functors across categories. We are now dealing with natural-transformations-thought-of-as-arrows-within-a-particular-functor-category. Some use double arrows for diagrams in a functor category, to remind us these are natural transformations (between functors relating some other categories); some use single arrows because these are being treated as arrows (in the functor category). I'm jumping the second way, following the majority and also getting slightly cleaner diagrams.

Definition 127 (Alternative). Suppose \mathcal{A} and \mathcal{B} are categories and $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ are functors. Then F is left adjoint to G and G is right adjoint to F , notated $F \dashv G$, iff

- (i) there are natural transformations $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$ such that
- (ii) $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$, and $G\varepsilon_B \circ \eta_{GB} = 1_{GB}$ for all $B \in \mathcal{B}$; or equivalently
- (ii') the following *triangle identities* hold in the functor categories $[\mathcal{A}, \mathcal{B}]$ and $[\mathcal{B}, \mathcal{A}]$ respectively:

$$\begin{array}{ccc}
 F & \xrightarrow{F\eta} & FGF \\
 & \searrow 1_F & \downarrow \varepsilon F \\
 & & F
 \end{array}
 \qquad
 \begin{array}{ccc}
 G & \xrightarrow{\eta G} & GFG \\
 & \searrow 1_G & \downarrow G\varepsilon \\
 & & G
 \end{array}$$

Note, η and ε are standardly called the *unit* and *counit* of the adjunction.

It remains to show that Defn. 125 and Defn. 127 are equivalent:

Theorem 160. For given functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$, $F \dashv G$ holds by our original definition iff it holds by the alternative definition.

Proof (If). Suppose there are natural transformations $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$ for which the triangle identities hold.

Take any f in $\mathcal{B}(F(A), B)$. Then $\eta_A: A \rightarrow GF(A)$ and $G(f): GF(A) \rightarrow GB$ compose. And so we can define $\varphi_{AB}: \mathcal{B}(F(A), B) \rightarrow \mathcal{A}(A, G(B))$ by putting $\varphi_{AB}(f) = G(f) \circ \eta_A$.

Likewise, we can define $\psi_{AB}: \mathcal{A}(A, G(B)) \rightarrow \mathcal{B}(F(A), B)$ by putting $\psi_{AB}(g) = \varepsilon_B \circ F(g)$ for any $g: A \rightarrow G(B)$.

Keep A fixed: then, as we vary B , the various components φ_{AB} assemble into a natural transformation $\varphi_A: \mathcal{B}(F(A), -) \Rightarrow \mathcal{A}(A, G(-))$. That's because the naturality square

$$\begin{array}{ccc}
 \mathcal{B}(F(A), B) & \xrightarrow{h \circ -} & \mathcal{B}(F(A), B') \\
 \downarrow \varphi_{AB} & & \downarrow \varphi_{AB'} \\
 \mathcal{A}(A, G(B)) & \xrightarrow{Gh \circ -} & \mathcal{A}(A, G(B'))
 \end{array}$$

commutes for every $h: B \rightarrow B'$, i.e. for every f in $\mathcal{B}(F(A), B)$ we have

$$\varphi_{AB'}(h \circ f) = G(h \circ f) \circ \eta_A = Gh \circ (Gf \circ \eta_A) = Gh \circ \varphi_{AB}(f)$$

which holds by the functoriality of G .

Now keep B fixed: then by a parallel argument, as we vary A , the various components φ_{AB} assemble into a natural transformation $\varphi_B: \mathcal{B}(F(-), B) \Rightarrow \mathcal{A}(-, G(B))$ between the two *contravariant* functors.

Adjoint introduced

Similarly if we keep A fixed, the various ψ_{AB} assemble into a natural transformation $\psi_A: \mathcal{A}(A, G(-)) \Rightarrow \mathcal{B}(F(A), -)$; and if we keep B fixed, the various ψ_{AB} assemble into $\psi_B: \mathcal{A}(-, G(B)) \Rightarrow \mathcal{B}(F(-), B)$.

We now need to show that these natural transformations are isomorphisms, from which the desired result will follow: i.e. $\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$ naturally in $A \in \mathcal{A}$ and in $B \in \mathcal{B}$.

We show each φ_{AB} and ψ_{AB} are mutually inverse. Take any $f: FA \rightarrow B$. Then

$$\begin{aligned}
 \psi_{AB}(\varphi_{AB}(f)) &= \psi_{AB}(G(f) \circ \eta_A) && \text{by definition of } \varphi \\
 &= \varepsilon_B \circ F(G(f) \circ \eta_A) && \text{by definition of } \psi \\
 &= \varepsilon_B \circ FGf \circ F\eta_A && \text{by functoriality of } F \\
 &= f \circ \varepsilon_{FA} \circ F\eta_A && \text{by naturality square for } \varepsilon \\
 &= f \circ 1_{FA} && \text{by triangle equality} \\
 &= f
 \end{aligned}$$

Hence $\psi_{AB} \circ \varphi_{AB} = 1$ (note how we did need to appeal to the added triangle equality, not just functoriality and the naturality of ε). Likewise $\varphi_{AB} \circ \psi_{AB} = 1$. \square

Proof (Only if). Suppose $\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$ naturally in $A \in \mathcal{A}$ and in $B \in \mathcal{B}$. We need to define a unit and counit for the adjunction, and show they satisfy the triangle equalities.

Take the identity arrow 1_{FA} in $\mathcal{B}(FA, FA)$. The natural isomorphism defining the adjunction sends 1_{FA} to a arrow we will hopefully call $\eta_A: A \rightarrow GF(A)$.

We first show that the components η_A do indeed assemble into a natural transformation from $1_{\mathcal{A}}$ to GF . So consider the following two diagrams:

$$\begin{array}{ccc}
 FA & \xrightarrow{Ff} & FA' \\
 \downarrow 1_{FA} & & \downarrow 1_{FA'} \\
 FA & \xrightarrow{Ff} & FA'
 \end{array}
 \qquad
 \begin{array}{ccc}
 A & \xrightarrow{f} & A' \\
 \downarrow \eta_A & & \downarrow \eta_{A'} \\
 GFA & \xrightarrow{GFf} & GFA'
 \end{array}$$

Trivially, the diagram on the left commutes for all $f: A \rightarrow A'$. That is to say, $\overline{Ff \circ 1_{FA}} = \overline{1_{FA'} \circ Ff}$. Transposition must evidently preserve identities. So $\overline{Ff \circ 1_{FA}} = \overline{1_{FA'} \circ Ff}$. But by the first of the naturality requirements in §28.3, $\overline{Ff \circ 1_{FA}} = \overline{GFf \circ 1_{FA}} = GFf \circ \eta_A$. And by the other naturality requirement, $\overline{1_{FA'} \circ Ff} = \overline{\eta_{A'} \circ f} = \eta_{A'} \circ f$. So we have $GFf \circ \eta_A = \eta_{A'} \circ f$ and the diagram on the right commutes for all f . Hence the components η_A do indeed assemble into a natural transformation.

Similarly the same natural isomorphism in the opposite direction sends 1_{GB} to its transpose $\varepsilon_B: FG(B) \rightarrow B$, and the components ε_B assemble into a natural transformation from FG to $1_{\mathcal{B}}$.

Now consider these two diagrams:

$$\begin{array}{ccc}
 A & \xrightarrow{\eta_A} & GFA \\
 \downarrow \eta_A & & \downarrow 1_{GFA} \\
 GFA & \xrightarrow{1_{GFA}} & GFA
 \end{array}
 \qquad
 \begin{array}{ccc}
 FA & \xrightarrow{F\eta_A} & FGFA \\
 \downarrow 1_{FA} & & \downarrow \varepsilon_{FA} \\
 FA & \xrightarrow{1_{FA}} & FA
 \end{array}$$

The diagram on the left trivially commutes. Transpose it via the natural isomorphism that defines the adjunction and use the naturality requirements again; we find that the diagram on the right must also commute. So $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$ – which gives us one of the triangle identities. The other identity we get dually. \square

We are done. But although the strategies for proving the equivalence of our definitions are entirely straightforward, checking the details was a bit tedious and required keeping our wits about us. So let's pause before resuming in the next chapter the exploration of adjunctions.

28.5 Adjoints and equivalent categories

Our second definition of an adjunction should remind you strongly of our earlier characterization of what it takes for categories to be equivalent. We should pause to say something about this.

We can slightly recast our definitions to highlight the parallelism:

Definition 109* An equivalence between categories \mathcal{A} and \mathcal{B} is a pair of functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ and a pair of natural *isomorphisms* $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$.

Definition 127* An adjunction between categories \mathcal{A} and \mathcal{B} is a pair of functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ and a pair of natural *transformations* $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$ such that $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$, and $G\varepsilon_B \circ \eta_{GB} = 1_{GB}$ for all $B \in \mathcal{B}$.

Since transformations need not be isomorphisms, an adjunction needn't be an equivalence (and indeed we have met lots of examples of adjunctions between non-equivalent categories). In the other direction, an isomorphism needn't satisfy the triangle identities, so an equivalence needn't be an adjunction either. However, we *do* have the following result:

Theorem 161. *If there is an equivalence between \mathcal{A} and \mathcal{B} constituted by a pair of functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ and a pair of natural isomorphisms $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\gamma: FG \Rightarrow 1_{\mathcal{B}}$, then there is an adjunction $F \dashv G$ with unit η and counit ε (defined in terms of γ and η), and further there is also an adjunction $G \dashv F$.*

In other words, take an equivalence, fix one of the natural transformations, but tinker (if necessary) with the other, and we get an adjunction. Further we can construct an adjunction in the opposite direction.

Adjoint introduced

Proof. Define the natural transformation ε by composition as follows:

$$\varepsilon: FG \xrightarrow{FG\gamma^{-1}} FGF G \xrightarrow{(F\eta G)^{-1}} FG \xrightarrow{\gamma} 1_{\mathcal{B}}$$

Since η and γ are isomorphisms, and by Theorem 107 whiskering natural isomorphisms yields another natural isomorphism, the inverses mentioned here must exist.

So we just need to establish that, with ε so defined, we get the usual triangle identities $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$, and $G\varepsilon_B \circ \eta_{GB} = 1_{GB}$ for all $B \in \mathcal{B}$.

So, firstly, for any A , we need the composite arrow (*)

$$FA \xrightarrow{F\eta_A} FGFA \xrightarrow{(FG\gamma^{-1})_{FA}} FGF GFA \xrightarrow{(F\eta G)^{-1}_{FA}} FGFA \xrightarrow{\gamma_{FA}} FA$$

to equal the identity arrow on FA (recall, the component of a ‘vertical’ composite of natural transformations for FA is the composite of the components of the individual transformations).

We begin by noting that, for any $A \in \mathcal{A}$, the following square commutes by the naturality of η :

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & GFA \\ \downarrow \eta_A & & \downarrow \eta_{GFA} \\ GFA & \xrightarrow{GF\eta_A} & GF GFA \end{array}$$

So we have $\eta_{GFA} \circ \eta_A = GF\eta_A \circ \eta_A$. But since η_A is an isomorphism, it is epic (right-cancellable), so we have $\eta_{GFA} = GF\eta_A$ for all A . Similarly, we have $\gamma_{FG B}^{-1} = (FG\gamma^{-1})_B$ for all $B \in \mathcal{B}$.

So now consider the following diagram:

$$\begin{array}{ccc} FA & \xrightarrow{F\eta_A} & FGFA \\ \downarrow (\gamma^{-1})_{FA} & & \downarrow (\gamma^{-1})_{FGFA} = (FG\gamma^{-1})_{FA} \\ FGFA & \xrightarrow{FGF\eta_A} & FGF GFA \\ \downarrow 1_{FGFA} & \swarrow (F\eta G)^{-1}_{FA} & \\ FGFA & & \\ \downarrow \gamma_{FA} & & \\ FA & & \end{array}$$

The top square commutes, being a standard naturality square. (Fill in the schema of Defn. 101 by putting the natural transformation $\alpha = \gamma^{-1}: 1_{\mathcal{B}} \rightarrow FG$, and put f to be the function $F\eta_A: FA \rightarrow FB$.) And the triangle below commutes

28.5 Adjoints and equivalent categories

because $FGF\eta_A = F\eta_{GFA}$ from the equation above and $F\eta_{GFA} = (F\eta G)_{FA}$ (since $\eta_{GFA} = (\eta G)_{FA}$), so the arrows along two sides are simply inverses, and therefore compose to the identity.

The whole diagram therefore commutes. The arrows on longer circuit from top-left to bottom form the composite (*). The arrows on the direct route from top to bottom compose to the identity 1_{FA} . The composites are equal and hence we have established that the first triangle identity holds.

The second triangle identity holds by a similar argument.

Hence $F \dashv G$. And finally we note that if we put $\eta' = \gamma^{-1}$ and $\gamma' = \eta^{-1}$, and put $F' = G$, $G' = F$, the same line of proof shows that $F' \dashv G'$, and so $G \dashv F$. \square

29 Adjoints further explored

NB: This chapter, like the previous one, is taken, unrevised, from an earlier set of Notes on Category Theory. It needs a great deal of rewriting, not to mention checking for bad errors! However, if you have got this far then it should still be useful.

We have given a pair of definitions of adjoint functors, mirroring the two alternative definitions of Galois connections. We showed the definitions to be equivalent, and met some initial examples of adjunctions.

In this chapter, after a couple of preliminary sections, we continue to generalize some of the most basic results we found for Galois connections to adjunctions more generally.

29.1 Adjunctions reviewed

Let's gather together what we know about adjunctions so far.

Suppose $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ are functors. Then F is left-adjoint to G (equivalently, G is right-adjoint to F), in symbols $F \dashv G$, or more fully $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$, iff the following conditions all hold together:

- (1) $\mathcal{B}(FA, B) \cong \mathcal{A}(A, GB)$ naturally in $A \in \mathcal{A}$, $B \in \mathcal{B}$ – the isomorphism in each direction is said to send an arrow f in one hom-set to its transpose \bar{f} in the other.
- (2) There are natural transformations $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$ such that $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$, and $G\varepsilon_B \circ \eta_{GB} = 1_{GB}$ for all $B \in \mathcal{B}$. η is said to be the unit, ε the counit of the adjunction.
- (3) The component $\eta_A: A \rightarrow GFA$ of the natural transformation η can be identified as the transpose of $1_{FA}: FA \rightarrow FA$ under the natural isomorphism between $\mathcal{B}(FA, FA)$ and $\mathcal{A}(A, GFA)$. Likewise, the component ε_B is the transpose of 1_{GB} under the natural isomorphism between $\mathcal{A}(GB, GB)$ and $\mathcal{B}(FGB, B)$.
- (4) The inverse isomorphisms from $\mathcal{B}(FA, B)$ to $\mathcal{A}(A, GB)$ and back can be identified as $G(-) \circ \eta_A: \mathcal{B}(FA, B) \xrightarrow{\sim} \mathcal{A}(A, GB)$ and $\varepsilon_B \circ F(-): \mathcal{A}(A, GB) \xrightarrow{\sim} \mathcal{B}(FA, B)$.

- (5) For any $f: FA \rightarrow B$ and $h: B \rightarrow B'$, $\overline{h \circ f} = Gh \circ \overline{f}$; and for any $g: A \rightarrow GB$ and $k: A' \rightarrow A$, $\overline{g \circ k} = \overline{g} \circ Fk$, i.e. $\overline{g} \circ Fk = g \circ k$.

These conditions are not independent, however: (1) and (2) are equivalent, and both then imply (3) to (5).

29.2 Two more theorems!

Using (2) and (4) in our conditions on adjunctions, it follows that if $F \dashv G$, then there is a natural transformation $\eta: 1_{\mathcal{A}} \Rightarrow GF$ which has the following ‘universal mapping property’: for any $g: A \rightarrow G(B)$ there is a unique associated $f: F(A) \rightarrow B$ such that $g = G(f) \circ \eta_A$.

It is worth noting that we can also prove the converse here, so we get a biconditional:

Theorem 162. *Given functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$, then $F \dashv G$ iff (i) there is a natural transformation $\eta: 1_{\mathcal{A}} \Rightarrow GF$, for which (ii) for any $g: A \rightarrow G(B)$ in \mathcal{A} there is a unique $f: F(A) \rightarrow B$ in \mathcal{B} such that $g = G(f) \circ \eta_A$.*

Proof for ‘if’. First use clause (i) and define $\varphi_{AB}: \mathcal{B}(F(A), B) \rightarrow \mathcal{A}(A, G(B))$ by putting $\varphi_{AB}(f) = G(f) \circ \eta_A$.

By same proof as for Theorem 160, when we keep A fixed the various components φ_{AB} assemble into a natural transformation $\varphi_A: \mathcal{B}(F(A), -) \Rightarrow \mathcal{A}(A, G(-))$. And when we keep B fixed, the various components φ_{AB} assemble into a natural transformation $\varphi_B: \mathcal{B}(F(-), B) \Rightarrow \mathcal{A}(-, G(B))$.

Further, by the uniqueness clause (ii) the components φ_{AB} are bijections, so the natural transformations are indeed natural isomorphisms. Therefore $\mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))$ naturally in $A \in \mathcal{A}, B \in \mathcal{B}$. \square

Our theorem has a dual companion of course:

Theorem 163. *Given functors $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$, then $F \dashv G$ iff (i) there is a natural transformation $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$, for which (ii) for any $f: F(A) \rightarrow B$ there is a unique $g: A \rightarrow G(B)$ such that $f = \varepsilon_B \circ F(g)$.*

Evidently, we could have recruited either of these companion theorems as the basis of two further alternative definitions for $F \dashv G$ – as, for example, in (Awodey, 2006, §9.1).

29.3 Adjunctions compose

Recall Theorem 150: in a different notation, if (f, g) is a Galois connection between the posets \mathcal{P} and \mathcal{Q} , and (f', g') is a Galois connection between the posets \mathcal{Q} and \mathcal{R} , then $(f' \circ f, g \circ g')$ is a Galois connections between \mathcal{P} and \mathcal{R} .

Adjunctions similarly compose:

Adjoint further explored

Theorem 164. Given $\mathcal{A} \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} \mathcal{B}$ and $\mathcal{B} \begin{array}{c} \xrightarrow{F'} \\ \perp \\ \xleftarrow{G'} \end{array} \mathcal{C}$, then $\mathcal{A} \begin{array}{c} \xrightarrow{F'F} \\ \perp \\ \xleftarrow{GG'} \end{array} \mathcal{C}$.

Proof via homsets. Since $F' \dashv G'$, we have $\mathcal{C}(F'FA, C) \cong \mathcal{B}(FA, G'C)$, naturally in A – by Theorem 105(3) – and also naturally in C .

Also, since $F \dashv G$, we have $\mathcal{B}(FA, G'C) \cong \mathcal{A}(A, GG'C)$, naturally in A and in C .

So by Theorem 105(2), $\mathcal{C}(F'FA, C) \cong \mathcal{A}(A, GG'C)$ naturally in A and in C . Hence $F'F \dashv GG'$ \square

That was quick and easy. But there is perhaps some additional fun to be had by working through another argument:

Proof by units and counits. Since $F \dashv G$, there are a pair of natural transformations $\eta: 1_{\mathcal{A}} \Rightarrow GF$ and $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$, satisfying the usual triangle identities.

Since $F' \dashv G'$, there are natural transformations $\eta': 1_{\mathcal{B}} \Rightarrow G'F'$ and $\varepsilon': F'G' \Rightarrow 1_{\mathcal{C}}$, again satisfying the triangle identities.

We now define two more natural transformations by composition,

$$\begin{aligned} \eta'' : 1_{\mathcal{A}} &\xrightarrow{\eta} GF \xrightarrow{G\eta'F} GG'F'F \\ \varepsilon'' : F'FGG' &\xrightarrow{F'\varepsilon G'} F'G' \xrightarrow{\varepsilon'} 1_{\mathcal{C}} \end{aligned}$$

To show $F'F \dashv GG'$ it suffices to check that η'' and ε'' also satisfy the triangle identities.

Consider, then, the following diagram:

$$\begin{array}{ccccc} F'F & \xrightarrow{F'F\eta} & F'FGF & \xrightarrow{F'FG\eta'F} & F'FGG'F'F \\ & \searrow 1_{F'F} & \downarrow F'\varepsilon F & & \downarrow F'\varepsilon G'F'F \\ & & F'F & \xrightarrow{F'\eta'F} & F'G'F'F \\ & & & \searrow 1_{F'F} & \downarrow \varepsilon'F'F \\ & & & & F'F \end{array}$$

‘Whiskering’ the triangle identity $\varepsilon F \circ F\eta = 1_F$ by F' shows that the top left triangle commutes. And whiskering the identity $\varepsilon'F' \circ F'\eta' = 1_{F'}$ on the other side by F shows that the bottom triangle commutes.

Further, the square commutes. For by either the naturality of ε or the naturality of η' , the following square commutes in the functor category:

$$\begin{array}{ccc}
 FG & \xrightarrow{FG\eta'} & FGG'F' \\
 \downarrow \varepsilon & & \downarrow \varepsilon_{G'F'} \\
 1 & \xrightarrow{\eta'} & G'F'
 \end{array}$$

And whiskering again gives the commuting square in the big diagram. [Exercise: check the claims about whiskering and the naturality square.]

So the whole big diagram commutes, and in particular the outer triangle commutes. But that tells us that $\varepsilon''F'F \circ F'F\eta'' = 1_{F'F}$ – which is one of the desired triangle identities for η'' and ε'' .

The other identity follows similarly. □

29.4 The uniqueness of adjoints

Now recall Theorem 151. This tells us that if (f, g) and (f, g') are both Galois connections between the posets \mathcal{P} and \mathcal{Q} , then $g = g'$. Likewise, if (f, g) and (f', g) are both Galois connections between the same posets, then $f = f'$.

The corresponding result for adjunctions more generally is this:

Theorem 165. *Adjoints are unique up to natural isomorphism. If $F \dashv G$ and $F \dashv G'$ then $G \cong G'$. If $F \dashv G$ and $F' \dashv G$ then $F \cong F'$.*

Proof. Assume we have $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$ and $F \dashv G': \mathcal{A} \rightarrow \mathcal{B}$. Then

$$\mathcal{A}(A, GB) \cong \mathcal{B}(FA, B) \cong \mathcal{A}(A, G'B)$$

naturally in $A \in \mathcal{A}$, $B \in \mathcal{B}$. It follows, using Theorem 105, that

$$(*) \quad \mathcal{A}(A, GB) \cong \mathcal{A}(A, G'B)$$

naturally in A and B .

(*)'s naturality in A means that $\mathcal{A}(-, GB) \cong \mathcal{A}(-, G'B)$, i.e. $\mathcal{Y}GB \cong \mathcal{Y}G'B$, where \mathcal{Y} is the Yoneda embedding. And then, by Theorem 127, $GB \cong G'B$.

Moreover, this holds naturally in B – intuitively, because the isomorphism is generated systematically from the isomorphism in (*) which is also natural in B – so $G \cong G'$.

To confirm this, note that \mathcal{Y} sends the diagram on the left in \mathcal{A} to the diagram on the right in **Set** for any $f: B \rightarrow B'$:

$$\begin{array}{ccc}
 GB & \xrightarrow{Gf} & GB' \\
 \downarrow \beta_B & & \downarrow \beta_{B'} \\
 G'B & \xrightarrow{G'f} & G'B'
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathcal{A}(-, GB) & \xrightarrow{\mathcal{A}(-, Gf)} & \mathcal{A}(-, GB') \\
 \downarrow \alpha_B & & \downarrow \alpha_{B'} \\
 \mathcal{A}(-, G'B) & \xrightarrow{\mathcal{A}(-, G'f)} & \mathcal{A}(-, G'B')
 \end{array}$$

where the α s are components of the natural transformation required by the naturality of $(*)$ in B , and $\beta_B = \alpha_B(1_{GB})$ by appeal to Theorem 122. But \mathcal{Y} is an embedding, remember, so each diagram commutes if and only if the other does. However, the diagram on the right commutes for all $f: B \rightarrow B'$ by the naturality in B ; hence the diagram on the left does too (embeddings must evidently preserve commutativity relations). So the β assemble into a natural transformation between G and G' .

The proof of the second half of the theorem is dual. □

We should note too an obvious companion theorem:

Theorem 166. *If $F \dashv G$ and $G \cong G'$ then $F \dashv G'$. Likewise, if $F \dashv G$ and $F \cong F'$ then $F' \dashv G$.*

Proof. By definition, given $F \dashv G: \mathcal{A} \rightarrow \mathcal{B}$, we have $\mathcal{B}(FA, B) \cong \mathcal{A}(A, GB)$ naturally in $A \in \mathcal{A}, B \in \mathcal{B}$.

But given $G \cong G'$, then it is almost immediate that $\mathcal{A}(A, GB) \cong \mathcal{A}(A, G'B)$, again naturally in $A \in \mathcal{A}, B \in \mathcal{B}$.

Hence by Theorem 105 again, $\mathcal{B}(FA, B) \cong \mathcal{A}(A, G'B)$, still naturally in $A \in \mathcal{A}, B \in \mathcal{B}$. Which means that $F \dashv G'$.

The other half of the theorem is dual. □

29.5 How left adjoints can be defined in terms of right adjoints

Theorem 151 states that each component of a Galois connection uniquely fixes the other. So we would hope to be able to explicitly define one such component in terms of the other, and Theorem 152 in fact tells us how to do this. For example, assuming there is a connection (f, g) between the posets (P, \leq) and (Q, \sqsubseteq) , we can define the left adjoint in terms of the right by setting $f(p)$ to be the minimum of $\{q \in Q \mid p \leq g(q)\}$ for every $p \in P$.

We have now shown, more generally, that each component of an adjunction uniquely fixes the other, at least up to isomorphism. We would expect that we can, similarly, characterize one functor in an adjunction in terms of its partner. So let's consider, in particular, how a left adjoint might be defined in terms of its right partner. (There will of course also be a dual story to be told about how right adjoints can be defined in terms of left ones. We can cheerfully leave spelling out the dual constructions and arguments as an exercise.)

Functions in Galois connections between posets correspond to adjoint functors between poset categories (see §28.1). And a minimum for the poset $\{q \in Q \mid p \leq g(q)\}$ corresponds to an initial object for the poset-as-category (see §6.1). So this suggests that we might be able to characterize a left adjoint as the initial object of some suitable category.

29.5 How left adjoints can be defined in terms of right adjoints

And this is indeed more or less the case. Suppose $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ are functors such that $F \dashv G$. Now consider the comma category $(A \downarrow G)$, for $A \in \mathcal{A}$ – we met this construction at the end of §19. To recap,

- (a) the objects of $(A \downarrow G)$ are pairs $\langle B, f \rangle$ where B is a \mathcal{B} -object and $f: A \rightarrow GB$ is an arrow in \mathcal{A} ,
- (b) an arrow in $(A \downarrow G)$ from $\langle B, f \rangle$ to $\langle B', f' \rangle$ is a \mathcal{B} -arrow $j: B \rightarrow B'$ making the following commute:

$$\begin{array}{ccc}
 & & GB \\
 & \nearrow f & \downarrow Gj \\
 A & & \\
 & \searrow f' & \\
 & & GB'
 \end{array}$$

The definitions for the identity arrows and for composition of arrows in $(A \downarrow G)$ are the obvious ones.

Theorem 167. *Given an adjunction $\mathcal{A} \xrightleftharpoons[G]{F} \mathcal{B}$, the pair $\langle FA, \eta_A \rangle$ is initial in $(A \downarrow G)$ for any $A \in \mathcal{A}$.*

Proof. Let $\langle B, f \rangle$ be any object of $(A \downarrow G)$. We need to show that there is a unique arrow in $(A \downarrow G)$ from $\langle FA, \eta_A: A \rightarrow GFA \rangle$ to $\langle B, f \rangle$. That is to say, there must be (i) an arrow $j: FA \rightarrow B$ such that $f = Gj \circ \eta_A$, i.e.

$$\begin{array}{ccc}
 & & GFA \\
 & \nearrow \eta_A & \downarrow Gj \\
 A & & \\
 & \searrow f & \\
 & & GB
 \end{array}$$

commutes, and (ii) this arrow must be unique. But we've already proved *that* – see one half of Theorem 162. \square

We have a converse result too:

Theorem 168. *Given functors $\mathcal{A} \xrightleftharpoons[G]{F} \mathcal{B}$, then if (C) $\eta: 1_{\mathcal{A}} \rightarrow GF$ is a natural transformation and the pair $\langle FA, \eta_A \rangle$ is initial in $(A \downarrow G)$ for every $A \in \mathcal{A}$, then $F \dashv G$.*

So that tells us how to characterize a left adjoint for G when it exists, since left adjoints are unique up to isomorphism, i.e. as a functor F satisfying condition (C).

Proof. Suppose η is natural transformation, and that $\langle FA, \eta_A \rangle$ is initial in $(A \downarrow G)$ for every $A \in \mathcal{A}$. Then for every $f: A \rightarrow GB$ there is a unique $j: FA \rightarrow B$ such that $f = Gj \circ \eta_A$. Apply the other half of Theorem 162. \square

Adjoints further explored

But there was no fun in that instant proof. So, as an instructive and amusing exercise in diagram chasing here is

Another proof, by constructing a counit for η from first principles. We need to find a natural transformation $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$ such that η and ε satisfy the triangle equalities, i.e. such that $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$ for all $A \in \mathcal{A}$, and $G\varepsilon_B \circ \eta_{GB} = 1_{GB}$ for all $B \in \mathcal{B}$.

Taken any $B \in \mathcal{B}$. By hypothesis $\langle FGB, \eta_{GB} \rangle$ is initial in $(GB \downarrow G)$, so there is a unique arrow to the object $\langle B, 1_{GB} \rangle$. Call this unique arrow (hopefully!) ε_B . Then just by its definition, for any B we have (*):

$$\begin{array}{ccc} & & GFGB \\ & \nearrow^{\eta_{GB}} & \downarrow^{G\varepsilon_B} \\ GB & & GB \\ & \searrow_{1_{GB}} & \end{array}$$

which gives us one lot of the triangle identities for free. So it remains to show that (i) we also have the other triangle identities, and (ii) the components ε_B do indeed assemble into a natural transformation. Try before reading on!

For (i), we need to show that the following diagram commutes:

$$\begin{array}{ccc} & & FGFA \\ & \nearrow^{F\eta_A} & \downarrow^{\varepsilon_{FA}} \\ FA & & FA \\ & \searrow_{1_{FA}} & \end{array}$$

Since $\eta: 1_{\mathcal{A}} \Rightarrow GF$ is natural, for every $f: A \rightarrow A'$ with $A, A' \in \mathcal{A}$, there is a commuting naturality square. Take in particular the case where $f = \eta_A(!)$. Then paste on the commuting triangle of the type (*), with $B = FA$, to get the commuting rhombus:

$$\begin{array}{ccccc} A & \xrightarrow{\eta_A} & GFA & & \\ \downarrow \eta_A & & \downarrow \eta_{GFA} & \searrow 1_{GFA} & \\ GFA & \xrightarrow{GF\eta_A} & GFGFA & \xrightarrow{G\varepsilon_{FA}} & GFA \end{array}$$

Composing arrows, using the functoriality of G , and re-arranging we get the commuting triangle on the left:

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & GFA \\ & \searrow_{\eta_A} & \downarrow^{G(\varepsilon_{FA} \circ F\eta_A)} \\ & & GFA \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{\eta_A} & GFA \\ & \searrow_{\eta_A} & \downarrow^{Gj} \\ & & GFA \end{array}$$

Now, $\langle FA, \eta_A \rangle$ is initial in $(A \downarrow G)$ so there must be a *unique* arrow j from the initial object to itself such the triangle on the right commutes. But evidently $j =$

29.5 How left adjoints can be defined in terms of right adjoints

1_{FA} makes the triangle commute. But so, as we've just seen, does $j = \varepsilon_{FA} \circ F\eta_A$. Hence $\varepsilon_{FA} \circ F\eta_A = 1_{FA}$. Which establishes (i).

To establish (ii) – the naturality of $\varepsilon: FG \Rightarrow 1_{\mathcal{B}}$, when assembled from the components ε_B – we need to show that for any $g: B \rightarrow B'$, the following commutes (**):

$$\begin{array}{ccc} FGB & \xrightarrow{FGg} & FGB' \\ \downarrow \varepsilon_B & & \downarrow \varepsilon_{B'} \\ B & \xrightarrow{g} & B' \end{array}$$

We again start by taking a naturality square for η , this time for $f = Gg: GB \rightarrow GB'$, and then paste on a commuting triangle of type (*), to get the commuting rhombus

$$\begin{array}{ccccc} GB & \xrightarrow{Gg} & GB' & & \\ \downarrow \eta_{GB} & & \downarrow \eta_{GB'} & \searrow 1_{GB'} & \\ GFGB & \xrightarrow{GFGg} & GFGB' & \xrightarrow{G\varepsilon_{B'}} & GB' \end{array}$$

Again composing arrows, using the functoriality of G , and re-arranging we get the commuting triangle on the left:

$$\begin{array}{ccc} & GFGB & \\ \eta_{GB} \nearrow & \downarrow G(\varepsilon_{B'} \circ FGg) & \\ GB & & GB' \\ Gg \searrow & & \end{array} \qquad \begin{array}{ccc} & GFGB & \\ \eta_{GB} \nearrow & \downarrow G\varepsilon_B & \downarrow G(g \circ \varepsilon_B) \\ GB & \xrightarrow{1_{GB}} & GB \\ Gg \searrow & \downarrow Gg & \downarrow \\ & GB' & \end{array}$$

On the right, we've pasted together (*) with a trivially commuting triangle, and then composed the downwards arrows to give the big triangle. However, by assumption, $\langle FGB, \eta_{GB} \rangle$ is initial in the comma category $(GB \downarrow G)$, so there is a *unique* arrow j to $\langle B', g \rangle$ such that $g = Gj \circ \eta_{GB}$. Whence $\varepsilon_{B'} \circ FGg = j = g \circ \varepsilon_B$, proving (**) commutes and establishing (ii). \square

Here's a nice corollary:

Theorem 169. *Suppose $G: \mathcal{B} \rightarrow \mathcal{A}$ is a functor. If the derived comma category $(A \downarrow G)$ has an initial object for every $A \in \mathcal{A}$, then G has a left adjoint.*

Proof. Choose an initial object for each $(A \downarrow G)$: it is a pair that we will write (hopefully!) as $\langle FA, \eta_A \rangle$, with $FA \in \mathcal{B}$, and $\eta_A: A \rightarrow GFA$.

So we now define a functor $F: \mathcal{A} \rightarrow \mathcal{B}$ which sends an object $A \in \mathcal{A}$ to this $FA \in \mathcal{B}$. How should F act on an arrow $f: A \rightarrow A'$? It must yield an arrow from FA to FA' . But since $\langle FA, \eta_A \rangle$ is initial, we know that there is exactly one arrow in $(A \downarrow G)$ from $\langle FA, \eta_A \rangle$ to $\langle FA', \eta_{A'} \circ f \rangle$. That is to say, there is a

Adjoint further explored

unique $g: FA \rightarrow FA'$ such that $\eta_{A'} \circ f = Gg \circ \eta_A$. Put $Ff = g$, and it is easy enough to check that F is functorial.

So now consider this diagram:

$$\begin{array}{ccc} A & \xrightarrow{f} & A' \\ \eta_A \downarrow & & \downarrow \eta_{A'} \\ GFA & \xrightarrow{GFf} & GFA' \end{array}$$

We've defined Ff to make this commute. But this is a naturality square showing that the components η_A assemble into a natural transformation $\eta: 1_{\mathcal{A}} \rightarrow GF$.

So, in sum, $F: \mathcal{A} \rightarrow \mathcal{B}$ as defined is such that (C), $\eta: 1_{\mathcal{A}} \rightarrow GF$ is a natural transformation and the pair $\langle FA, \eta_A \rangle$ is initial in $(A \downarrow G)$ for every $A \in \mathcal{A}$. Hence, by the previous theorem, $F \dashv G$. \square

29.6 Another way of getting new adjunctions from old

We've already met one way of getting new adjunctions from old, i.e. simple composition. Finally in this chapter, we now introduce another.

Definition 128. Given a functor $F: \mathcal{C} \rightarrow \mathcal{D}$ and small category \mathbf{J} , then the functor $[\mathbf{J}, F]: [\mathbf{J}, \mathcal{C}] \rightarrow [\mathbf{J}, \mathcal{D}]$ sends a functor $K: \mathbf{J} \rightarrow \mathcal{C}$ to $F \circ K: \mathbf{J} \rightarrow \mathcal{D}$.

Strictly speaking that's an incomplete definition. We need to specify not just how $[\mathbf{J}, F]$ acts on objects in $[\mathbf{J}, \mathcal{C}]$ (i.e. acts on functors), but how it acts on arrows (i.e. on natural transformations). But the needed completion, as often in defining functors, writes itself. For what is the obvious way for $[\mathbf{J}, F]$ to act on a natural transformation from K to K' with components $\alpha_J: KJ \rightarrow K'J$ (for $J \in \mathbf{J}$ and functors $K, K': \mathbf{J} \rightarrow \mathcal{C}$)? By sending it, of course, to the natural transformation from $F \circ K$ to $F \circ K'$ with components $F\alpha_J: FKJ \rightarrow FK'J$. Full functoriality is then immediate.

We can now state our result about how a given adjunction between functors F and G generates a new adjunction between new-style functors $[\mathbf{J}, F]$ and $[\mathbf{J}, G]$:

Theorem 170. *If $F \dashv G: \mathcal{C} \rightarrow \mathcal{D}$ then $[\mathbf{J}, F] \dashv [\mathbf{J}, G]: [\mathbf{J}, \mathcal{C}] \rightarrow [\mathbf{J}, \mathcal{D}]$.*

Proof. Take functors $K: \mathbf{J} \rightarrow \mathcal{C}$, $L: \mathbf{J} \rightarrow \mathcal{D}$. Then take any natural transformation $\beta: FK \Rightarrow L$ living as an arrow in $[\mathbf{J}, \mathcal{D}]$. This has components $\beta_J: FKJ \rightarrow LJ$ living in $\mathcal{D}(FKJ, LJ)$. By the adjunction $F \dashv G$ these components are in a natural bijection with arrows $\alpha_J: KJ \rightarrow GLJ$ living in $\mathcal{C}(KJ, GLJ)$, and these assemble into a natural transformation $\alpha: K \Rightarrow GL$ which lives in $[\mathbf{J}, \mathcal{C}]$ (the adjunction is easily checked to associate naturality squares with naturality squares). In this way we set up a natural one-to-one correspondence between natural transformations like α and β .

29.6 Another way of getting new adjunctions from old

So we have established that there is, naturally, a bijection

$$[\mathbf{J}, \mathcal{D}]([\mathbf{J}, F]K, L) \cong [\mathbf{J}, \mathcal{C}](K, [\mathbf{J}, G]L),$$

which proves $[\mathbf{J}, F] \dashv [\mathbf{J}, G]$. □

30 Adjoint functors and limits

NB: This chapter, like the previous two, is taken, unrevised, from an earlier set of Notes on Category Theory. It needs a great deal of rewriting, not to mention checking for bad errors! However, if you have got this far then it should still be useful, and it gets us to a sensible interim stopping point.

We now turn to some key results which tell us how adjoint functors interact with limits. A key bit of news is that right adjoints preserve limits: and dually, exactly as you would now expect, left adjoints preserve co-limits.

30.1 Limit functors as adjoints

(a) Suppose the category \mathcal{C} has all limits of shape \mathbf{J} . Three observations:

- (1) By Theorem 49, the cones over $D: \mathbf{J} \rightarrow \mathcal{C}$ with vertex C correspond one-to-one with \mathcal{C} -arrows from C to $\lim_{\leftarrow \mathbf{J}} D$.
- (2) But by the remark after Theorem 109, the set of cones over $D: \mathbf{J} \rightarrow \mathcal{C}$ with vertex C is the hom-set $[\mathbf{J}, \mathcal{C}](\Delta(C), D)$. Here $\Delta: \mathcal{C} \rightarrow [\mathbf{J}, \mathcal{C}]$ is the functor introduced just after that theorem, which sends an object $C \in \mathcal{C}$ to the constant functor $\Delta_C: \mathbf{J} \rightarrow \mathcal{C}$. (For convenience, understand the cones here austerey).
- (3) The set of \mathcal{C} -arrows from C to the limit vertex $\lim D$ is $\mathcal{C}(C, \lim(D))$, where $\lim: [\mathbf{J}, \mathcal{C}] \rightarrow \mathcal{C}$ is the functor introduced in §22.6, a functor that exists if \mathcal{C} has all limits of shape \mathbf{J} and that sends a diagram D of shape \mathbf{J} in \mathcal{C} to some limit object in \mathcal{C} .

So, in summary, still assuming that \mathcal{C} has all limits of shape \mathbf{J} , the situation is this. We have a pair of functors $\mathcal{C} \xrightleftharpoons[\lim]{\Delta} [\mathbf{J}, \mathcal{C}]$ such that

$$[\mathbf{J}, \mathcal{C}](\Delta(C), D) \cong \mathcal{C}(C, \lim(D)).$$

Moreover, the isomorphism that is given in our proof of Theorem 49 arises in a natural way, without making any arbitrary choices.¹ So, we can take it that the

¹Careful: there were arbitrary choices made in determining what \lim does. But once \lim is fixed, the isomorphism arises naturally.

isomorphism is natural in $C \in \mathcal{C}$ and $D \in [\mathbf{J}, \mathcal{C}]$. Hence Δ has a right adjoint, and one such right adjoint is Lim .

We now argue in the opposite direction starting from the assumption that the diagram Δ has a right adjoint, call it L .

Suppose that D is a diagram $D: \mathbf{J} \rightarrow \mathcal{C}$. Applying Theorem 163 about a universal mapping property of adjunctions, for any arrow $c: \Delta(C) \rightarrow D$ in $[\mathbf{J}, \mathcal{C}]$ – in other words for any cone over D with vertex C – there is a unique arrow $u: C \rightarrow L(D)$ in \mathcal{C} , such that $c = \varepsilon_D \circ \Delta(u)$, where ε is the co-unit of the adjunction.

By the definition of Δ , $\Delta(u)$ is the natural transformation from Δ_C to $\Delta_{L(D)}$ with every component equal to u .

And by §29.1 (3), ε_D is the transpose of $1_{L(D)}$, i.e. is some arrow $\pi: \Delta_{L(D)} \rightarrow D$ in $[\mathbf{J}, \mathcal{C}]$, i.e. is some particular cone π over D with vertex $L(D)$.

Taken component-wise, the equation $c = \varepsilon_D \circ \Delta(u)$ tells us that for each $J \in \mathbf{J}$, $c_J = \pi_J \circ u$. In other words any cone c factors through our cone π via the unique u . Hence the cone π with vertex $L(D)$ and projection arrows π_J is a limit cone for D . However, D was any diagram $D: \mathbf{J} \rightarrow \mathcal{C}$. Therefore \mathcal{C} has all limits of shape \mathbf{J} .

Summing up, we get the following nice theorem:

Theorem 171. *If category \mathcal{C} has all limits of shape \mathbf{J} , then Δ has a right adjoint, and indeed $\Delta \dashv Lim$. Conversely, if Δ has any right adjoint, then \mathcal{C} has all limits of shape \mathbf{J} .*

(b) Keeping \mathbf{J} fixed, we can make Δ 's dependence on \mathcal{C} explicit by writing $\Delta_{\mathcal{C}}: \mathcal{C} \rightarrow [\mathbf{J}, \mathcal{C}]$. Similarly we can explicitly write $Lim_{\mathcal{C}}: [\mathbf{J}, \mathcal{C}] \rightarrow \mathcal{C}$. Then we have the following easy corollary of the last theorem:

Theorem 172. *Suppose the categories \mathcal{B} and \mathcal{C} have all limits of shape \mathbf{J} . Then if $G: \mathcal{C} \rightarrow \mathcal{B}$ is a right adjoint (i.e. has a left adjoint), $G \circ Lim_{\mathcal{C}} \cong Lim_{\mathcal{B}} \circ [\mathbf{J}, G]$.*

Proof. Let $F: \mathcal{B} \rightarrow \mathcal{C}$ be left adjoint to G , and consider this pair of diagrams:

$$\begin{array}{ccc}
 \mathcal{B} & \xrightarrow{F} & \mathcal{C} \\
 \Delta_{\mathcal{B}} \downarrow & & \downarrow \Delta_{\mathcal{C}} \\
 [\mathbf{J}, \mathcal{B}] & \xrightarrow{[\mathbf{J}, F]} & [\mathbf{J}, \mathcal{C}]
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathcal{B} & \xleftarrow{G} & \mathcal{C} \\
 Lim_{\mathcal{B}} \uparrow & & \uparrow Lim_{\mathcal{C}} \\
 [\mathbf{J}, \mathcal{B}] & \xleftarrow{[\mathbf{J}, G]} & [\mathbf{J}, \mathcal{C}]
 \end{array}$$

Claim: the left-hand diagram commutes. (i) On the south-west path, an object $B \in \mathcal{B}$ is sent by $\Delta_{\mathcal{B}}$ to the functor $\Delta_B: \mathbf{J} \rightarrow \mathcal{B}$ which sends every object to B and every arrow to 1_B ; and this is sent in turn by $[\mathbf{J}, F]$ to the functor which sends every object to FB and every arrow to 1_{FB} , i.e. the functor Δ_{FB} . (ii) On the north-east path, an object $B \in \mathcal{B}$ is sent by F to FB , and this is sent by $\Delta_{\mathcal{C}}$ to the functor Δ_{FB} again.

Now, given the assumption that \mathcal{B} and \mathcal{C} have all limits of shape \mathbf{J} , $\Delta_{\mathcal{B}}$ and $\Delta_{\mathcal{C}}$ have right adjoints $Lim_{\mathcal{B}}$ and $Lim_{\mathcal{C}}$. And since $F \dashv G$, $[\mathbf{J}, F] \dashv [\mathbf{J}, G]$, by Theorem 170.

So our right-hand diagram records the adjoints of the functors in the left-hand diagram. We now know that the composite left-adjoint functors $\Delta_{\mathcal{C}} \circ F$ and $[\mathbf{J}, F] \circ \Delta_{\mathcal{B}}$ are the same. By Theorem 164 about the composition of adjunctions, their right-adjoints are $G \circ Lim_{\mathcal{C}}$ and $Lim_{\mathcal{B}} \circ [\mathbf{J}, G]$. And these composites, being right adjoint to the same functor, must be naturally isomorphic by Theorem 165. \square

30.2 Right adjoints preserve limits

We can usefully begin by restating part of a key definition and reminding ourselves of a basic theorem:

Definition 92 A functor $G: \mathcal{C} \rightarrow \mathcal{B}$ preserves limits of shape \mathbf{J} iff, for any diagram $D: \mathbf{J} \rightarrow \mathcal{C}$, if $[L, \pi_J]$ is a limit cone over D , then $[GL, G\pi_J]$ is a limit cone over $G \circ D: \mathbf{J} \rightarrow \mathcal{B}$.

Theorem 101 The covariant hom-functor $\mathcal{C}(A, -): \mathcal{C} \rightarrow \mathbf{Set}$, for any A in the category \mathcal{C} , preserves all limits that exist in \mathcal{C} .

Now, this theorem is easily seen to imply the following:

Theorem 173. Any set-valued functor $G: \mathcal{C} \rightarrow \mathbf{Set}$ which is a right adjoint (i.e. has a left adjoint) preserves all limits that exist in \mathcal{C} .

Proof. Suppose we have a functor F such that $F \dashv G$. Then

$$GA \cong \mathbf{Set}(1, GA) \cong \mathcal{C}(F1, A)$$

with both isomorphisms natural in A (the first relies on the familiar association in \mathbf{Set} between elements of a set and arrows from a terminal object into that set). Hence G is naturally isomorphic to the hom-functor $\mathcal{C}(F1, -)$. But the latter preserves limits, by Theorem 101. Hence so does G , by Theorem 134. \square

We now show that there is in fact nothing special here about set-valued functors. We can prove quite generally:

Theorem 174. If the functor $G: \mathcal{C} \rightarrow \mathcal{B}$ is a right adjoint (i.e. has a left adjoint), it preserves all limits that exist in \mathcal{C} .

Proof from basic principles about limits and adjoints. Suppose that G has the left adjoint $F: \mathcal{B} \rightarrow \mathcal{C}$; and suppose also that the diagram $D: \mathbf{J} \rightarrow \mathcal{C}$ has a limit cone $[L, \pi_J]$ in \mathcal{C} .

Then $[GL, G\pi_J]$ is certainly a cone over $G \circ D$ in \mathcal{B} . We need to show, however, that it is a *limit* cone. That is to say, we need to show that, if we take any cone

$[B, b_J]$ over $G \circ D$, there is a unique $u: B \rightarrow GL$ such that (i) for all $J \in \mathbf{J}$, $b_J = G\pi_J \circ u$.

Well, take such a cone $[B, b_J]$ over $G \circ D$. Then, going back in the other direction, $[FB, \bar{b}_J]$ is a cone over D in \mathcal{C} , where $\bar{b}_J: FB \rightarrow D_J$ is the transpose of $b_J: B \rightarrow GD_J$ under the adjunction.

Why is $[FB, \bar{b}_J]$ a cone? Suppose we have an arrow $d: D_K \rightarrow D_K$. Then by assumption, since $[B, b_J]$ is a cone over $G \circ D$, $b_K = Gd \circ b_J$. Hence $\bar{b}_K = \overline{Gd \circ b_J} = d \circ \bar{b}_J$, with the second equation by Theorem 159 (1). Which indeed makes $[FB, \bar{b}_J]$ a cone too.

And now we add that $[FB, \bar{b}_J]$ must factor through $[L, \pi_J]$ via a unique $v: FB \rightarrow L$ such that (ii) for all $J \in \mathbf{J}$, $\bar{b}_J = \pi_J \circ v$.

So the state of play is: we have found a unique $v: FB \rightarrow L$; we want to find a suitable $u: B \rightarrow GL$. The hopeful thought is that one will turn out to be the transpose of the other under the adjunction.

The adjunction means that $\mathcal{C}(FB, C) \cong \mathcal{B}(B, GC)$ naturally in C . Which in turn means that the following square commutes, for any $\pi_J: L \rightarrow D_J$:

$$\begin{array}{ccc} \mathcal{C}(FB, L) & \xrightarrow{\pi_J \circ -} & \mathcal{C}(FB, D_J) \\ \downarrow & & \downarrow \\ \mathcal{B}(B, GL) & \xrightarrow{G\pi_J \circ -} & \mathcal{B}(B, GD_J) \end{array}$$

where the vertical arrows are components of the natural transformation which sends an arrow to its transform. Chase the arrow $v: FB \rightarrow L$ round the diagram in both directions and we get $G\pi_J \circ \bar{v} = \pi_J \circ \bar{v}$. Therefore, using (ii), if we put $u = \bar{v}$, we indeed get as required that (i) for all $J \in \mathbf{J}$, $b_J = G\pi_J \circ u$.

It just remains to confirm u 's uniqueness. Suppose that $[B, b_J]$ factors through $[GL, G\pi_J]$ by some $u' = \bar{w}$. Then for all $J \in \mathbf{J}$, $b_J = G\pi_J \circ \bar{w}$. We show as before that $\bar{b}_J = \pi_J \circ w$, whence $[FB, \bar{b}_J]$ factors through $[L, \pi_J]$ via w . By the uniqueness of factorization, $w = v$ again. \square

A more compressed proof. Again, suppose that G has the left adjoint $F: \mathcal{B} \rightarrow \mathcal{C}$; and suppose also that the diagram $D: \mathbf{J} \rightarrow \mathcal{C}$ has a limit cone $[L, \pi_J]$ in \mathcal{C} . Then, using the notation ' $\mathcal{C}(X, D)$ ' as shorthand for the functor $\mathcal{C}(X, -) \circ D$, we have

$$\begin{aligned} \mathcal{B}(B, GL) &\cong \mathcal{C}(FB, L) \\ &\cong \text{Lim } \mathcal{C}(FB, D) \\ &\cong \text{Lim } \mathcal{B}(B, GD) \\ &\cong \text{Cone}(B, GD). \end{aligned}$$

all naturally in B . So the functor $\text{Cone}(-, GD)$, being naturally isomorphic to $\mathcal{B}(-, GL)$ is representable, and is represented by GL , and therefore has a universal element of the form $\langle GL, g \rangle$. But such a universal element is a limit cone with vertex GL . Hence G preserves the limit $[L, \pi_J]$. \square

But compression doesn't always make for illumination, and our second proof (see Leinster 2014, p. 158; compare Awodey 2006, pp. 225–6) needs some commentary.

The first line of course comes from the adjunction, and the second from the fact that the hom-functor $\mathcal{C}(FB, -)$ preserves limits, by Theorem 101. The move from the third to the fourth line is by Theorem ?? [*The referenced theorem needs to be replaced!*]. And the arguments at the end about representability, universal elements and limits appeal to Theorems 141 and 146.

So that leaves the move from the second to the third line, which obviously invokes the adjunction between F and G again. We know that $\mathcal{C}(FB, X) \cong \mathcal{B}(B, GX)$ naturally in X , i.e. $\mathcal{C}(FB, -)$ is naturally isomorphic to $\mathcal{B}(B, G-)$, hence by whiskering, $\mathcal{C}(FB, -) \circ D$ is naturally isomorphic to $\mathcal{B}(B, G-) \circ D$. Now apply Theorem 110 and we can conclude that $\text{Lim } \mathcal{C}(FB, D) \cong \text{Lim } \mathcal{B}(B, GD)$.

Which all goes to combine a bunch of earlier results into a neat package: but my own feeling is that the first direct proof from the underlying principles reveals better what is really going on here.

30.3 Some examples

Right adjoints preserve limits. Dually, of course, left adjoints preserve colimits (we surely needn't pause at this stage in the game to state the duals of the theorems in the last couple of sections!). So we now mention just a few elementary examples of (co)limit preservation – and also some examples where we can argue from non-preservation to the non-existence of adjoints.

- (1) Back in §17.2, Ex. (4) we noted that the forgetful functor $U: \mathbf{Mon} \rightarrow \mathbf{Set}$ preserves limits. But we now have another proof: U has a left adjoint (by §28.2, Ex. (3)) i.e. it *is* a right adjoint, so indeed must preserve limits.

There are other examples of this kind, involving a forgetful functor $U: \mathbf{Alg} \rightarrow \mathbf{Set}$, where \mathbf{Alg} is a category of sets equipped with some algebraic structure for U to ignore. Such a forgetful U standardly has a left adjoint, so must preserve whatever limits exist in the relevant \mathbf{Alg} .

Further, a left-adjoint to U must preserve existing colimits in \mathbf{Set} . But \mathbf{Set} has *all* colimits; so that this indeed requires the left-adjoints in such cases to be rather lavish constructions (as we saw them to be).

- (2) Consider exponentials again.

We noted that if \mathcal{C} is a category with exponentiation, and hence with products, exponentiation is right adjoint to taking products: $(- \times B) \dashv (-)^B$.

Since the functor $(-)^B$ is a right adjoint, it preserves such limits as exist in \mathcal{C} . So take in particular the functor $A: 2 \rightarrow \mathcal{C}$ (where as before 2 is the discrete two object category with objects 0, 1). Then $A_0 \times A_1$ is the vertex of a limit over A . Hence $(A_0 \times A_1)^B$ is the vertex of a limit over $(-)^B \circ A$.

But the canonical limit over that composite functor is $A_0^B \times A_1^B$. Hence $(A_0 \times A_1)^B \cong A_0^B \times A_1^B$.

Since the functor $- \times B$ is a left adjoint, it preserves such colimits as exist in \mathcal{C} . Assume \mathcal{C} has coproducts. Then, by a similar argument, $(A_0 + A_1) \times B \cong (A_0 \times B) + (A_1 \times B)$.

- (3) Take the discussion in §27.3, Ex. (6) where we looked at the Galois connection between two functions between posets of equivalence classes of wffs, with the left adjoint a trivial ‘add a dummy variable’ map, and the right adjoint provided by applying a universal quantifier. This carries over to an adjunction of functors between certain poset categories. Since quantification is a right adjoint, it preserves limits, and in particular preserves products, which are conjunctions in this category. Which reflects the familiar logical truth that $\forall x(Px \wedge Qx) \equiv (\forall xPx \wedge \forall xQx)$.
- (4) Claim: the forgetful functor $F: \mathbf{Grp} \rightarrow \mathbf{Set}$ has no right adjoint. Proof: the trivial one-object group is initial in \mathbf{Grp} ; but a singleton is not initial in \mathbf{Set} ; so there is a colimit which F doesn’t preserve and it therefore cannot be a left adjoint.
- (5) The proof of Theorem 77 tells us that the forgetful functor $F: \mathbf{Mon} \rightarrow \mathbf{Set}$ fails to preserve all epimorphisms. By Theorem 95 this implies that F doesn’t preserve all pushouts, and hence doesn’t preserve all colimits. Hence this forgetful functor too can’t be a left adjoint. Compare the arm-waving argument to the same conclusion in §28.2. Ex. (5).

30.4 The Adjoint Functor Theorems

Right adjoints preserve limits. What about the converse? If a functor preserves limits must it be a right adjoint? Well, given some results already to hand, we can easily prove the following:

Theorem 175. *If the category \mathcal{B} has all limits, and the functor $G: \mathcal{B} \rightarrow \mathcal{A}$ preserves them, then G is a right adjoint.*

Proof. If \mathcal{B} has all limits and G preserves them, then for any $A \in \mathcal{A}$, $(A \downarrow G)$ has all limits (by Theorem 104, and the remark immediately after its proof).

So any $(A \downarrow G)$ in particular has a limit for the big diagram-as-part-of-a-category consisting of the whole of $(A \downarrow G)$ – or in terms of diagrams-as-functors, it has a limit for the identity functor $1_{(A \downarrow G)}$. Hence by Theorem 50, each $(A \downarrow G)$ has an initial object. Hence by Theorem 169, there is a functor $F: \mathcal{A} \rightarrow \mathcal{B}$ such that $F \dashv G$. □

And now we see the proof, we see that the condition that \mathcal{B} has *all* limits overshoots: the result will go through so long as \mathcal{B} has sufficiently large limits, enough to guarantee that all the functors $1_{(A \downarrow G)}$ have a limit.

Adjoint functors and limits

This theorem looks neat but is in fact not very useful. Having all sufficiently large limits is a hard condition to fulfil. More precisely, we have

Theorem 176. *If a category \mathcal{C} has limits for diagrams over all categories of size up to the size of the collection of \mathcal{C} 's arrows, then \mathcal{C} has at most one arrow between any two objects.*

For example, the condition of having small limits is not satisfied by typical small categories – because, in the terminology of §3.3 *preordercatex*, a complete small category has to be a *pre-order* category.

Proof. Let \mathbf{J} be a discrete category of the same cardinality as the set of arrows of \mathcal{C} . Let $D: \mathbf{J} \rightarrow \mathcal{C}$ be the diagram which sends every object in \mathbf{J} to B . By hypothesis, D has a limit, namely the product $\prod_{J \in \mathbf{J}} D(J)$ (so this is the product of B with itself, \mathbf{J} -many times).

Suppose there are objects $A, B \in \mathcal{C}$ with arrows $f_1, f_2: A \rightarrow B$ where $f_1 \neq f_2$. Simple cardinality considerations show that this further supposition leads to contradiction. Which proves the theorem.

We start by asking: how many different arrows $A \rightarrow \prod_{J \in \mathbf{J}} D(J)$ are there?

Theorem 34 showed that if \mathbf{J} is the discrete two object category, then there are four such arrows. Generalizing the proof in the obvious way shows that if $|\mathbf{J}|$ is the cardinality of the objects of \mathbf{J} , there are $2^{|\mathbf{J}|}$ different arrows from $A \rightarrow \prod_{J \in \mathbf{J}} D(J)$.

Hence our suppositions imply that there is a subset of the arrows in \mathcal{C} whose cardinality is strictly greater than the cardinality of the set of arrows in \mathcal{C} . Contradiction. \square

So, in sum, Theorem 175 is of very limited application. If we want a more widely useful result of the form ‘Given such-and such conditions on the functor $G: \mathcal{B} \rightarrow \mathcal{A}$ and the categories it relates, then G is a right adjoint’, we’ll need to consider a new bunch of conditions.

Here are two such theorems of rather wider application (the labels are standard):

Theorem 177 (The General Adjoint Functor Theorem). *If category \mathcal{B} is a locally small category with all small limits, and the functor $G: \mathcal{B} \rightarrow \mathcal{A}$ is such that for each $A \in \mathcal{A}$, the category $(A \downarrow G)$ has a weakly initial set, then G is a right adjoint iff it preserves all small limits.*

(GAFT: Alternative version) *If category \mathcal{B} is a locally small category with all small limits, and $G: \mathcal{B} \rightarrow \mathcal{A}$ is a functor, then G is a right adjoint iff it preserves all small limits and satisfies the solution set condition.*

Theorem 178 (The Special Adjoint Functor Theorem). *If the categories \mathcal{A} and \mathcal{B} are locally small, and \mathcal{B} has all small limits, is well powered, and has a coseparating set of objects, then G is a right adjoint iff it preserves all small limits.*

But to investigate these theorems properly would require not just explaining the concepts ‘weekly initial set’, ‘solution set condition’, ‘well powered’ and ‘coseparating’ and then doing the proofs, but also explaining what might motivate the conditions our new concepts are used to state, and also explaining why the resulting theorems, with just those conditions in play, might be of interest and use. That’s a non-trivial expositional task, and one I am going to shirk in this current version of these Notes. If you want to follow up the technical details, which aren’t particularly difficult, I can refer you to for example Leinster (2014, pp. 159–164, 171–173) and Awodey (2006, §9.8). But I’m not sure I yet have a sufficiently good grip on the place of these theorems in the scheme of things to give an illuminating account of the motivations here.

Indeed, the Adjoint Functor Theorems arguably sit at the boundary between basic category theory and the beginnings of more serious stuff. So given the intended limited remit of *these* Notes, this is in any case the point at which I should probably stop for the moment.

Bibliography

- Adámek, J., Herrlich, H., and Strecker, G., 2009. *Abstract and Concrete Categories: The Joy of Cats*. Mineola, New York: Dover Publications. URL <http://www.tac.mta.ca/tac/reprints/articles/17/tr17.pdf>. Originally published 1990.
- Awodey, S., 2006. *Category theory*, vol. 49 of *Oxford Logic Guides*. Oxford: Oxford University Press.
- Borceux, F., 1994. *Handbook of Categorical Algebra 1, Basic Category Theory*, vol. 50 of *Encyclopedia of Mathematics and its Applications*. Cambridge: Cambridge University Press, Cambridge.
- Eilenberg, S. and Mac Lane, S., 1942. Natural isomorphisms in group theory. *Proceedings of the National Academy of Sciences of the United States of America*, 28: 537–543.
- Eilenberg, S. and Mac Lane, S., 1945. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58: 231–294.
- Forster, T., 1995. *Set Theory with a Universal Set*. Oxford: Clarendon Press, 2nd edn.
- Freyd, P., 1965. The theories of functors and models. In J. W. Addison, L. Henkin, and A. Tarski (eds.), *The Theory of Models*, pp. 107–120. North-Holland Publishing Co.
- Goedecke, J., 2013. Category theory. URL <https://www.dpms.cam.ac.uk/~jg352/pdf/CategoryTheoryNotes.pdf>.
- Goldblatt, R., 2006. *Topoi: The Categorical Analysis of Logic*. Mineola, New York: Dover Publications, revised edn.
- Johnstone, P., 2002. *Sketches of an Elephant: A Topos Theory Compendium, Vol. 1*, vol. 43 of *Oxford Logic Guides*. Clarendon Press.
- Lawvere, F. W., 1969. Adjointness in foundations. *Dialectica*, 23: 281–296.
- Lawvere, F. W. and Schanuel, S. H., 2009. *Conceptual Mathematics: A first introduction to categories*. Cambridge: Cambridge University Press, 2nd edn.
- Leinster, T., 2014. *Basic Category Theory*. Cambridge: Cambridge University Press.

- Mac Lane, S., 1997. *Categories for the Working Mathematician*. New York: Springer, 2nd edn.
- Marquis, J.-P., 2008. *From a Geometrical Point of View: A Study of the History and Philosophy of Category Theory*. New York: Springer.
- Mazur, B., 2008. When is one thing equal to some other thing? In B. Gold and R. Simons (eds.), *Proof and Other Dilemmas: Mathematics and Philosophy*. Mathematical Association of America. URL http://www.math.harvard.edu/~mazur/preprints/when_is_one.pdf.
- Munkres, J. R., 2000. *Topology*. Prentice Hall, 2nd edn.
- Schubert, H., 1972. *Categories*. New York, Heidelberg, Berlin: Springer.
- Sellars, W., 1963. Philosophy and the scientific image of man. In *Science, Perception and Reality*. Routledge & Kegan Paul.
- Simmons, H., 2011. *An Introduction to Category Theory*. Cambridge: Cambridge University Press.