Category Theory

Prof. Dr. B. Pareigis

Summer Semester 2004

June 14, 2004

Contents

Introduction 3
1. Foundations 6
1.1. Graphs 6
1.2. Monoids 8
1.3. Categories 12
1.4. Constructions on categories 18
1.4.1. Slice categories 18
1.4.2. Subcategories 20
1.4.3. The product of categories 20
1.4.4. The dual of a category 21
1.4.5. The free category generated by a graph 21
1.5. Functors 22
1.6. Special types of functors 24
1.6.1. Underlying functors 24
1.6.2. Free functors 25
1.6.3. Powerset functors 26
1.6.4. Hom functors or Mor functors 26
1.7. Natural transformations 27
1.7.1. Natural transformations between functors I 27
1.7.2. Diagrams 30
1.7.3. Commutative diagrams 31
1.7.4. Graph homomorphisms by commutative diagrams 33
1.7.5. Associativity by commutative diagrams 33
1.7.6. Natural transformations 35

1.7.7. Natural isomorphisms 38
1.7.8. Natural transformations between functors II 39
1.7.9. Natural transformations of graphs 42
1.7.10. Combining natural transformations and functors 43
2. Representable functors and the Yoneda Lemma 45
2.1. Representable functors 45
2.2. Terminal and initial objects 46
2.3. The extension principle 47
2.4. Isomorphisms 48
2.5. Monomorphisms and epimorphisms 50
2.6. Subobjects 54
2.7. Products 55
2.8. Equalizers 61
2.9. Universal elements 66
2.10. The Yoneda Lemma 71
2.11. Algebraic Structures 74
2.12. Limits 79
3. Adjoint functors and monads 84
3.1. Adjoint functors 84
3.2. Monads or Triples 86
3.3. Factorizations of a monad 87
3.4. Algebraic Categories 89
3.5. More facts about adjoint functors 89
References 90

Introduction

0.0.1. **Remark (History.).** – 1945 Categories first appear "officially" in S. Eilenberg and S. Mac Lane's paper "General theory of natural equivalences", (Trans. AMS 58, 1945, 231-294).
– 1950's The main applications were originally in the fields of algebraic topology, particularly homology theory, and abstract algebra.
– 1960's Grothendieck et al. began using category theory with great success in algebraic geometry.
– 1970's Lawvere and others began applying categories to logic, revealing some deep and surprising connections.
– 1980's Applications began appearing in computer science, linguistics, cognitive science, philosophy, . . .
– 1990's Applications to theoretical physics – quantum groups, algebraic quantum field theories, higher category theory.

0.0.2. **Remark (Why do we study categories?).**

(1) Abundance: Categories abound in mathematics and in related fields such as computer science. Categories are formed e.g. by
   (a) Vector spaces,
   (b) Abelian groups,
   (c) groups,
   (d) sets,
   (e) monoids,
   (f) semigroups,
   (g) topological spaces,
   (h) Banach spaces,
   (i) manifolds,
   (j) ordered sets,
   (k) graphs,
   (l) (computer-)languages,
   (m) automata.
(2) Insight into similar constructions:
   (a) products,
   (b) free objects,
   (c) direct sums,
   (d) kernel and cokernels,
   (e) limits,
   (f) associated groups and algebras, such as homology groups, Grothendieck rings, fundamental groups, associative envelopes, compactifications, completions.
(3) Category theory proves that all information about a mathematical object can also be drawn from the knowledge of all structure preserving maps into this object. The knowledge of the maps is equivalent to the knowledge of the interior structure of an object. "Functions are everywhere!"
(4) Category theory is a language that allows to express similar properties and phenomena that occur in different mathematical areas. It helps to recognize certain properties and notions to have a general description and to be "categorical". E.g:

(a) each finite dimensional vector space is isomorphic to its dual and hence also to its double dual. The second correspondence is considered "natural" and the first is not. Category theory can explain what the notion of "natural" really means. This was in fact the starting point of category theory.
(b) Topological spaces can be defined in many different ways, e.g. via open sets, via closed sets, via neighborhoods, via convergent filters, and via closure operations. Why do these definitions describe "essentially the same" objects? Category theory provides an answer via the notion of *concrete isomorphism*.
(c) Initial structures, final structures, and factorization structures occur in many different situations. Category theory allows one to formulate and investigate such concepts with an appropriate degree of generality.

(5) Category theory offers many convenient symbols that allow one to quickly perform the necessary calculations:
(a) commutative diagrams,
(b) braid diagrams,
(c) computations with symbolic elements.

(6) Transportation of problems from one area of mathematics (via suitable functors) to another area, where solutions are sometimes easier and then a transport back to the original area (sometimes via adjoint functors). For example, algebraic topology can be described as an investigation of topological problems (via suitable functors) by algebraic methods, such as associated homotopy groups.

(7) Duality: The concept of category is well-balanced, which allows an economical and useful *duality*.Thus is category theory the "two for the price of one" principle holds: every concept is two concepts, and every result is two results.

(8) Category theory provides the means to distinguish between general problems ("categorical" problems that may occur in many different areas in similar form) and specific problems (that a closely linked to the very special area and object one is studying).

0.0.3. **Remark (Quotes).** (1) (Steenrod) "Category theory is generalized abstract nonsense."

(2) (Hoare) "Category theory is quite the most general and abstract branch of pure mathematics. The corollary of a high degree of generality and abstraction is that the theory gives almost no assistance in solving the more specific problems within any of the subdisciplines to which it applies. It is a tool for the generalist, of little benefit for the practitioner."

(3) (Barr-Wells) "'Categories originally arose in mathematics out of the need of a formalism to describe the passage from one type of mathematical structure to another. A category in this way represents a kind of mathematics, and may be described as *category as mathematical workspace.*

A category is also a *mathematical structure*. As such, it is a common generalization of both ordered sets and monoids (the latter are a simple type of algebraic structure that include transition systems as examples), and questions motivated by those topics often have interesting answers for categories. This is category as mathematical structure.

Finally, a category can be seen as a structure that formalizes a mathematician's description of a type of structure. This is the role of category as theory. Formal descriptions in mathematical logic are traditionally given as formal languages with rules for forming terms, axioms and equations. Algebraists long ago invented a formalism based on tuples, the method of signatures and equations, to describe algebraic structures. Category theory provides another approach: *the category is a theory and functors with that category as domain are models of the theory.*"'

(4) (Stanford Encyclopedia of Philosophy) "Categories, functors, natural transformations, limits and colimits appeared almost out of nowhere in 1945 in Eilenberg & Mac Lane's paper entitled "General Theory of Natural Equivalences". We said "almost", because when one looks at their 1942 paper "Group Extensions and Homology", one discovers specific functors and natural transformations at work, limited to groups. In fact, it was basically the need to clarify and abstract from their 1942 results that Eilenberg & Mac Lane came up with the notions of category theory. The central notion for them, as the title indicates, was the notion of natural transformation. In order to give a general definition of the latter, they defined the notion of functor, borrowing the terminology from Carnap, and in order to give a general definition of functor, they defined the notion of category, borrowing this time from Kant and Aristotle."

1. Foundations

1.1. **Graphs.**

1.1.1. **Definition.** A *(directed) graph* (*digraph*) is a monad

$$\mathcal{G} := (V, E, \pi)$$

consisting of the set V of *vertices*, the set E of *edges* or *arrows* and the map to the endpoints $\pi : E \longrightarrow V \times V$. For each edge $f \in E$ the vertices A and B with $\pi(f) = (A, B)$ are called *source and target* or *domain and codomain* of f. We write the edge f also as $f : A \longrightarrow B$.
Later on we will prefer the notation $\mathcal{G}_0$ for the set V of vertices and $\mathcal{G}_1$ for the set E of edges.
There may be one or more arrows – or none at all – with given nodes as source and target.
Moreover, the source and target of a given arrow need not be distinct. An arrow with the same source and target node will be called an *endoarrow* or *endomorphism* of that node.
The set of arrows with source A and target B will be denoted by $\mathcal{G}(A, B)$ or $\mathrm{Mor}_{\mathcal{G}}(A, B)$.
A graph $\mathcal{G}$ is called *simple*, if π is injektive. (In the literature one also requires that there are no endoarrows.)

1.1.2. **Definition.** Let $\mathcal{G} = (V, E, \pi)$ and $\mathcal{H} = (V', E', \pi')$ be directed graphs. A *homomorphism* $\mathcal{F} : \mathcal{G} \longrightarrow \mathcal{H}$ consists of two maps

$$\mathcal{F}_V : V \longrightarrow V', \mathcal{F}_E : E \longrightarrow E'$$

(or

$$\mathcal{F}_0 : \mathcal{G}_0 \longrightarrow \mathcal{H}_0, \quad \mathcal{F}_1 : \mathcal{G}_1 \longrightarrow \mathcal{H}_1)$$

with

$$\forall f \in E[\pi(f) = (A, B) \Longrightarrow \pi'\mathcal{F}_E(f) = (\mathcal{F}_V(A), \mathcal{F}_V(B))].$$

Thus homomorphisms send vertices to vertices and arrows to arrows in such a way that incidence and the direction of arrows are preserved.
In particular we get for every pair $(A, B) \in V \times V$ a map

$$\mathcal{F}_E : \mathcal{G}(A, B) \longrightarrow \mathcal{H}(\mathcal{F}_V(A), \mathcal{F}_V(B)).$$

An *isomorphism* $\mathcal{F} : \mathcal{G} \longrightarrow \mathcal{H}$ is a homomorphism, for which there is a second homomorphism $\overline{\mathcal{F}} : \mathcal{H} \longrightarrow \mathcal{G}$ such that $\overline{\mathcal{F}}\mathcal{F} = \mathrm{id}_{\mathcal{G}}$ and $\mathcal{F}\overline{\mathcal{F}} = \mathrm{id}_{\mathcal{H}}$. Two graphs are called *isomorphic*, if there exists an isomorphism between them.

1.1.3. **Examples.** (1) Let $V_{\mathcal{G}} := \{1, 2\}$ and $E_{\mathcal{G}} := \{a, b, c\}$,

$$\mathrm{source}(a) = \mathrm{target}(a) = \mathrm{source}(b) = \mathrm{target}(c) = 1,$$

and

$$\mathrm{target}(b) = \mathrm{source}(c) = 2.$$

Then we can represent $\mathcal{G}$ as

$$\begin{tikzcd} 1 \arrow[loop above, "a"] \arrow[r, shift left, "b"] & 2 \arrow[l, shift left, "c"] \end{tikzcd}$$

(2) The graph of sets and functions has all sets as nodes and all functions between sets as arrows. The source of a function is its domain, and its target is its codomain. This is a large graph: because of Russell's paradox, its nodes and its arrows do not form sets.

(3) It is often convenient to picture a relation on a set as a graph. For example, let $A = \{1, 2, 3, 4\}$ and $\alpha = \{(1, 2), (2, 2), (2, 3), (1, 4)\}$. Then α can be pictured as

$$\begin{tikzcd} 1 \arrow[r] \arrow[d] & 2 \arrow[r] \arrow[loop above] & 3 \\ 4 & & \end{tikzcd}$$
22.04.04

(4) A data structure can sometimes be represented by a graph. The following graph represents the set $\mathbb{N}$ of natural numbers in terms of zero and the successor function (adding 1):

$$\begin{tikzcd} 1 \arrow[r, "0"] & n \arrow[loop above, "\text{succ}"] \end{tikzcd}$$

The name '1' for the left node is the conventional notation to require that the node denote a singleton set, that is, a set with exactly one element. One can give a formal mathematical meaning to the idea that this graph generates the natural numbers.

1.1.4. **Remark. What are the natural numbers?**

(1) Traditional, set-theoretic answer (Peano, one century ago):
The natural numbers form a set $\mathbb{N}$ such that:
 (a) $\exists$ zero $\in \mathbb{N}$,
 (b) $\forall n \in \mathbb{N}, \exists\ \mathrm{succ}(n) \in \mathbb{N}$,
 (c) $\forall n \in \mathbb{N} :\ \mathrm{succ}(n) \neq\ \mathrm{zero} \in \mathbb{N}$
 (d) $\forall m, n \in \mathbb{N} :\ \mathrm{succ}(m) =\ \mathrm{succ}(n) \Longrightarrow m = n$ (injectivity),
 (e) $\forall A \subseteq \mathbb{N} : (\ \mathrm{zero} \in A \wedge (\forall a \in A :\ \mathrm{succ}(a) \in A)) \Longrightarrow A = \mathbb{N}$.
These axioms determine $\mathbb{N}$ uniquely up to isomorphism.

(2) Categorical answer (Lawvere, 60's): A natural number object

$$0 \in \mathbb{N} \xrightarrow{s} \mathbb{N}$$

(in **Set**) consists of
- a set $\mathbb{N}$
- with a distinguished element $0 \in \mathbb{N}$
- and an endofunction $s : \mathbb{N} \longrightarrow \mathbb{N}$

which is *universal* in the sense that for every structure

$$e \in X \xrightarrow{g} X$$

there exists a unique function $f : \mathbb{N} \longrightarrow X$ such that
- $f(0) = e$
- $f(s(n)) = g(f(n))$ for all $n \in \mathbb{N}$.

The two characterizations are equivalent. This is a consequence of the theorem on simple recursion [Pareigis, 2000] Theorem 2.2.1 and the uniqueness given by the universal property [Pareigis, 2000] Theorem 2.2.3.

1.1.5. **Remark.** Notation of the form $f : A \longrightarrow B$ is overloaded in several ways. It can denote a set-theoretic function, a graph homomorphism or an arrow in a graph. In fact, all three are instances of the third since there is a large graph whose nodes are sets and arrows are functions and another whose nodes are (small) graphs and arrows are graph homomorphisms. Another form of overloading is that if $\mathcal{F} : \mathcal{G} \longrightarrow \mathcal{H}$ is a graph homomorphism, ϕ actually stands for a pair of functions which we call $\mathcal{F}_V : V_\mathcal{G} \longrightarrow V_\mathcal{H}$ and $\mathcal{F}_E : E_\mathcal{G} \longrightarrow E_\mathcal{H}$. In fact, it is customary to omit the subscripts and use $\mathcal{F}$ for all three (the graph homomorphism as well as its components $\mathcal{F}_V$ and $\mathcal{F}_E$). This does not lead to ambiguity in practice; in reading about graphs you are nearly always aware of whether the author is talking about nodes or arrows.

1.1.6. **Example.** (1) If $\mathcal{G}$ is any graph, the *identity homomorphism* $\mathrm{id}_\mathcal{G} : \mathcal{G} \longrightarrow G$ is defined by $(\mathrm{id}_\mathcal{G})_V = \mathrm{id}_{V_\mathcal{G}}$ (the identity function on the set of nodes of $\mathcal{G}$) and $(\mathrm{id}_\mathcal{G})_E = \mathrm{id}_{E_\mathcal{G}}$.

(2) If $\mathcal{G}$ is the graph

$$\begin{tikzcd}
1 \arrow[r] \arrow[d] & 2 \arrow[r] \arrow[loop above] & 3 \\
4 & &
\end{tikzcd}$$

and $\mathcal{H}$ is the graph

$$\begin{tikzcd}
S \arrow[r] \arrow[d] & F \arrow[loop above] \arrow[loop below] \\
Q &
\end{tikzcd}$$

then there is a homomorphism $\mathcal{F} : \mathcal{G} \longrightarrow \mathcal{H}$ for which $\mathcal{F}_V(1) = S, \mathcal{F}_V(2) = \mathcal{F}_V(3) = F$ and $\mathcal{F}_V(4) = Q$, and $\mathcal{F}_E$ takes the loop on 2 and the arrow from 2 to 3 both to the upper loop on F; what $\mathcal{F}_E$ does to the other two arrows is forced by the definition of homomorphism. Because there are two loops on F there are actually four possibilities for $\mathcal{F}_E$ on arrows (while keeping $\mathcal{F}_V$ fixed).

(3) If $\mathcal{H}$ is any graph with a node n and a loop $u : n \longrightarrow n$, then there is a homomorphism from any graph $\mathcal{G}$ to $\mathcal{H}$ that takes every node of $\mathcal{G}$ to n and every arrow to u. This construction gives two other homomorphisms from $\mathcal{G}$ to $\mathcal{H}$ in Example 1.5 (2) besides the four mentioned there. (There are still others.)

1.2. **Monoids.**

1.2.1. **Definition.** A *monoid* is a monad $\mathcal{M} = (M, \mu, e)$ consisting of

(1) a set M,
(2) a *multiplication* map $\mu : M \times M \longrightarrow M$, and
(3) a *unit* or *neutral* element $e \in M$

satisfying the following axioms

- $\forall a, b, c \in M$:

$$a \circ (b \circ c) = (a \circ b) \circ c, \text{ (associativity)}$$

- $\forall a, b \in M$:

$$e \circ a = a \text{ and } b \circ e = b. \text{ (left and right unit element)}$$

Here we use the notation $a \circ b := \mu(a, b)$.

1.2.2. **Definition.** Let $\mathcal{M} = (M, \mu, e)$ and $\mathcal{M}' = (M', \mu', e')$ be monoids. A *homomorphism of monoids* $\mathcal{F} : \mathcal{M} \longrightarrow \mathcal{M}'$ is a map $\mathcal{F} : M \longrightarrow M'$ satisfying the following axioms

- $\forall a, b \in M : \mathcal{F}(a \circ b) = \mathcal{F}(a) \circ \mathcal{F}(b)$,

- $\mathcal{F}(e) = e'$.

An *isomorphism* $\mathcal{F} : \mathcal{M} \longrightarrow \mathcal{M}'$ is a homomorphism such that there exists another homomorphism $\overline{\mathcal{F}} : \mathcal{M}' \longrightarrow \mathcal{M}$ with $\overline{\mathcal{F}}\mathcal{F} = \mathrm{id}_{\mathcal{M}}$ and $\mathcal{F}\overline{\mathcal{F}} = \mathrm{id}_{\mathcal{M}'}$. Two monoids are called *isomorphic* if there exists an isomorphism from one monoid to the other.

23.04.04

1.2.3. **Definition.** Let $\mathcal{M}$ be a monoid. An $\mathcal{M}$-*graded monoid* is a monad $\mathcal{U} = ((U_i), (\mu_{i,j}), 1)$ consisting of

(1) a family of mutually disjoint sets $(U_i | i \in M)$,
(2) a family of (*multiplication*) maps $(\mu_{i,j} : U_i \times U_j \longrightarrow U_{i \cdot j} | i, j \in M)$, and
(3) an element $1 \in U_e$

satisfying the following axioms (using the notation $g \circ f := \mu_{i,j}(f, g)$):

- $\forall i, j, k \in M \; \forall f \in U_i, g \in U_j, h \in U_k$:
$$h \circ (g \circ f) = (h \circ g) \circ f \in U_{(i \cdot j) \cdot k} = U_{i \cdot (j \cdot k)},$$
- $\forall i, j \in M \; \forall f \in U_i, g \in U_j$:
$$1 \circ f = f \text{ and } g \circ 1 = g.$$

1.2.4. **Definition.** Let $\mathcal{M}$ and $\mathcal{M}'$ be monoids, $\mathcal{U}$ be an $\mathcal{M}$-graded monoid, and $\mathcal{U}'$ be an $\mathcal{M}'$-graded monoid. A *homomorphism of graded monoids* $\mathcal{F} : (\mathcal{M},\mathcal{U}) \longrightarrow (\mathcal{M}',\mathcal{U}')$ consists of

(1) a homomorphism of monoids $\mathcal{F}_M : \mathcal{M} \longrightarrow \mathcal{M}'$ and
(2) maps $\mathcal{F}_i : U_i \longrightarrow U'_{\mathcal{F}_M(i)}$ for all $i \in M$

such that the following conditions are satisfied

- $\forall i, j \in M, f \in U_i, g \in U_j$:
$$\mathcal{F}_{i \cdot j}(g \circ f) = \mathcal{F}_j(g) \circ \mathcal{F}_i(f)$$
- $\mathcal{F}_e(1) = 1 \in U'_{e'}$.

It follows from the definition that a monoid is not allowed to be empty: it must contain an identity element. It also follows that we can extend the notation of powers to 0 by defining x^0 to be the identity element of the monoid. The laws $s^k s^n = s^{k+n}$ and $(s^k)^n = s^{kn}$ then hold for all nonnegative k and n.

1.2.5. **Examples.** One example of a monoid is the set of non-negative integers with addition as the operation. This includes 0 as a neutral element.

The *Kleene closure* A^* of a set A is the set of strings (or lists) of finite length of elements of A. We write the lists in parentheses; for example (a, b, d, a) is an element of $\{a, b, c, d\}^*$. Some parts of the computer science literature call these strings instead of lists and write them this way: '*abda*'. A^* includes the empty list () and for each element $a \in A$ the list (a) of length one.

The operation of concatenation makes the Kleene closure a monoid $F(A)$, called the *free monoid* determined by A. The empty list is the identity element. We write concatenation as juxtaposition; thus
$$(a, b, d, a)(c, a, b) = (a, b, d, a, c, a, b).$$
Note that the underlying set of the free monoid is A^*, not A. In the literature, A is usually assumed finite, but the Kleene closure is defined for any set A. The elements of A^* are lists of finite length in any case. When A is nonempty, A^* is an infinite set.

The concept of freeness is a general concept applied to many kinds of structures. It is treated systematically later on.

1.2.6. **Definition.** A *submonoid* of a monoid $\mathcal{M}$ is a subset S of M with the properties:

SM-1 The identity element of $\mathcal{M}$ is in S.

SM-2 If $m, n \in S$ then $mn \in S$. (One says that S is *closed under the operation.*)

1.2.7. **Example.** The natural numbers with addition form a submonoid of the integers with addition. For another example, consider the integers with multiplication as the operation, so that 1 is the identity element. Again the natural numbers form a submonoid, and so does the set of positive natural numbers, since the product of two positive numbers is another one. Finally, the singleton set $\{0\}$ is a subset of the integers that is closed under multiplication, and it is a monoid, but it is not a submonoid of the integers on multiplication because it does not contain the identity element 1.

In this section, we discuss categories whose objects are semigroups or monoids. These are typical of categories of algebraic structures; we have concentrated on semigroups and monoids because transition systems naturally form monoids.

1.2.8. **Definition.** A *semigroup* is a set S together with an associative binary operation $m : S \times S \longrightarrow S$. The set S is called the *underlying set* of the semigroup.

1.2.9. **Remark.** Normally for s and t in S, $m(s,t)$ is written 'st' and called '*multiplication*', but note that it does not have to satisfy the commutative law; that is, we may have $st \neq ts$. A *commutative semigroup* is a semigroup whose multiplication is commutative.
It is standard practice to talk about '*the semigroup* S', naming the semigroup by naming its underlying set. This will be done for other mathematical structures such as posets as well. Mathematicians call this practice '*abuse of notation*'. It is occasionally necessary to be more precise.
We set $s^1 = s$ and, for any positive integer k, $s^k = ss^{k-1}$. Such *powers* of an element obey the laws $s^k s^n = s^{k+n}$ and $\left(s^k\right)^n = s^{kn}$ (for positive k and n). On the other hand, the law $(st)^k = s^k t^k$ requires commutativity.
We specifically allow the *empty semigroup*, which consists of the empty set and the empty function from the empty set to itself. (Note that the cartesian product of the empty set with itself is the empty set.) This is not done in most of the non-category theory literature; it will become evident later why we should include the empty semigroup.
An *identity element* e for a semigroup S is an element of S that satisfies the equation $se = es = s$ for all $s \in S$. There can be at most one identity element in a semigroup.

1.2.10. **Definition (Homomorphisms of semigroups).** If S and T are semigroups, a function $h : S \longrightarrow T$ is a *homomorphism* if for all $s, s' \in S, h(ss') = h(s)h(s')$.

1.2.11. **Examples.** The identity function on any monoid is a monoid homomorphism. If M is a monoid and S is a submonoid, the inclusion function from S to M is a monoid homomorphism. Another example is the function that takes an even integer to 0 and an odd integer to 1. This is a monoid homomorphism from the monoid of integers on multiplication to the set $\{0, 1\}$ on multiplication. It is easy to see that identity functions are homomorphisms and homomorphisms compose to give homomorphisms. Thus we have two categories: **Sem** is the category of semigroups and semigroup homomorphisms, and **Mon** is the category of monoids and monoid homomorphisms.

1.2.12. **Example.** Let S be a semigroup with element s. Let $\mathbb{N}^+$ denote the semigroup of positive integers with addition as operation. There is a semigroup homomorphism $p : \mathbb{N}^+ \longrightarrow S$ for which $p(k) = s^k$. That this is a homomorphism is just the statement $s^{k+n} = s^k s^n$.

1.2.13. **Remark.** A semigroup homomorphism between monoids need not preserve the identities. An example of this involves the trivial monoid E with only one element e (which is perforce the identity element) and the monoid of all integers with multiplication as the operation, which is a monoid with identity 1. The function that takes the one element of E to 0 is a semigroup homomorphism that is not a monoid homomorphism. And, by the way, even though $\{0\}$ is a subsemigroup of the integers with multiplication and even though it is actually a monoid, it is not a submonoid.

1.2.14. **Remark (Inverses of homomorphisms).** As an example of how to use the definition of homomorphism, we show that the inverse of a bijective semigroup homomorphism is also a semigroup homomorphism. Let $f : S \longrightarrow T$ be a bijective semigroup homomorphism with inverse g. Let $t, t' \in T$. We have to show that $g(t)g(t') = g(tt')$. Since f is injective, it is sufficient to show that $f(g(t)g(t')) = f(g(tt'))$.
The right hand side is tt' because g is the inverse of f, and the left hand side is $f(g(t)g(t')) = f(g(t))f(g(t'))$ because f is a homomorphism, but that is also tt' since g is the inverse of f. This sort of theorem is true of other algebraic structures, such as monoids. It is not true for posets.

1.2.15. **Remark (Isomorphisms of semigroups).** If a homomorphism of semigroups has an inverse that is a homomorphism (equivalently, as we just saw, if it is bijective), we say that the homomorphism is an isomorphism. In this case, the two semigroups in question have the same abstract structure and are said to be *isomorphic*. As we will see later, the property of possessing an inverse is taken to define the categorical notion of isomorphism.
It is important to understand that there may in general be many different isomorphisms between isomorphic semigroups. For example, there are two distinct isomorphisms between the monoid with underlying set $\{0, 1, 2, 3\}$ and addition (mod 4) as operation and the monoid with underlying set $\{1, 2, 3, 4\}$ and multiplication (mod 5) as operation.

We now discuss two important types of examples of monoid homomorphisms that will reappear later in the notes. The first example is a basic property of free monoids.

1.2.16. **Remark (Kleene closure induces homomorphisms).** Let A and B denote sets, thought of as alphabets. Let $f : A \longrightarrow B$ be any set function. We define $f^* : A^* \longrightarrow B^*$ by $f^*((a_1, a_2, \ldots, a_k)) = (f(a_1), f(a_2), \ldots, f(a_k))$. In particular, $f^*() = ()$ and for any $a \in A$, $f^*(a) = f(a)$. Then f^* is a homomorphism of monoids, a requirement that, in this case, means it preserves identity elements (by definition) and concatenation, which can be seen from the following calculation: Let $a = (a_1, a_2, \ldots, a_m)$ and $a' = (a'_1, a'_2, \ldots, a'_n)$ be lists in A^*. Concatenating them gives the list

$$aa' = (a_1, a_2, \ldots, a_m, a'_1, a'_2, \ldots, a'_n).$$

Then

$$\begin{aligned}
f^*(a)f^*(a') &= f^*(a_1, a_2, \ldots, a_m)f^*(a'_1, a'_2, \ldots, a'_n) \\
&= (f(a_1), f(a_2), \ldots, f(a_m))(f(a'_1), f(a'_2), \ldots, f(a'_n)) \\
&= (f(a_1), f(a_2), \ldots, f(a_m), f(a'_1), f(a'_2), \ldots, f(a'_n)) \\
&= f^*(a_1, a_2, \ldots, a_m, a'_1, a'_2, \ldots, a'_n) \\
&= f^*(aa')
\end{aligned}$$

Thus any set function between sets induces a monoid homomorphism between the corresponding free monoids. The reader may wish to verify that if f is an isomorphism then so is f^*. The function f^* is called αf in [Backus, 1981a] and in modern functional languages is usually called map f or map list f.

The other important example is a basic construction of number theory.

1.2.17. **Remark** (**The remainder function).** The set $\mathbb{Z}$ of all integers forms a monoid with respect to either addition or multiplication. If k is any positive integer, the set $Z_k = \{0, 1, \ldots, k-1\}$ of remainders of k is also a monoid with respect to addition or multiplication (mod k). Here are more precise definitions.

1.2.18. **Definition.** Let k be a positive integer and n any integer. Then $n \mod k$ is the unique integer $r \in Z_k$ for which there is an integer q such that $n = qk + r$ and $0 \leq r < k$.

It is not difficult to see that there is indeed a unique integer r with these properties. Define an operation '$+_k$' of addition (mod k) by requiring that $r +_k s = (r + s) \mod k$.
The operation of addition of the contents of two registers in a microprocessor may be addition (mod k) for k some power of 2 (often complicated by the presence of sign bits).

1.2.19. **Proposition.** $(\mathbb{Z}_k, +_k)$ *is a monoid with identity 0.*

We also have the following.

1.2.20. **Proposition.** *The function* $n \mapsto (n \mod k)$ *is a monoid homomorphism from* $(\mathbb{Z}, +)$ *to* $(Z_k, +_k)$*: A similar definition and proposition can be given for multiplication.*

29.04.04

1.3. **Categories.**

1.3.1. **Definition.** A *category* is a quadruple $\mathcal{C} = (\mathrm{Ob}(\mathcal{C}), \mathrm{Mor}(\mathcal{C}), (\mu), (1))$ consisting of

(1) a class of *objects*

$$A, B, C \in \mathrm{Ob}(\mathcal{C}),$$

(2) a family of mutually disjoint sets

$$(\mathrm{Mor}_{\mathcal{C}}(A, B) | A, B \in \mathrm{Ob}(\mathcal{C}))$$

whose elements $f, g, h \ldots \in \mathrm{Mor}_{\mathcal{C}}(A, B)$ are called *morphisms*,

(3) a family of (*composition*) maps

$$(\mu_{A,B,C} : \mathrm{Mor}_{\mathcal{C}}(A, B) \times \mathrm{Mor}_{\mathcal{C}}(B, C) \ni (f, g) \mapsto gf \in \mathrm{Mor}_{\mathcal{C}}(A, C) | A, B, C \in \mathrm{Ob}(\mathcal{C})),$$

and

(4) a family of elements called *identities*:

$$(1_A \in \mathrm{Mor}_{\mathcal{C}}(A, A) | A \in \mathrm{Ob}(\mathcal{C}))$$

satisfying the following axioms

- $\forall A, B, C, D \in \mathrm{Ob}(\mathcal{C})\ \forall f \in \mathrm{Mor}_C(A, B), g \in \mathrm{Mor}_{\mathcal{C}}(B, C),$ and $h \in \mathrm{Mor}_{\mathcal{C}}(C, D)$:
$$h(gf) = (hg)f \in \mathrm{Mor}_{\mathcal{C}}(A, D),$$
- $\forall A, B, C \in \mathrm{Ob}(\mathcal{C}) \forall f \in \mathrm{Mor}_{\mathcal{C}}(A, B), g \in \mathrm{Mor}_{\mathcal{C}}(B, C)$:
$$1_B f = f \text{ and } g 1_B = g.$$

1.3.2. **Remark. Alternative definition (Barr-Wells):**
Let $k > 0$. In a graph $\mathcal{G}$, a *path* from a node A to a node B of length k is a sequence $(f_1, f_2, \ldots, f_k; A, B)$ of (not necessarily distinct) arrows for which

(1) $\text{source}(f_k) = A$,
(2) $\text{target}(f_i) = \text{source}(f_{i-1})$ for $i = 2, \ldots, k$, and
(3) $\text{target}(f_1) = B$.

By convention, for each node A there is a unique path of length 0 from A to A that is denoted (; A, A). It is called the *empty path* at A.
Observe that if you draw a path as follows:

$$\cdot \xrightarrow{f_k} \cdot \xrightarrow{f_{k-1}} \cdots \xrightarrow{f_2} \cdot \xrightarrow{f_1} \cdot$$

with the arrows going from left to right, f_k will be on the left and the subscripts will go down from left to right. We do it this way for consistency with composition.
For any arrow $f : A \longrightarrow B$, $(f; A, B)$ is a path of length 1.
Definition: The set of paths of length k in a graph $\mathcal{G}$ is denoted $\mathcal{G}_k$.
In particular, $\mathcal{G}_2$, which will be used in the definition of category, is the set of pairs of arrows $(f, g; \text{source}(g), \text{target}(f))$ for which the target of g is the source of f. These are called *composable pairs* of arrows: $\cdot \xrightarrow{g} \cdot \xrightarrow{f} \cdot$.
We have now assigned two meanings to $\mathcal{G}_0$ and $\mathcal{G}_1$. This will cause no conflict as $\mathcal{G}_1$ refers indifferently either to the collection of arrows of $\mathcal{G}$ or to the collection of paths of length 1, which is essentially the same thing. Similarly, we use $\mathcal{G}_0$ to represent either the collection of nodes of $\mathcal{G}$ or the collection of empty paths, of which there is one for each node. In each case we are using the same name for two collections that are not the same but are in a natural one to one correspondence. Compare the use of '2' to denote either the integer or the real number. As this last remark suggests, one might want to keep the two meanings of $\mathcal{G}_1$ separate for purposes of implementing a graph as a data structure.
The one to one correspondences mentioned in the preceding paragraph were called '*natural*'. The word is used informally here, but in fact these correspondences are natural in the technical sense.
Definition: A *category* is a graph $\mathcal{C}$ together with two functions $c : \mathcal{C}_2 \longrightarrow C_1$ and $u : C_0 \longrightarrow C_1$ with properties C-1 through C-4 below. (Recall that $\mathcal{C}_2$ is the set of paths of length 2.) The elements of $\mathcal{C}_0$ are called *objects* and those of $\mathcal{C}_1$ are called *arrows*. The function c is called *composition*, and if $(f, g; A, C)$ is a composable pair, $c(f, g; A, C)$ is written $gf = g \circ f$ and is called the *composite* of f and g. If A is an object of $\mathcal{C}$, $u(A)$ is denoted 1_A, which is called the *identity* of the object A.

C-1 The source of gf is the source of f and the target of gf is the target of g.
C-2 $(hg)f = h(gf)$ whenever either side is defined.
C-3 The source and target of 1_A are both A.
C-4 If $f : A \longrightarrow B$, then $f1_A = 1_B f = f$.

The significance of the fact that the composite c is defined on $\mathcal{G}_2$ is that gf is defined if and only if the source of g is the target of f. This means that composition is a function whose domain is an equationally defined subset of $\mathcal{G}_1 \times \mathcal{G}_1$: the equation requires that the source of g equal the target of f. It follows from this and C-1 that in C-2, one side of the equation is defined if and only if the other side is defined.
In the category theory literature, 1_A is often written just A.

1.3.3. **Remark** (Terminology)**.** In much of the categorical literature, '*morphism*' is more common than '*arrow*'. We will normally denote objects of categories by capital letters but nodes of graphs (except when we think of a category as a graph) by lower case letters. Arrows are always lower case.
In the computing science literature, the composite $g \circ f$ is sometimes written f; g , a notation suggested by the perception of a typed functional programming language as a category.
We have presented the concept of category as a two-sorted data structure; the sorts are the objects and the arrows. Categories are sometimes presented as one-sorted – arrows only. The objects can be recovered from the fact that C-3 and C-4 together characterize 1_A (not hard to prove), so that there is a one to one correspondence between the objects and the identity arrows 1_A.

1.3.4. **Definition.** A category is *small* if its objects and arrows constitute sets; otherwise it is *large*.
A category with the property that $\mathrm{Mor}_{\mathcal{C}}(A, B)$ is a set for all objects A and B is called *locally small*. All categories in these notes are locally small.

1.3.5. **Remark.** The category of sets and functions is an example of a large category. Although one must in principle be wary in dealing with large classes, it is not in practice a problem; category theorists have rarely, if ever, run into set theoretic difficulties.

1.3.6. **Definition.** For any path $(f_1, f_2, \ldots, f_n)$ in a category $\mathcal{C}$, define $f_1 \circ f_2 \circ \ldots \circ f_n$ recursively by

$$f_1 \circ f_2 \circ \ldots \circ f_n := (f_1 \circ f_2 \circ \ldots \circ f_{n-1}) \circ f_n, \quad n > 2.$$

1.3.7. **Proposition** (The general associative law.)**.** *For any path $(f_1, f_2, \ldots, f_n)$ in a category $\mathcal{C}$ and any integer k with $1 < k < n$,*

$$(f_1 \circ \ldots \circ f_k) \circ (f_{k+1} \circ \ldots \circ f_n) = f_1 \circ \ldots \circ f_n.$$

In other words, you can unambiguously drop the parentheses.

In this proposition, the notation $f_{k+1} \circ \ldots \circ f_n$ when $k = n - 1$ means simply f_n.
This is a standard fact for associative binary operations (see [Jacobson, 1974], Section 1.4) and can be proved in exactly the same way for categories.

1.3.8. **Examples.** (1) Degenerate categories such as:

- Sets X – i.e. X is the class of objects and all arrows are identities.
- Monoids $\mathcal{M}$ – any monoid $\mathcal{M}$ determines a category $\mathcal{C}(\mathcal{M})$.
  - CM-1 $\mathcal{C}(\mathcal{M})$ has one object, which we will denote $*$, $*$ can be chosen arbitrarily. A simple uniform choice is to take $* = M$.
  - CM-2 The arrows of $\mathcal{C}(\mathcal{M})$ are the elements of M with $*$ as source and target.
  - CM-3 Composition is the binary operation of $\mathcal{M}$.

  Thus a category can be regarded as a generalized monoid, or a '*monoid with many objects*'. This point of view has not been as fruitful in mathematics as the perception of a category as a generalized poset. It does have some applications in computing science.
- Preorders – any preorder $(P, \leq)$ determines a category $\mathcal{C}(P)$. The class of objects of $\mathcal{C}(P)$ is the set P. There is exactly one arrow between $x \in P$ and $y \in P$ if and only if $x \leq y$ (otherwise there is no arrow). So there is at most one arrow between any two objects (see 1.3.8(11)).

Many categorists define a monoid to be a category with one object and a preordered set to be a category in which every hom set is either empty or a singleton.

(2) Categories determined by graphs:
- **0** – the category with empty class of objects (and no morphisms).
- **1** – the category with exactly one object and exactly one morphism (the identity of the object).
- **2** $= (\cdot \longrightarrow \cdot)$ – the category with two objects and one arrow between them. Together with the identity arrows there are 3 arrows.
  This can be represented by the category $\mathcal{C}(S,\alpha)$ for $S = \{C, D\}$ and $\alpha = \{\langle C,C\rangle, \langle C,D\rangle, \langle D,D\rangle\}$.
- ω – the category with a countably infinite number of objects and ascending arrows (succ) between them.
- Categories freely generated by directed graphs $\mathcal{G}$.

(3) The opposite category $\mathcal{C}^{op}$ obtained by reversing the arrows of a given category $\mathcal{C}$, while keeping the same objects.

(4) **Set** – the category of (small) sets and functions; composition is the usual function composition (as well as in the remaining examples). Note that type matters: the identity on the natural numbers is a different function from the inclusion of the natural numbers into the integers.

(5) **Set**$_*$ – pointed sets (i.e. sets with a selected base-point) and functions preserving the base point.

(6) A *partial function* from a set S to a set T is a function with domain S_0 and codomain T, where S_0 is some subset of S. The category **Pfn** of sets and partial functions has all sets as objects and all partial functions as arrows. If $f : S \longrightarrow T$ and $g : T \longrightarrow V$ are partial functions with f defined on $S_0 \subseteq S$ and g defined on $T_0 \subseteq T$, the composite $g \circ f : S \longrightarrow V$ is the partial function from S to V defined on the subset $S_0 \cap f^{-1}(T_0)$ of S by the requirement $(g \circ f)(x) = g(f(x))$.

(7) $\mathbf{\Delta}$ – finite ordinals (the ordinal n *is* the totally ordered set of all the preceding ordinals $\{0, 1, \ldots, n-1\}$ and 0 is the empty set) and monotone or order preserving functions. This category is called the *simplicial category*.

(8) **Fin** – finite sets and functions. 30.4.04

(9) Let α be a relation from a set S to a set T and β a relation from T to U. The *composite* $\beta \circ \alpha$ is the relation from S to U defined as follows: If $x \in S$ and $z \in U$ then $(x, z) \in \beta \circ \alpha$ if and only if there is an element $y \in T$ for which $(x, y) \in \alpha$ and $(y, z) \in \beta$. With this definition of composition, the *category* **Rel** *of sets and relations* has sets as objects and relations as arrows. The identity for a set S is the diagonal relation $\Delta_S = \{(x, x) | x \in A\}$.

   In order to make the morphism sets disjoint, one has to take monads (α, S, T) for the morphisms, where $\alpha \subseteq S \times T$.

(10) Other examples of categories whose objects are sets are the category of sets and injective functions and the category of sets and surjective functions.

(11) Preordered sets: If S is a set, a subset $\alpha \subset S \times S$ is called a *binary relation* on S. It is often convenient to write $x\alpha y$ as shorthand for $(x, y) \in \alpha$. We say that α is *reflexive* if $x\alpha x$ for all $x \in S$ and *transitive* if $x\alpha y$ and $y\alpha z$ implies $x\alpha z$ for all $x, y, z \in S$.

   A set S with a reflexive, transitive relation α on it is a structure $(S; \alpha)$, i.e. a preordered set determines a category $\mathcal{C}(S; \alpha)$ defined as follows.

CO-1 The objects of $\mathcal{C}(S,\alpha)$ are the elements of S.

CO-2 If $x, y \in S$ and $x\alpha y$, then $\mathcal{C}(S,\alpha)$ has exactly one arrow from x to y, denoted (y,x). (The reader might have expected (x,y) here. This choice of notation fits better with the right-to-left composition that we use. Note that the domain of (y,x) is x and the codomain is y.)

CO-3 If x is not related by α to y there is no arrow from x to y. The identity arrows of $\mathcal{C}(S,\alpha)$ are those of the form (x,x); they belong to α because it is reflexive. The transitive property of α is needed to ensure the existence of the composite described in 2.1.3, so that $(z,y)\circ(y,x) = (z,x)$.

(12) **Preord** – preorders and monotone functions: The objects of this category are preordered sets. If (S,α) and (T,β) are preordered sets, a function $f : S \longrightarrow T$ is *monotone* if whenever $x\alpha y$ in S, $f(x)\beta f(y)$ in T. If we write, as usual, $x \le y$ for $(x,y) \in \alpha$ and use the same for β: $u \le v$ if $(u,v) \in \beta$, then the condition for a monotone function is

$$\forall x, y \in S :\; x \le y \Longrightarrow f(x) \le f(y).$$

The identity function on a preordered set is clearly monotone, and the composite of two monotone functions is easily seen to be monotone, so that preordered sets with monotone functions form a category.

A variation on this is to consider only *strictly monotone functions*, which are functions f with the property that if

$$x\alpha y \text{ and } x \ne y \text{ then } f(x)\beta f(y) \text{ and } f(x) \ne f(y).$$

We saw how a single preordered set is a category. Now we are considering the category of preordered sets.

We must give a few words of warning on terminology. The usual word in mathematical texts for what we have called '*monotone*' is '*increasing*' or '*monotonically increasing*'. The word '*monotone*' is used for a function that either preserves or reverses the order relation. That is, in mathematical texts a function $f : (X,\alpha) \longrightarrow (T,\beta)$ is also called monotone if whenever $x\alpha y$ in S, $f(y)\beta f(x)$ in T.

(13) **Poset** – partial orders and monotone functions: A preordered set (S,α) for which α is antisymmetric (that is $x\alpha y$ and $y\alpha x$ imply $x = y$) is called a *partially ordered set* or *poset*. Two examples of posets are $(R,\le)$, the real numbers with the usual ordering, and for any set S, the poset $(\mathcal{P}(S),\subseteq)$, the set of subsets of S with inclusion as ordering. The posets and the monotone functions form the category **Poset**.

(14) **Mon** – monoids and monoid homomorphisms.

(15) **Grp** – groups and group homomorphisms.

(16) **SL** – semi-lattices and join-preserving functions.

(17) **Top** – topological spaces and continuous functions.

(18) **Met** – metric spaces and non-expansive (contracting) functions.

(19) **CMet**: complete metric spaces and non-expansive functions.

(20) **Vec** – vector spaces,

(21) **Ab** – Abelian groups,

(22) **SGrp** – semigroups:

(23) **Ban** – Banach spaces and linear functions,

(24) $\mathbf{Ban}_1$ – Banach spaces and linear functions of norm ≤ 1,

(25) **Mf** – manifolds (topological or differential),

(26) **Grf** – graphs: The category of graphs has graphs as objects and homomorphisms of graphs as arrows. It is denoted **Grf**.

The identity homomorphism $\mathrm{id}_{\mathcal{G}}$ is the identity function for both nodes and arrows. Let us check that the composite of graph homomorphisms is a graph homomorphism (identities are easy). Suppose $\phi : \mathcal{G} \longrightarrow \mathcal{H}$ and $\psi : \mathcal{H} \longrightarrow \mathcal{K}$ are graph homomorphisms, and suppose that $u : m \longrightarrow n$ in $\mathcal{G}$. Then by definition $\phi_E(u) : \phi_V(m) \longrightarrow \phi_V(n)$ in H , and so by definition

$$\psi_E(\phi_E(u)) : \psi_V(\phi_V(m)) \longrightarrow \psi_V(\phi_V(n)) \text{ in } K.$$

Hence $\psi \circ \phi$ is a graph homomorphism.

(27) An example from computer science:

A functional programming language has:

FPL-1 Primitive data types, given in the language.

FPL-2 Constants of each type.

FPL-3 Operations, which are functions between the types.

FPL-4 Constructors, which can be applied to data types and operations to produce derived data types and operations of the language.

The language consists of the set of all operations and types derivable from the primitive data types and primitive operations. The word '*primitive*' means given in the definition of the language rather than constructed by a constructor. Some authors use the word '*constructor*' for the primitive operations.

Given a functional programming language L, there's an associated category, where the objects are the data types of L, and the arrows are the computable functions of L ("processes", "procedures", "programs"). The composition of two such programs $X \xrightarrow{f} Y \xrightarrow{g!} Z$ is given by applying g to the output of f, sometimes also written $g \circ f = f; g$. The identity is the "do nothing" program. This example is closely related to the notion of "cartesian closed category" to be considered later.

As a concrete example, we will suppose we have a simple such language with three data types, NAT (natural numbers), BOOLEAN (true or false) and CHAR (characters). We give a description of its operations in categorical style.

(i) NAT should have a constant $0 : 1 \longrightarrow$ NAT and an operation $\textsc{succ}$: NAT $\longrightarrow$ NAT.

(ii) There should be two constants $\textsc{true}; \textsc{false} : 1 \longrightarrow$ BOOLEAN and an operation $\neg$ subject to the equations $\neg \circ \textsc{true} = \textsc{false}$ and $\neg \circ \textsc{false} = \textsc{true}$.

(iii) CHAR should have one constant $c : 1 \longrightarrow$ CHAR for each desired character c.

(iv) There should be two type conversion operations $\textsc{ord}$: CHAR $\longrightarrow$ NAT and $\textsc{chr}$: NAT $\longrightarrow$ CHAR. These are subject to the equation $\textsc{chr} \circ \textsc{ord} = \mathrm{id}_{\mathrm{CHAR}}$. (You can think of $\textsc{chr}$ as operating modulo the number of characters, so that it is defined on all natural numbers.)

An example program is the arrow '*next*' defined to be the composite $\textsc{chr} \circ \textsc{succ} \circ \textsc{ord}$: CHAR $\longrightarrow$ CHAR.

It calculates the next character in order. This arrow 'next' is an arrow in the category representing the language, and so is any other composite of a sequence of operations. The objects of the category $\mathcal{C}(L)$ of this language are the types NAT, BOOLEAN, CHAR and **1**. Observe that typing is a natural part of the syntax in this approach.

The arrows of $\mathcal{C}(L)$ consist of all programs, with two programs being identified if they must be the same because of the equations. For example, the arrow

$$\mathrm{CHR} \circ \mathrm{SUCC} \circ \mathrm{ORD} : \mathrm{CHAR} \longrightarrow \mathrm{CHAR}$$

just mentioned and the arrow

$$\mathrm{CHR} \circ \mathrm{SUCC} \circ \mathrm{ORD} \circ \mathrm{CHR} \circ \mathrm{ORD} : \mathrm{CHAR} \longrightarrow \mathrm{CHAR}$$

must be the same because of the equation in (iv).

Observe that NAT has constants $\mathrm{SUCC} \circ \mathrm{SUCC} \circ \ldots \circ \mathrm{SUCC} \circ 0$ where SUCC occurs zero or more times.

Composition in the category is composition of programs. Note that for composition to be well defined, if two composites of primitive operations are equal, then their composites with any other program must be equal. For example, we must have

$$\mathrm{ORD} \circ (\mathrm{CHR} \circ \mathrm{SUCC} \circ \mathrm{ORD}) = \mathrm{ORD} \circ (\mathrm{CHR} \circ \mathrm{SUCC} \circ \mathrm{ORD} \circ \mathrm{CHR} \circ \mathrm{ORD})$$

as arrows from CHAR to NAT.

1.4. **Constructions on categories.**

If you are familiar with some branch of abstract algebra (for example the theory of semigroups, groups or rings) then you know that given two structures of a given type (e.g., two semigroups), you can construct a '*direct product*' structure, defining the operations coordinatewise. Also, a structure may have substructures, which are subsets closed under the operations, and quotient structures, formed from equivalence classes modulo a congruence relation. Another construction that is possible in many cases is the formation of a '*free*' structure of the given type for a given set. All these constructions can be performed for categories. We will outline the constructions here, except for quotients. We will also describe the construction of the slice category, which does not quite correspond to anything in abstract algebra (although it is akin to the adjunction of a constant to a logical theory). You do not need to be familiar with the constructions in other branches of abstract algebra, since they are all defined from scratch here.

1.4.1. *Slice categories.*

1.4.1. **Definition.** If $\mathcal{C}$ is a category and A any object of $\mathcal{C}$, the *slice category* $\mathcal{C}/A$ is described this way:

- SC-1 An object of $\mathcal{C}/A$ is an arrow $f : C \longrightarrow A$ of $\mathcal{C}$ for some object C.
- SC-2 An arrow of $\mathcal{C}/A$ from $f : C \longrightarrow A$ to $f' : C' \longrightarrow A$ is an arrow $h : C \longrightarrow C'$ with the property that $f = f' \circ h$.
- SC-3 The composite of $h : f \longrightarrow f'$ and $h' : f' \longrightarrow f''$ is $h' \circ h$.

It is necessary to show that $h' \circ h$, as defined in SC-2, satisfies the requirements of being an arrow from f to f''. Let $h : f \longrightarrow f'$ and $h' : f' \longrightarrow f''$. This means $f' \circ h = f$ and $f'' \circ h' = f'$. To show that $h' \circ h : f \longrightarrow f''$ is an arrow of $\mathcal{C}/A$, we must show that $f'' \circ (h' \circ h) = f$. That follows from this calculation:

$$f'' \circ (h' \circ h) = (f'' \circ h') \circ h = f' \circ h = f.$$

The usual notation for arrows in $\mathcal{C}/A$ is deficient: the same arrow h can satisfy $f = f' \circ h$ and $g = g' \circ h$ with $f' \neq g'$ or $f \neq g$ (and then $f' \neq g'$). Then $h : f \longrightarrow f'$ and $h : g \longrightarrow g'$ are different arrows of $\mathcal{C}/A$

$$\begin{tikzcd} B \arrow[rr, "h"] \arrow[dr, "f"'] & & C \arrow[dl, "f'"] \\ & A & \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} B \arrow[rr, "h"] \arrow[dr, "g"'] & & C \arrow[dl, "g'"] \\ & A. & \end{tikzcd}$$

Indeed $h : (B, f) \longrightarrow (C, f')$ and $h : (B, g) \longrightarrow (C, g')$, or $h \in \mathrm{Mor}_{\mathcal{C}/A}((B, f), (C, f'))$ and $h \in \mathrm{Mor}_{\mathcal{C}/A}((B, g), (C, g'))$. But $\mathrm{Mor}_{\mathcal{C}/A}((B, f), (C, f')) \cap \mathrm{Mor}_{\mathcal{C}/A}((B, g), (C, g')) = \emptyset$ so that the two morphisms, both denoted h, are different.
Thus one often says, that an "arrow" in the slice category is a "*commutative diagram*" as above.

1.4.2. **Example.** Let (P, α) be a poset and let $\mathcal{C}(P)$ be the corresponding category as in 1.3.8(11). For an element $x \in P$, the slice category $\mathcal{C}(P)/x$ is the category corresponding to the set of elements greater than or equal to x. The dual notion of *coslice*, i.e. the category $A\backslash\mathcal{C}$ of arrows $f : A \longrightarrow C$ for some C in $\mathcal{C}$, gives the set of elements less than or equal to a given element.

1.4.3. **Remark.** The importance of slice categories comes in part with their connection with indexing. An *S-indexed set* or *graded set* is a set X together with a function $\tau : X \longrightarrow S$. If $x \in X$ and $\tau(x) = s$ then we say x is of *type* s, and we also refer to X as a *typed set*. The terminology '*S-indexed set*' is that used by category theorists. Many mathematicians would cast the discussion in terms of the collection $(\tau^{-1}(s)|s \in S)$ of subsets of X, which would be called a *family of sets indexed by* S.

1.4.4. **Example.** The set $G = G_0 \cup G_1$ of objects and arrows of a graph $\mathcal{G}$ is an example of a typed set, typed by the function $\tau : G \longrightarrow \{0, 1\}$ that takes a node to 0 and an arrow to 1. Note that this depends on the fact that a node is not an arrow: G_0 and G_1 are disjoint.

1.4.5. **Remark (Indexed functions).** A function from a set X typed by S to a set X' typed by the same set S that preserves the typing (takes an element of type s to an element of type s) is exactly an arrow of the slice category $\mathbf{Set}/S$. Such a function is called an *indexed function*, *typed function*, or *graded function*. It has been fruitful for category theorists to pursue this analogy by thinking of objects of any slice category $\mathcal{C}/A$ as objects of $\mathcal{C}$ indexed by A.

1.4.6. **Example.** A graph homomorphism $f : \mathcal{G} \longrightarrow \mathcal{H}$ corresponds to a typed function according to the construction in Example 1.4.4.
However, there are typed functions between graphs that are not graph homomorphisms, for example the function from the graph in 1.1.3(1)

$$\begin{tikzcd} 1 \arrow[loop above, "a"] \arrow[r, shift left, "b"] & 2 \arrow[l, shift left, "c"] \end{tikzcd}$$

to the graph in 1.1.3(4)

$$\begin{tikzcd} 1 \arrow[r, "0"] & \mathrm{n} \arrow[loop above, "\mathrm{succ}"] \end{tikzcd}$$

defined by

$$1 \mapsto 1, 2 \mapsto \mathrm{n}, a \mapsto 0, b \mapsto 0, c \mapsto \mathrm{succ}.$$

This is not a graph homomorphism because it does not preserve source and target.

1.4.2. *Subcategories.*

1.4.7. **Definition.** A *subcategory* $\mathcal{D}$ of a category $\mathcal{C}$ is a category for which:

S-1 All the objects of $\mathcal{D}$ are objects of $\mathcal{C}$ and all the arrows of $\mathcal{D}$ are arrows of $\mathcal{C}$ (in other words, $\mathcal{D}_0 \subseteq \mathcal{C}_0$ and $\mathcal{D}_1 \subseteq \mathcal{C}_1$).

S-2 The source and target of an arrow of $\mathcal{D}$ are the same as its source and target in $\mathcal{C}$ (in other words, the source and target maps for $\mathcal{D}$ are the restrictions of those for $\mathcal{C}$). It follows that for any objects A and B of $\mathcal{D}$, $\mathrm{Mor}_{\mathcal{D}}(A,B) \subseteq \mathrm{Mor}_{\mathcal{C}}(A,B)$.

S-3 If A is an object of $\mathcal{D}$ then its identity arrow 1_A in $\mathcal{C}$ is in $\mathcal{D}$.

S-4 If $f : A \longrightarrow B$ and $g : B \longrightarrow C$ in $\mathcal{D}$, then the composite $g \circ f$ (in $\mathcal{C}$) is in $\mathcal{D}$ and is the composite in $\mathcal{D}$.

1.4.8. **Examples.** As an example, the category **Fin** of finite sets and all functions between them is a subcategory of **Set**, and in turn **Set** is a subcategory of the category of sets and partial functions between sets (see 2.1.12 and 2.1.13). These examples illustrate two phenomena:

(i) If A and B are finite sets, then $\mathrm{Mor}_{\mathbf{Fin}}(A,B) = \mathrm{Mor}_{\mathbf{Set}}(A,B)$. In other words, every arrow of **Set** between objects of **Fin** is an arrow of **Fin**.

(ii) The category of sets and the category of sets and partial functions, on the other hand, have exactly the same objects.

The phenomenon of (i) does not occur here: there are generally many more partial functions between two sets than there are full functions.

Example (i) motivates the following definition.

1.4.9. **Definition.** If $\mathcal{D}$ is a subcategory of $\mathcal{C}$ and for every pair of objects A, B of $\mathcal{D}$, $\mathrm{Mor}_{\mathcal{D}}(A,B) = \mathrm{Mor}_{\mathcal{C}}(A,B)$, then $\mathcal{D}$ is a *full subcategory* of $\mathcal{C}$.

Thus **Fin** is a full subcategory of **Set** but **Set** is not a full subcategory of the category of sets and partial functions.
Example 1.4.8(ii) also motivates a (less useful) definition, as follows.

1.4.10. **Definition.** If $\mathcal{D}$ is a subcategory of $\mathcal{C}$ with the same objects, then $\mathcal{D}$ is a *wide subcategory* of $\mathcal{C}$.

Thus in the case of a wide subcategory, only the arrows are different from those of the larger category. In 2.4.9 we provide an improvement on this concept. As an example, **Set** is a wide subcategory of the category **Pfn** of sets and partial functions.

1.4.11. **Example.** Among all the objects of the category of semigroups are the monoids, and among all the semigroup homomorphisms between two monoids are those that preserve the identity. Thus the category of monoids is a subcategory of the category of semigroups that is neither wide nor full (see **??**). As it stands, being a subcategory requires the objects and arrows of the subcategory to be identical with some of the objects and arrows of the category containing it. This requires an uncategorical emphasis on what something is instead of on the specification it satisfies.

06.05.04

1.4.3. *The product of categories.*

1.4.12. **Definition.** If $\mathcal{C}$ and $\mathcal{D}$ are categories, the *product* $\mathcal{C} \times \mathcal{D}$ is the category whose objects are all ordered pairs (C,D) with C an object of $\mathcal{C}$ and D an object of $\mathcal{D}$, and in which an arrow $(f,g) : (C,D) \longrightarrow (C',D')$ is a pair of arrows $f : C \longrightarrow C'$ in $\mathcal{C}$ and $g : D \longrightarrow D'$ in $\mathcal{D}$. The identity of (C,D) is $(1_C, 1_D)$. If $(f',g') : (C',D') \longrightarrow (C'',D'')$ is another arrow, then the composite is defined by

$$(f',g') \circ (f,g) = (f' \circ f, g' \circ g) : (C,D) \longrightarrow (C'',D').$$

1.4.4. *The dual of a category.*

1.4.13. **Definition.** The *dual or opposite* $\mathcal{C}^{op}$ of a category $\mathcal{C}$ is defined by:

D-1 The objects and arrows of $\mathcal{C}^{op}$ are the objects and arrows of $\mathcal{C}$.
D-2 If $f : A \longrightarrow B$ in $\mathcal{C}$, then $f : B \longrightarrow A$ in $\mathcal{C}^{op}$, i.e. $\mathrm{Mor}_{\mathcal{C}^{op}}(B,A) := \mathrm{Mor}_{\mathcal{C}}(A,B)$.
D-3 If $h = g \circ f$ in $\mathcal{C}$, then $h = f \circ g$ in $\mathcal{C}^{op}$.

The meaning of D-2 is that source and target have been reversed. It is easy to see that the identity arrows have to be the same in the two categories $\mathcal{C}$ and $\mathcal{C}^{op}$ and that C-1 through C-4 of definition 1.3.1 hold, so that $\mathcal{C}^{op}$ is a category.

1.4.14. **Example.** If $\mathcal{M}$ is a monoid, then the opposite of the category $\mathcal{C}(\mathcal{M})$ is the category determined by a monoid $\mathcal{M}^{op}$, if $xy = z$ in $\mathcal{M}$, then $yx = z$ in $\mathcal{M}^{op}$. (Hence if $\mathcal{M}$ is commutative then $\mathcal{C}(\mathcal{M})$ is its own dual. Similar remarks may be made about the opposite of the category $\mathcal{C}(\mathcal{P})$ determined by a poset $\mathcal{P}$. The opposite of the poset $(\mathbb{Z}, \leq)$, for example, is $(\mathbb{Z}, \geq)$.

Both the construction of the product of two categories and the construction of the dual of a category are purely formal constructions. Even though the original categories may have, for example, structure-preserving functions of some kind as arrows, the arrows in the product category are simply pairs of arrows of the original categories.
Consider **Set**, for example. Let A be the set of letters of the English alphabet. The function $v : A \longrightarrow \{0,1\}$ that takes consonants to 0 and vowels to 1 is an arrow of **Set**. Then the arrow $(1_A, v) : (A,A) \longrightarrow (A, \{0,1\})$ of $\mathbf{Set} \times \mathbf{Set}$ is not a function, not even a function of two variables, since there are no reasonable elements in (A,A). It is merely the arrow of a product category and as such is an ordered pair of functions.
A similar remark applies to duals. In $\mathbf{Set}^{op}$, v is an arrow from $\{0,1\}$ to A. And that is all it is. It is in particular not a function from $\{0,1\}$ to A.
Nevertheless, it is possible in some cases to prove that the dual of a familiar category is essentially the same as some other familiar category. One such category is **Fin**, which is equivalent to the opposite of the category of finite Boolean algebras.
The product of categories is a formal way to make constructions dependent on more than one variable. The major use we make of the concept of dual is that many of the definitions we make have another meaning when applied to the dual of a category that is often of independent interest. The phrase dual concept or dual notion is often used to refer to a concept or notion applied in the dual category.

1.4.5. *The free category generated by a graph.*

1.4.15. **Definition.** For any given graph $\mathcal{G}$ there is a category $\mathcal{F}(\mathcal{G})$ whose objects are the nodes of $\mathcal{G}$ and whose arrows are the paths in $\mathcal{G}$. Composition is defined by the formula

$$(f_1, f_2, \ldots, f_k) \circ (f_{k+1}, \ldots, f_n) = (f_1, f_2, \ldots, f_n).$$

This composition is associative, and for each object A, 1_A is the empty path from A to A. The category $\mathcal{F}(\mathcal{G})$ is called the *free category generated by the graph* $\mathcal{G}$. It is also called the *path category of* $\mathcal{G}$.

1.4.16. **Examples.** – The free category generated by the graph with one node and no arrows is the category with one object and only the identity arrow, which is the empty path.
– The free category generated by the graph with one node and one loop on the node is the free monoid with one generator (Kleene closure of a one-letter alphabet); this is isomorphic with the nonnegative integers with + as operation.
– The free category generated by the graph in Example 1.1.3(4) has the following arrows

(a) An arrow $1_1 : 1 \longrightarrow 1$.
(b) For each nonnegative integer k, the arrow $\mathrm{succ}^k : \mathrm{n} \longrightarrow \mathrm{n}$. This is the path $(\mathrm{succ}, \mathrm{succ}, \dots, suc\mathrm{succ})$ (k occurrences of succ). This includes $k = 0$ which gives 1_n.
(c) For each nonnegative integer k, the arrow $\mathrm{succ}^k \circ 0 : 1 \longrightarrow \mathrm{n}$. Here $k = 0$ gives $0 : 1 \longrightarrow \mathrm{n}$. Composition obeys the rule $\mathrm{succ}^k \circ \mathrm{succ}^m = \mathrm{succ}^{k+m}$.

It is useful to regard the free category generated by any graph as analogous to the Kleene closure (free monoid) generated by a set (as in 1.2.5). The paths in the free category correspond to the strings in the Kleene closure. The difference is that you can concatenate any symbols together to get a string, but arrows can be strung together only head to tail, thus taking into account the typing. In **??** we give a precise technical meaning to the word 'free'.

1.5. **Functors.**

A functor $\mathcal{F}$ from a category $\mathcal{C}$ to a category $\mathcal{D}$ is a graph homomorphism which preserves identities and composition. It plays the same role as monoid homomorphisms for monoids and monotone maps for posets: it preserves the structure that a category has. Functors have another significance, however: since one sort of thing a category can be is a mathematical workspace (see Introduction), many of the most useful functors used by mathematicians are transformations from one type of mathematics to another. Less obvious, but perhaps more important is the fact that many categories that are mathematically interesting appear as categories whose objects are a natural class of functors into the category of sets.
A functor is a structure-preserving map between categories, in the same way that a homomorphism is a structure-preserving map between graphs or monoids. Here is the formal definition.

1.5.1. **Definition.** Let $\mathcal{C} = (\mathrm{Ob}(\mathcal{C}), \mathrm{Mor}(\mathcal{C}), (\mu), (1))$ and $\mathcal{D} = (\mathrm{Ob}(\mathcal{D}), \mathrm{Mor}(\mathcal{D}), (\mu), (1))$ be categories. Let $\mathcal{F}$ consist of

(1) a map $\mathcal{F} : \mathrm{Ob}(\mathcal{C}) \longrightarrow \mathrm{Ob}(\mathcal{D})$ and a
(2) a family of maps

$$(\mathrm{Mor}_{\mathcal{C}}(A, B) \ni f \mapsto \mathcal{F}(f) \in \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(A), \mathcal{F}(B)) \mid A, B \in \mathrm{Ob}(\mathcal{C})).$$

$\mathcal{F}$ is called a *covariant functor* if

- $\forall A \in \mathrm{Ob}(\mathcal{C})$:
$$\mathcal{F}(1_A) = 1_{\mathcal{F}(A)};$$
- $\forall A, B, C \in \mathrm{Ob}(\mathcal{C}),\ \forall f \in \mathrm{Mor}_{\mathcal{C}}(A, B), g \in \mathrm{Mor}_{\mathcal{C}}(B, C)$:
$$\mathcal{F}(gf) = \mathcal{F}(g)\mathcal{F}(f).$$

An equivalent definition is:

1.5.2. **Definition.** A functor $\mathcal{F} : \mathcal{C} \longrightarrow \mathcal{D}$ is a pair of functions $\mathcal{F}_0 : \mathcal{C}_0 \longrightarrow \mathcal{D}_0$ and $\mathcal{F}_1 : \mathcal{C}_1 \longrightarrow \mathcal{D}_1$ for which

F-1 If $f : A \longrightarrow B$ in $\mathcal{C}$, then $\mathcal{F}_1(f) : \mathcal{F}_0(A) \longrightarrow \mathcal{F}_0(B)$ in $\mathcal{D}$.

F-2 For any object A of $\mathcal{C}$, $\mathcal{F}_1(1_A) = 1_{\mathcal{F}_0(A)}$.

F-3 If $g \circ f$ is defined in $\mathcal{C}$, then $\mathcal{F}_1(g) \circ \mathcal{F}_1(f)$ is defined in $\mathcal{D}$ and $\mathcal{F}_1(g \circ f) = \mathcal{F}_1(g) \circ \mathcal{F}_1(f)$.

By F-1, a functor is in particular a homomorphism of graphs. Following the practice for graph homomorphisms, the notation is customarily overloaded: if A is an object, $\mathcal{F}(A) = \mathcal{F}_0(A)$ is an object, and if f is an arrow, $\mathcal{F}(f) = \mathcal{F}_1(f)$ is an arrow. The notation for the constituents $\mathcal{F}_0 : \mathcal{C}_0 \longrightarrow D_0$ and $\mathcal{F}_1 : \mathcal{C}_1 \longrightarrow \mathcal{D}_1$ is not standard, and we will use it only for emphasis.

1.5.3. **Example.** It is easy to see that a monoid homomorphism $f : \mathcal{M} \longrightarrow \mathcal{N}$ determines a functor from $\mathcal{C}(\mathcal{M})$ to $\mathcal{C}(\mathcal{N})$ as defined in 1.3.8(1). On objects, a homomorphism f must take the single object of $\mathcal{C}(\mathcal{M})$ to the single object of $\mathcal{C}(\mathcal{N})$, and F-1 is trivially verified since all arrows in $\mathcal{C}(\mathcal{M})$ have the same domain and codomain and similarly for $\mathcal{C}(\mathcal{N})$. Then F-2 and F-3 say precisely that f is a monoid homomorphism. Conversely, every functor is determined in this way by a monoid homomorphism.

1.5.4. **Example.** Let us see what a functor from $\mathcal{C}(S, \alpha)$ to $\mathcal{C}(T, \beta)$ must be when (S, α) and (T, β) are posets as in 1.3.8(1). It is suggestive to write both relations α and β as '$\leq$' and the posets simply as S and T. Then there is exactly one arrow from x to y in S (or in T) if and only if $x \leq y$; otherwise there are no arrows from x to y.
Let $f : S \longrightarrow T$ be the functor. F-1 says if there is an arrow from x to y, then there is an arrow from $f(x)$ to $f(y)$; in other words, if $x \leq y$ then $f(x) \leq f(y)$.
Thus f is a monotone map. F-2 and F-3 impose no additional conditions on f because they each assert the equality of two specified arrows between two specified objects and in a poset as category all arrows between two objects are equal.

1.5.5. **Remark** (**The category of categories).** The category **Cat** has all small categories as objects and all functors between such categories as arrows. The composite of functors is their composite as graph homomorphisms: if $\mathcal{F} : \mathcal{C} \longrightarrow \mathcal{D}$ and $\mathcal{G} : \mathcal{D} \longrightarrow \mathcal{E}$, then $\mathcal{G} \circ \mathcal{F} : \mathcal{C} \longrightarrow \mathcal{E}$ satisfies $\mathcal{G} \circ \mathcal{F}(C) = \mathcal{G}(\mathcal{F}(C))$ for any object C of $\mathcal{C}$, and $G \circ \mathcal{F}(f) = \mathcal{G}(\mathcal{F}(f))$ for any arrow f of $\mathcal{C}$. Thus $(G \circ \mathcal{F})_i = \mathcal{G}_i \circ \mathcal{F}_i$ for $i = 0, 1$. We note that the composition circle is usually omitted when composing functors so that we write $\mathcal{G}\mathcal{F}(C) = \mathcal{G}(\mathcal{F}(C))$. It is sometimes convenient to refer to a category **CAT** which has all small categories and ordinary large categories as objects, and functors between them. Since trying to have **CAT** be an object of itself would raise delicate foundational questions, we do not attempt here a formal definition of **CAT**.

1.5.6. **Example.** If $\mathcal{C}$ is a category, the functor $P_1 : \mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}$ which takes an object (C, D) to C and an arrow $(f, g) : (C, D) \longrightarrow (C', D')$ to f is called the *first projection*. There is an analogous *second projection functor* P_2 taking an object or arrow to its second coordinate.

1.5.7. **Example.** Let $\mathbf{2} + \mathbf{2}$ be the category that can be pictured as

$$0 \longrightarrow 1 \qquad 1' \longrightarrow 2$$

with no other nonidentity arrows, and the category $\mathbf{3}$ the one that looks like

$$\begin{tikzcd} 0 \arrow[rr] \arrow[dr] & & 1 \arrow[dl] \\ & 2 & \end{tikzcd}$$

Define the functor $\mathcal{F} : \mathbf{2} + \mathbf{2} \longrightarrow \mathbf{3}$ to take 0 to 0, 1 and 1' to 1, and 2 to 2. Then what it does on arrows is forced.

Note that the image of $\mathcal{F}$ includes all of **3** except the composite arrow from $0 \longrightarrow 2$. This example shows that the image of a functor need not be a subcategory of the codomain category.

1.5.8. **Example.** The inclusion map of a subcategory is a functor. The categorical point of view does not require that the object and arrows of a subcategory actually be objects and arrows of the bigger category, only that there be an injective identification of objects or morphisms of one category with the objects or morphisms of the other. For example, **Set** is a subcategory of **Rel**: the functor takes every set to itself and each function $f : S \longrightarrow T$ to its graph $\{(s, t)|t = f(s)\}$, which is indeed a relation from S to T.
This approach has the strange result that two different categories can each be regarded as subcategories of the other one.
Observe that the category Ri of rings is not a subcategory of the category Ab of abelian group, although each ring is also an abelian group (under the addition) and every homomorphism of rings is also a homomorphism of abelian groups. There can be two nonisomorphic rings, that have the same additive abelian group.

1.6. **Special types of functors.**

1.6.1. *Underlying functors.*

1.6.1. **Example.** Forgetting some of the structure in a category of structures and structure-preserving functions gives a functor called an *underlying functor* or *forgetful functor*. The functor U : **Mon** $\longrightarrow$ **Sem** which embeds the category of monoids into the category of semigroups by forgetting that a monoid has an identity is an example of an underlying functor.
Another example is the functor which forgets all the structure of a semigroup. This is a functor U : **Sem** $\longrightarrow$ **Set**. There are lots of semigroups with the same set of elements; for example, the set $\{0, 1, 2\}$ is a semigroup on addition (mod 3) and also a different semigroup on multiplication (mod 3). The functor U applied to these two different semigroups gives the same set, so U is not injective on objects, in contrast to the forgetful functor from monoids to semigroups. We will not give a formal definition of underlying functor. It is reasonable to expect any underlying functor U to be faithful, i.e. injective on the morphism sets, and that if f is an isomorphism and $U(f)$ is an identity arrow then f is an identity arrow.

1.6.2. **Example.** A small graph has *two* underlying sets: its set of nodes and its set of arrows. Thus there is an underlying functor U : **Grf** $\longrightarrow$ **Set** $\times$ **Set** for which for a graph $\mathcal{G}$, $U(\mathcal{G}) = (\mathcal{G}_0, \mathcal{G}_1)$, an arrowset functor A : **Grf** $\longrightarrow$ **Set** which takes a graph to its set of arrows and a graph homomorphism to the corresponding function from arrows to arrows, and a similarly defined nodeset functor N : **Grf** $\longrightarrow$ **Set** which takes a graph to its set of nodes.

1.6.3. **Example.** If you forget you can compose arrows in a category and you forget which arrows are the identities, then you have remembered only that the category is a graph. This gives an underlying functor U : **Cat** $\longrightarrow$ **Grf**, since every functor is a graph homomorphism although not vice versa. As for graphs, there are also set-of-objects and set-of-arrows functors O : **Cat** $\longrightarrow$ **Set** and A : **Cat** $\longrightarrow$ **Set** which take a category to its set of objects and set of arrows respectively, and a functor to the appropriate set map.

1.6.4. **Example.** In 1.4.1, we described the notion of a slice category $\mathcal{C}/A$ based on a category $\mathcal{C}$ and an object A. An object is an arrow $B \longrightarrow A$ and an arrow from $f : B \longrightarrow A$ to $g : C \longrightarrow A$ is an arrow $h : B \longrightarrow C$ for which $g \circ h = f$. There is a functor $U : \mathcal{C}/A \longrightarrow \mathcal{C}$ that takes the object $f : B \longrightarrow A$ to B and the arrow h from $B \longrightarrow A$ to $C \longrightarrow A$ to $h : B \longrightarrow C$. This is called the *underlying functor of the slice.* In the case that $\mathcal{C} = \mathbf{Set}$, an object $T \longrightarrow S$ of $\mathbf{Set}/S$ for some set S is an S-indexed object, and the effect of the underlying functor is to forget the indexing.

1.6.2. *Free functors.*

1.6.5. **Example.** The free monoid functor from **Set** to the category of monoids takes a set A to the free monoid $F(A)$, which is the Kleene closure A^* with concatenation as operation (see 1.2.5), and a function $f : A \longrightarrow B$ to the function $F(f) = f^* : F(A) \longrightarrow F(B)$ defined in 1.2.16. To see that the free monoid functor is indeed a functor it is necessary to show that if $f : A \longrightarrow B$ and $g : B \longrightarrow C$, then $F(g \circ f) : F(A) \longrightarrow F(C)$ is the same as $F(g) \circ F(f)$, which is immediate from the definition, and that it preserves identity arrows, which is also immediate. The Kleene closure is itself a functor from $\mathbf{Set}\, to\, \mathbf{Set}$, taking A to A^* and f to f^*. It is the composite $U \circ F$ of the underlying functor $U : \mathbf{Mon} \longrightarrow \mathbf{Set}$ and the free functor $F : \mathbf{Set} \longrightarrow \mathbf{Mon}$, but of course it can be defined independently of U and F.

1.6.6. **Example.** The free category on a graph is also the object part of a functor $F : \mathbf{Grf} \longrightarrow \mathbf{Cat}$. What it does to a graph is described in 1.4.15. Suppose $\varphi : \mathcal{G} \longrightarrow \mathcal{H}$ is a graph homomorphism. The objects of the free category on a graph are the nodes of the graph, so it is reasonable to define $F(\varphi)_0 = \varphi_0$. Now suppose $(f_n, f_{n-1}, \ldots, f_1)$ is a path, that is, an arrow, in $F(\mathcal{G})$. Since functors preserve domain and codomain, we can define $F(\varphi)_1(f_n, f_{n-1}, \ldots, f_1)$ to be $(\varphi_1(f_n), \varphi_1(f_{n-1}), \ldots, \varphi_1(f_1))$ and know we get a path in $F(\mathcal{H})$. That F preserves composition of paths is also clear.

1.6.7. **Remark (The map-lifting property).** The free category functor $F : \mathbf{Grf} \longrightarrow \mathbf{Cat}$ and also other free functors, such as the free monoid functor 1.6.5, have a map lifting property called its *universal mapping property* which will be seen as the defining property of freeness. We will describe the property for free categories since we use it later.
Let $\mathcal{G}$ be a graph and $F(\mathcal{G})$ be the free category generated by $\mathcal{G}$. There is a graph homomorphism with the special name $\eta\mathcal{G} : \mathcal{G} \longrightarrow U(F(\mathcal{G}))$ which includes a graph $\mathcal{G}$ into $U(F(\mathcal{G}))$, the underlying graph of the free category $F(\mathcal{G})$. The map $(\eta G)_0$ is the identity, since the objects of $F(\mathcal{G})$ are the nodes of $\mathcal{G}$. For an arrow f of $\mathcal{G}$, $(\eta\mathcal{G})_1(f)$ is the path (f) of length one. This is an inclusion arrow in the generalized categorical sense, since f and (f) are really two distinct entities.

1.6.8. **Proposition.** *Let $\mathcal{G}$ be a graph and $\mathcal{C}$ a category. Then for every graph homomorphism $h : \mathcal{G} \longrightarrow U(\mathcal{C})$, there is a unique functor $\widehat{h} : F(\mathcal{G}) \longrightarrow \mathcal{C}$ with the property that $U(\widehat{h}) \circ \eta\mathcal{G} = h$.*

Proof. If () is the empty path at an object a, we set $\widehat{h}() = 1_a$. For an object a of $F(\mathcal{G})$ (that is, node of $\mathcal{G}$), define $\widehat{h}(a) = h(a)$. And for a path $(a_n, a_{n-1}, \ldots, a_1)$, $\widehat{h}$ is 'map h':

$$\widehat{h}(a_n, a_{n-1}, \ldots, a_1) = (h(a_n), h(a_{n-1}), \ldots, h(a_1)).$$

As noted in 1.1.1, there is a unique empty path for each node a of $\mathcal{G}$. Composing the empty path at a with any path p from a to b gives p again, and similarly on the other side. □

1.6.3. *Powerset functors.*
Any set S has a powerset $\mathcal{P}S$, the set of all subsets of S. There are three different functors $\mathcal{F}$ for which $\mathcal{F}_0$ takes a set to its powerset; they differ on what they do to arrows. One of them is fundamental in topos theory; that one we single out to be called the powerset functor.
If $f : A \longrightarrow B$ is any set function and C is a subset of B, then the *inverse image* of C, denoted $f^{-1}(C)$, is the set of elements of A which f takes into C: $f^{-1}(C) = \{a \in A | f(a) \in C\}$. Thus f^{-1} is a function from $\mathcal{P}B$ to $\mathcal{P}A$. Note that for a bijection f, the symbol f^{-1} is also used to denote the inverse function. Context makes it clear which is meant, since the input to the inverse image function must be a subset of the codomain of f, whereas the input to the actual inverse of a bijection must be an element of the codomain.

1.6.9. **Definition.** The *powerset functor* $\mathcal{P} : \mathbf{Set}^{op} \longrightarrow \mathbf{Set}$ takes a set S to the powerset $\mathcal{P}S$, and a set function $f : A \longrightarrow B$ (that is, an arrow from B to A in $\mathbf{Set}^{op}$) to the inverse image function $f^{-1} : \mathcal{P}B \longrightarrow \mathcal{P}A$.
Although we will continue to use the notation f^{-1}, it is denoted f^* in much of the categorical literature. To check that $\mathcal{P}$ is a functor requires showing that $1_A^{-1} = 1_{\mathcal{P}A}$ and that if $g : B \longrightarrow C$, then $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$, where both compositions take place in $\mathbf{Set}$.

1.6.10. **Definition.** A functor $\mathcal{F} : \mathcal{C}^{op} \longrightarrow \mathcal{D}$ is also called a *contravariant functor* from $\mathcal{C}$ to $\mathcal{D}$.

As illustrated in the preceding definition, the functor is often defined in terms of arrows of $\mathcal{C}$ rather than of arrows of $\mathcal{C}^{op}$. Opposite categories are most commonly used to provide a way of talking about contravariant functors as ordinary (covariant) functors: the opposite
13.05.04 category in this situation is a purely formal construction of no independent interest.

1.6.11. **Remark.** The other two functors which take a set to its powerset are both covariant. The *direct or existential image functor* takes $f : A \longrightarrow B$ to the function $f_* : \mathcal{P}A \longrightarrow \mathcal{P}B$, where $f_*(A_0) = \{f(x) | x \in A_0\}$, the set of values of f on A_0.
The *universal image functor* takes A_0 to those values of f which come only from A_0: formally, it takes $f : A \longrightarrow B$ to $f_! : \mathcal{P}A \longrightarrow \mathcal{P}B$, with

$$f_!(A_0) = \{y \in B | f(x) = y \text{ implies } x \in A_0\} = \{y \in B | f^{-1}(\{y\}) \subseteq A_0\}.$$

1.6.4. *Hom functors or Mor functors.*
Let $\mathcal{C}$ be a category with an object C and an arrow $f : A \longrightarrow B$. f induces a set function $\mathrm{Mor}(A, f) : \mathrm{Mor}(A, B) \longrightarrow \mathrm{Mor}(A, C)$ defined by composing by f on the left: for any $g \in \mathrm{Mor}(A, B)$, that is, for any $g : A \longrightarrow B$, $\mathrm{Mor}(A, f)(g) = f \circ g$, which does indeed go from A to C. Similarly, for any object D, $f : B \longrightarrow C$ induces a set function $\mathrm{Mor}(f, D) : \mathrm{Mor}(C, D) \longrightarrow \mathrm{Mor}(B, D)$ (note the reversal) by requiring that $\mathrm{Mor}(f, D)(h) = h \circ f$ for $h \in \mathrm{Mor}(C, D)$.

1.6.12. **Definition.** For a given object C the *covariant Mor functor* $\mathrm{Mor}(C, \text{-}) : \mathcal{C} \longrightarrow \mathbf{Set}$ is defined as follows:

HF-1 $\mathrm{Mor}(C, \text{-})(A) := \mathrm{Mor}(C, A)$ for each object A of $\mathcal{C}$;
HF-2 $\mathrm{Mor}(C, \text{-})(f) := \mathrm{Mor}(C, f) : \mathrm{Mor}(C, A) \longrightarrow \mathrm{Mor}(C, B)$ for $f : A \longrightarrow B$.

The following calculations show that $\mathrm{Mor}(C, \text{-})$ is a functor. For an object A, $\mathrm{Mor}(C, 1_A) : \mathrm{Mor}(C, A) \longrightarrow \mathrm{Mor}(C, A)$ takes an arrow $f : C \longrightarrow A$ to $1_A \circ f = f$; hence $\mathrm{Mor}(C, 1_A) =$

$1_{\mathrm{Mor}(C,A)}$. Now suppose $f : A \longrightarrow B$ and $g : B \longrightarrow D$. Then for any arrow $k : C \longrightarrow A$,

$$\begin{aligned}\big(\mathrm{Mor}(C,g)\circ\mathrm{Mor}(C,f)\big)(k) &= \mathrm{Mor}(C,g)\big(\mathrm{Mor}(C,f)(k)\big)\\ &= \mathrm{Mor}(C,g)(f\circ k)\\ &= g\circ(f\circ k)\\ &= (g\circ f)\circ k\\ &= \mathrm{Mor}(C,g\circ f)(k).\end{aligned}$$

There is a distinct covariant Mor functor $\mathrm{Mor}(C,\text{-})$ for each object C. In this expression, C is a parameter for a family of functors. The argument of each of these functors is indicated by the dash. An analogous definition in calculus would be to define the function which raises a real number to the nth power as $f(\text{-}) = (\text{-})^n$ (here n is the parameter). One difference in the Mor functor case is that the Mor functor is overloaded and so has to be defined on two different kinds of things: objects and arrows.

1.6.13. **Definition.** For a given object D, the *contravariant mor functor* $\mathrm{Mor}(\text{-},D) : \mathcal{C}^{op} \longrightarrow \mathbf{Set}$ is defined for each object A by $\mathrm{Mor}(\text{-},D)(A) := \mathrm{Mor}(A,D)$ and for each arrow $f : A \longrightarrow B$, $\mathrm{Mor}(\text{-},D)(f) := \mathrm{Mor}(f,D) : \mathrm{Mor}(B,D) \longrightarrow \mathrm{Mor}(A,D)$. Thus if $g : B \longrightarrow D$, $\mathrm{Mor}(f,D)(g) = g\circ f$.

1.6.14. **Definition.** The *two-variable Mor functor*

$$\mathrm{Mor}(\text{-},\text{-}) : \mathcal{C}^{op}\times\mathcal{C} \longrightarrow \mathbf{Set}$$

takes a pair (C,D) of objects of $\mathcal{C}$ to $\mathrm{Mor}(C,D)$, and a pair (f,g) of arrows with $f : C \longrightarrow A$ and $g : B \longrightarrow D$ to

$$\mathrm{Mor}(f,g) : \mathrm{Mor}(A,B) \longrightarrow \mathrm{Mor}(C,D)$$

where for $h : A \longrightarrow B$,

$$\mathrm{Mor}(f,g)(h) = g\circ h\circ f$$

which is indeed an arrow from C to D.

In this case we also use the product of categories as a formal construction to express functors of more than one argument. From the categorical point of view, a functor always has one argument, which as in the present case might well be an object in a product category (an ordered pair).

1.7. **Natural transformations.**

In order to see one of the aims of this section we first introduce natural transformation of functors and study their most elementary properties.

1.7.1. *Natural transformations between functors I.*

1.7.1. **Definition.** Let $D, E : \mathcal{C} \longrightarrow \mathcal{D}$ be two covariant functors. A *natural transformation* $\alpha : D \longrightarrow E$ is given by a family of morphisms α_A of $\mathcal{D}$ indexed by the objects of $\mathcal{C}$ such that:

NTF-1 $\alpha_A : D(A) \longrightarrow E(A)$ for each node A of $\mathcal{C}$.

NTF-2 For any morphism $f : A \longrightarrow B$ in $\mathcal{C}$, the diagram

$$\begin{tikzcd} D(A) \arrow[r, "\alpha_A"] \arrow[d, "D(f)"'] & E(A) \arrow[d, "E(f)"] \\ D(B) \arrow[r, "\alpha_B"'] & E(B) \end{tikzcd} \tag{1}$$

commutes.

1.7.2. **Definition.** Let D, E and F be functors from $\mathcal{C}$ to $\mathcal{D}$, and $\alpha : D \longrightarrow E$ and $\beta : E \longrightarrow F$ natural transformations. The composite $\beta \circ \alpha : D \longrightarrow F$ is defined componentwise: $(\beta \circ \alpha)_A = \beta_A \circ \alpha_A$.

1.7.3. **Proposition.** *The composite of two natural transformations is also a natural transformation.*

Proof. The diagram that has to be shown commutative is the outer rectangle of

$$\begin{tikzcd} D(A) \arrow[r, "\alpha_A"] \arrow[d, "D(f)"'] & E(A) \arrow[r, "\beta_A"] \arrow[d, "E(f)"'] & F(A) \arrow[d, "F(f)"] \\ D(B) \arrow[r, "\alpha_B"'] & E(B) \arrow[r, "\beta_B"'] & F(B) \end{tikzcd} \tag{2}$$

for each arrow $f : A \longrightarrow B$ in $\mathcal{C}$. The rectangle commutes because the two squares do; the squares commute as a consequence of the naturality of α and β. □

1.7.4. **Example.** Examples for natural transformations – like for functors – are built into the definition of categories. Let $f : A \longrightarrow B$ be a morphism in a category $\mathcal{C}$. In 1.6.12 we have already studied covariant functors $\mathrm{Mor}_{\mathcal{C}}(A,\text{-}), \mathrm{Mor}_{\mathcal{C}}(B,\text{-}) : \mathcal{C} \longrightarrow \mathbf{Set}$.
The morphism f induces a family of maps

$$\mathrm{Mor}_{\mathcal{C}}(f, C) : \mathrm{Mor}_{\mathcal{C}}(B, C) \ni g \mapsto g \circ f \in \mathrm{Mor}_{\mathcal{C}}(A, C)$$

which is a natural transformation by Exercise **??**.

1.7.5. **Definition.** A natural transformation $\alpha : F \longrightarrow G : \mathcal{C} \longrightarrow \mathcal{D}$ is called a *natural isomorphism* if there is a natural transformation $\beta : G \longrightarrow F$ which is an inverse to α.

1.7.6. **Definition.** A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is an *equivalence of categories* if there are:

E-1 A functor $G : \mathcal{D} \longrightarrow \mathcal{C}$.
E-2 A family $(u_C : C \longrightarrow G(F(C)) | C \in \mathrm{Ob}(\mathcal{C}))$ of isomorphisms of $\mathcal{C}$ indexed by the objects of $\mathcal{C}$ with the property that for every arrow $f : C \longrightarrow C'$ of $\mathcal{C}$, $G(F(f)) = u_{C'} \circ f \circ u_C^{-1}$.
E-3 A family $v_D : D \longrightarrow F(G(D))$ of isomorphisms of $\mathcal{D}$ indexed by the objects of $\mathcal{D}$, with the property that for every arrow $g : D \longrightarrow D'$ of $\mathcal{D}$, $F(G(g)) = v_{D'} \circ g \circ v_D^{-1}$.

If F is an equivalence of categories, the functor G of E-1 is called a *pseudo-inverse* of F.
The idea behind the definition is that not only is every object of $\mathcal{D}$ isomorphic to an object in the image of F, but the isomorphisms are compatible with the arrows of $\mathcal{D}$; and similarly
14.05.04 for $\mathcal{C}$.

1.7.7. **Example.** An isomorphism of categories is an equivalence of categories, and its inverse is its pseudo-inverse.

1.7.8. **Example** (Monoids and one-object categories.)**.** For each monoid M we constructed a small category $C(M)$ in 1.3.8. We make the choice mentioned there that the one object of $C(M)$ is M. Note that although an element of $C(M)$ is now an arrow from M to M, it is not a set function. For each monoid homomorphism $h : M \longrightarrow N$, construct a functor $C(h) : C(M) \longrightarrow C(N)$ as follows:

CF-1 On objects, $C(h)(M) = N$.
CF-2 $C(h)$ must be exactly the same as h on arrows (elements of M).

It is straightforward to see that $C(h)$ is a functor and that this construction makes C a functor from **Mon** to the full subcategory of **Cat** of categories with exactly one object. We will denote this full subcategory as **Ooc**.
There is also a functor $U : \mathbf{Ooc} \longrightarrow \mathbf{Mon}$ going the other way.

UO-1 For a category $\mathcal{C}$ with one object, $U(\mathcal{C})$ is the monoid whose elements are the arrows of $\mathcal{C}$ and whose binary operation is the composition of $\mathcal{C}$.

UO-2 If $F : \mathcal{C} \longrightarrow \mathcal{D}$ is a functor between one- object categories, $U(F) = F_1$, that is, the functor F on arrows.

The functors U and C are not inverse to each other, and it is worthwhile to see in detail why. The construction of C is in part arbitrary. We needed to regard each monoid as a category with one object. The choice of the elements of M to be the arrows of the category is obvious, but what should be the one object? We chose M itself, but we could have chosen some other thing, such as the set $\{e\}$, where e is the identity of M. The only real requirement is that it not be an element of M (such as its identity) in order to avoid set-theoretic problems caused by the category being an element of itself. The consequence is that we have given a functor $C : \mathbf{Mon} \longrightarrow \mathbf{Ooc}$ in a way which required arbitrary choices. The arbitrary choice of one object for $C(M)$ means that if we begin with a one-object category $\mathcal{C}$, construct $M = U(\mathcal{C})$, and then construct $C(M)$, the result will not be the same as $\mathcal{C}$ unless it happens that the one object of $\mathcal{C}$ is M. Thus $C \circ U \neq \mathrm{id}_{\mathbf{Ooc}}$, so that U is not the inverse of C. (In this case $U \circ C$ is indeed $\mathrm{id}_{\mathbf{Mon}}$.)
C is not surjective on objects, since not every small category with one object is in the image of C; in fact a category $\mathcal{D}$ is $C(M)$ for some monoid M only if the single object of $\mathcal{D}$ is actually a monoid and the arrows of $\mathcal{D}$ are actually the arrows of that monoid. This is entirely contrary to the spirit of category theory: we are talking about specific elements rather than specifying behavior. Indeed, in terms of specifying behavior, the category of monoids and the category of small categories with one object ought to be essentially the same thing.
The fact that C is not an isomorphism of categories is a signal that isomorphism is the wrong idea for capturing the concept that two categories are essentially the same. However, every small category with one object is isomorphic to one of those constructed as $C(M)$ for some monoid M. This is the starting point for the definition of equivalence.

1.7.9. **Proposition.** *A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is an equivalence of categories with pseudo-inverse $G : \mathcal{D} \longrightarrow \mathcal{C}$ if and only if $G \circ F$ is naturally isomorphic to $\mathrm{id}_{\mathcal{C}}$ and $F \circ G$ is naturally isomorphic to $\mathrm{id}_{\mathcal{D}}$.*

Proof. Conditions E-2 and E-3 of 1.7.6 can be recast as the statement that $G(F(f)) \circ u_C = u_{C'} \circ f$ and that $F(G(g)) \circ v_D = v_{D'} \circ g$, in other words that the following diagrams commute:

$$\begin{tikzcd} C \arrow[r, "u_C"] \arrow[d, "f"'] & G(F(C)) \arrow[d, "G(F(f))"] \\ C' \arrow[r, "u_{C'}"'] & G(F(C')) \end{tikzcd} \qquad \begin{tikzcd} D \arrow[r, "v_D"] \arrow[d, "g"'] & F(G(D)) \arrow[d, "F(G(g))"] \\ D' \arrow[r, "v_{D'}"'] & F(G(D')) \end{tikzcd} \tag{3}$$

In this form, they are the statements that u is a natural transformation from $\mathrm{id}_{\mathcal{C}}$ to $G \circ F$ and that v is a natural transformation from $\mathrm{id}_{\mathcal{D}}$ to $F \circ G$. Since each component of u and each component of v is an isomorphism, u and v are natural isomorphisms. $\square$

1.7.2. *Diagrams.*
Commutative diagrams are the categorist's way of expressing equations. Natural transformations are maps between functors; one way to think of them is as a deformation of one construction (construed as a functor) into another.
We begin with diagrams in a graph and discuss commutativity later.

1.7.10. **Definition.** Let I and $\mathcal{G}$ be graphs. A *diagram in* $\mathcal{G}$ of *shape* I is a homomorphism $\mathcal{D} : I \longrightarrow \mathcal{G}$ of graphs. I is called the *shape graph* of the diagram $\mathcal{D}$. We have thus given a new name to a concept which was already defined (not uncommon in mathematics). A *diagram* is a graph homomorphism from a different point of view.

1.7.11. **Example.** At first glance, Definition 1.7.10 may seem to have little to do with what are informally called diagrams, for example

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "h"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

The connection is this: a diagram in the sense of Definition 1.7.10 is pictured on the page with a drawing of nodes and arrows as for example in the previous diagram, which could be the picture of a diagram $\mathcal{D}$ with shape graph

$$\begin{tikzcd} i \arrow[rr, "u"] \arrow[dr, "w"'] & & j \arrow[dl, "v"] \\ & k & \end{tikzcd}$$

defined by $\mathcal{D}(i) = A$, $\mathcal{D}(j) = B$, $\mathcal{D}(k) = C$, $\mathcal{D}(u) = f$, $\mathcal{D}(v) = g$ and $D(w) = h$.

1.7.12. **Example.** Here is an example illustrating some subtleties involving the concept of diagram. Let $\mathcal{G}$ be a graph with objects A, B and C (and maybe others) and arrows $f : A \longrightarrow B$, $g : B \longrightarrow C$ and $h : B \longrightarrow B$. Consider these two diagrams, where here we use the word '*diagram*' informally:

$$\begin{tikzcd} A \arrow[r, "f"] & B \arrow[r, "g"] & C \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] & B \arrow[loop above, "\text{h}"] \end{tikzcd}$$

These are clearly of different shapes (again using the word '*shape*' informally). But the diagram

$$\begin{tikzcd} A \arrow[r, "f"] & B \arrow[r, "h"] & B \end{tikzcd}$$

is the same shape as $\mathcal{I}$ even though as a graph it is the same as $\mathcal{J}$. To capture the difference thus illustrated between a graph and a diagram, we introduce two shape graphs

$$\mathcal{I}:\ \begin{tikzcd} 1 \arrow[r, "u"] & 2 \arrow[r, "v"] & 3 \end{tikzcd} \qquad \mathcal{J}:\ \begin{tikzcd} 1 \arrow[r, "u"] & 2 \arrow[loop above, "\text{w}"] \end{tikzcd}$$

(where, as will be customary, we use numbers for the nodes of shape graphs). Now the first diagram is seen to be the diagram $\mathcal{D} : \mathcal{I} \longrightarrow \mathcal{G}$ with $\mathcal{D}(1) = A$, $\mathcal{D}(2) = B$, $\mathcal{D}(3) = C$, $\mathcal{D}(u) = f$, and $\mathcal{D}(v) = g$; whereas the second diagram is $\mathcal{E} : \mathcal{J} \longrightarrow \mathcal{G}$ with $\mathcal{E}(1) = A$, $\mathcal{E}(2) = B$, $\mathcal{E}(u) = f$, and $\mathcal{E}(w) = h$. Moreover, the third diagram is just like $\mathcal{D}$ (has the same shape), except that v goes to h and 3 goes to B.

1.7.13. **Remark.** Our definition in 1.7.10 of a diagram as a graph homomorphism, with the domain graph being the shape, captures both the following ideas:

(1) A diagram can have repeated labels on its nodes and (although the examples did not show it) on its arrows, and
(2) Two diagrams can have the same labels on their nodes and arrows but be of different shapes: the second diagram and the third diagram are different diagrams because they have different shapes.

1.7.3. *Commutative diagrams.*
When the target graph of a diagram is the underlying graph of a category some new possibilities arise, in particular the concept of commutative diagram, which is the categorist's way of expressing equations. In this situation, we will not distinguish in notation between the category and its underlying graph: if I is a graph and $\mathcal{C}$ is a category we will refer to a diagram $\mathcal{D} : I \longrightarrow \mathcal{C}$. We say that $\mathcal{D}$ is *commutative* (or *commutes*) provided for any nodes i and j of I and any two paths

$$\begin{tikzcd}
& k_1 \arrow[r, "s_2"] & k_2 \arrow[r] & \ldots \arrow[r] & k_{n-2} \arrow[r, "s_{n-1}"] & k_{n-1} \arrow[dr, "s_n"] & \\
i \arrow[ur, "s_1"] \arrow[dr, "t_1"'] & & & & & & j \\
& l_1 \arrow[r, "t_2"] & l_2 \arrow[r] & \ldots \arrow[r] & l_{m-2} \arrow[r, "t_{m-1}"] & l_{m-1} \arrow[ur, "t_m"'] &
\end{tikzcd}$$

from i to j in I, the two paths

$$\begin{tikzcd}
& \mathcal{D}k_1 \arrow[r, "\mathcal{D}s_2"] & \mathcal{D}k_2 \arrow[r] & \ldots \arrow[r] & \mathcal{D}k_{n-2} \arrow[r, "\mathcal{D}s_{n-1}"] & \mathcal{D}k_{n-1} \arrow[dr, "\mathcal{D}s_n"] & \\
\mathcal{D}i \arrow[ur, "\mathcal{D}s_1"] \arrow[dr, "\mathcal{D}t_1"'] & & & & & & \mathcal{D}j \\
& \mathcal{D}l_1 \arrow[r, "\mathcal{D}t_2"] & \mathcal{D}l_2 \arrow[r] & \ldots \arrow[r] & \mathcal{D}l_{m-2} \arrow[r, "\mathcal{D}t_{m-1}"] & \mathcal{D}l_{m-1} \arrow[ur, "\mathcal{D}t_m"'] &
\end{tikzcd}$$

compose to the same arrow in $\mathcal{C}$. This means that

$$\mathcal{D}s_n \circ \mathcal{D}s_{n-1} \circ \ldots \circ \mathcal{D}s_1 = \mathcal{D}t_m \circ \mathcal{D}t_{m-1} \circ \ldots \circ \mathcal{D}t_1.$$

1.7.14. **Remark** (Much ado about nothing)**.** There is one subtlety to the definition of commutative diagram: what happens if one of the numbers m or n in the last diagram should happen to be 0? If, say, $m = 0$, then we interpret the above equation to be meaningful only if the nodes i and j are the same (you go nowhere on an empty path) and the meaning in this case is that

$$\mathcal{D}s_n \circ \mathcal{D}s_{n-1} \circ \ldots \circ \mathcal{D}s_1 = \mathrm{id}_{\mathcal{D}i}$$

(you do nothing on an empty path). In particular, a diagram $\mathcal{D}$ based on the graph

$$\begin{tikzcd}
\mathrm{i} \arrow[loop above, "\mathrm{e}"]
\end{tikzcd}$$

commutes if and only if $\mathcal{D}(e)$ is the identity arrow from $\mathcal{D}(i)$ to $\mathcal{D}(i)$. Note, and note well, that both shape graphs

$$\begin{tikzcd}
\mathrm{i} \arrow[loop above, "\mathrm{e}"]
\end{tikzcd} \qquad \text{and} \qquad \begin{tikzcd} i \arrow[r, "d"] & j \end{tikzcd}$$

have models that one might think to represent the diagram

$$\begin{tikzcd}
\mathrm{A} \arrow[loop above, "\mathrm{f}"]
\end{tikzcd}$$

but the diagram based on the first graph commutes if and only if $f = \mathrm{id}_A$, while the diagram based on the second graph commutes automatically (no two nodes have more than one path between them so the commutativity condition is vacuous). We will always picture diagrams so that distinct nodes of the shape graph are represented by distinct (but possibly identically labeled) nodes in the picture. Thus a diagram based on the second graph in which d goes to f and i and j both go to A will be pictured as

$$A \xrightarrow{f} A.$$

In consequence, one can always deduce the shape graph of a diagram from the way it is pictured, except of course for the actual names of the nodes and arrows of the shape graph.

1.7.15. **Example** (Examples of commutative diagrams – and others)**.** The prototypical commutative diagram is the triangle

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "h"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

that commutes if and only if h is the composite $g \circ f$. The reason this is prototypical is that any commutative diagram – unless it involves an empty path – can be replaced by a set of commutative triangles. This fact is easy to show and not particularly enlightening, so we are content to give an example. The diagram

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & B \arrow[d, "g"] \\ C \arrow[r, "k"'] & D \end{tikzcd}$$

commutes if and only if the two diagrams

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, "g \circ h"] & \\ C \arrow[r, "k"'] & D \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "h"] \arrow[dr, "k \circ f"'] & B \arrow[d, "g"] \\ & D \end{tikzcd}$$

commute (in fact if and only if either one does).

1.7.16. **Definition.** Let $\mathcal{C}$ be a category. A morphism $f : X \longrightarrow Y$ is called an *isomorphism*, if there exists a morphism $g : Y \longrightarrow X$ with $g \circ f = 1_X$ and $f \circ g = 1_Y$.

g is uniquely determined by f and is itself an isomorphism. We use the notation $f^{-1} := g$. Thus we have $(f^{-1})^{-1} = g^{-1} = f$.

1.7.17. **Example.** An arrow $f : A \longrightarrow B$ is an isomorphism with inverse $g : B \longrightarrow A$ if and only if

$$\begin{tikzcd} A \arrow[r, shift left, "f"] & B \arrow[l, shift left, "g"] \end{tikzcd}$$

commutes. The reason for this is that for this diagram to commute, the two paths () and (g, f) from A to A must compose to the same value in the diagram, which means that $g \circ f = 1_A$. A similar observation shows that $f \circ g$ must be 1_B.

20.05.04

1.7.4. *Graph homomorphisms by commutative diagrams.*

1.7.18. **Remark.** The definition of graph homomorphism in 1.1.2 can be expressed by a commutative diagram. Let $\varphi = (\varphi_0; \varphi_1)$ be a graph homomorphism from $\mathcal{G}$ to $\mathcal{H}$. For any arrow $u : m \longrightarrow n$ in $\mathcal{G}$, 1.1.2 requires that $\varphi_1(u) : \varphi_0(m) \longrightarrow \varphi_0(n)$ in $\mathcal{H}$. This says that $\varphi_0(\text{source}(u)) = \text{source}(\varphi_1(u))$, and a similar statement about targets. In other words, these diagrams must commute:

$$
\begin{tikzcd}
\mathcal{G}_1 \arrow[r, "\varphi_1"] \arrow[d, "\text{source}"'] & \mathcal{H}_1 \arrow[d, "\text{source}"] \\
\mathcal{G}_0 \arrow[r, "\varphi_0"'] & \mathcal{H}_0
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{G}_1 \arrow[r, "\varphi_1"] \arrow[d, "\text{target}"'] & \mathcal{H}_1 \arrow[d, "\text{target}"] \\
\mathcal{G}_0 \arrow[r, "\varphi_0"'] & \mathcal{H}_0
\end{tikzcd}
\tag{4}
$$

In these two diagrams the two arrows labeled 'source' are of course different functions; one is the source function for $\mathcal{G}$ and the other for $\mathcal{H}$. A similar remark is true of 'target'.
This point of view provides a pictorial proof that the composite of two graph homomorphisms is a graph homomorphism (see 1.3.8(26)). If $\varphi : \mathcal{G} \longrightarrow \mathcal{H}$ and $\psi : \mathcal{H} \longrightarrow \mathcal{K}$ are graph homomorphisms, then to see that $\psi \circ \varphi$ is a graph homomorphism requires checking that the outside rectangle below commutes, and similarly with target in place of source:

$$
\begin{tikzcd}
\mathcal{G}_1 \arrow[r, "\varphi_1"] \arrow[d, "\text{source}"'] & \mathcal{H}_1 \arrow[r, "\psi_1"] \arrow[d, "\text{source}"'] & \mathcal{K}_1 \arrow[d, "\text{source}"] \\
\mathcal{G}_0 \arrow[r, "\varphi_0"'] & \mathcal{H}_0 \arrow[r, "\psi_0"'] & \mathcal{K}_0
\end{tikzcd}
\tag{5}
$$

The outside rectangle commutes because the two squares commute. This can be checked by tracing (mentally or with a finger or pointer) the paths from $\mathcal{G}_1$ to $\mathcal{K}_0$ to verify that $\text{source} \circ \psi_1 \circ \varphi_1 = \psi_0 \circ \text{source} \circ \varphi_1$ because the right square commutes, and that $\psi_0 \circ \text{source} \circ \varphi_1 = \psi_0 \circ \varphi_0 \circ \text{source}$ because the left square commutes. The verification process just described is called '*chasing the diagram*'. Of course, one can verify the required fact by writing the equations down, but those equations hide the source and target information given in diagram and thus provide a possibility of writing an impossible composite down. For many people, the diagram is much easier to remember than the equations. However, diagrams are more than informal aids; they are formally-defined mathematical objects just like automata and categories. The proof in 1.4.1 that the composition of arrows in a slice gives another arrow in the category can be represented by a similar diagram:

$$
\begin{tikzcd}
C \arrow[r, "h"] \arrow[dr, "f"'] & C' \arrow[r, "h'"] \arrow[d, "f'"] & C'' \arrow[dl, "f''"] \\
& A &
\end{tikzcd}
$$

These examples are instances of pasting commutative diagrams together to get bigger ones.

1.7.5. *Associativity by commutative diagrams.*
The fact that the multiplication in a monoid or semigroup is associative can be expressed as the assertion that a certain diagram in **Set** commutes.
Let S be a semigroup. Define the following functions:

(1) $\text{mult} : S \times S \longrightarrow S$ satisfies $\text{mult}(x, y) = xy$.

(2) $S \times \text{mult} : S \times S \times S \longrightarrow S \times S$ satisfies $(S \times \text{mult})(x, y, z) = (x, yz)$.
(3) $\text{mult} \times S : S \times S \times S \longrightarrow S \times S$ satisfies $(\text{mult} \times S)(x, y, z) = (xy, z)$.

That the following diagram commutes is exactly the associative law.

$$\begin{tikzcd} S \times S \times S \arrow[r, "S \times \text{mult}"] \arrow[d, "\text{mult} \times S"'] & S \times S \arrow[d, "\text{mult}"] \\ S \times S \arrow[r, "\text{mult}"'] & S \end{tikzcd}$$

Normally, associativity is expressed by the equation $x(yz) = (xy)z$ for all x, y, z in the semigroup. The commutative diagram expresses this same fact without the use of variables. Of course, we did use variables in defining the functions involved, but we remedy that deficiency in Chapter 6 when we give a categorical definition of products. Another advantage of using diagrams to express equations is that diagrams show the source and target of the functions involved. This is not particularly compelling here but in other situations the two-dimensional picture of the compositions involved makes it much easier to follow the discussion.

1.7.19. **Remark** (Functors by commutative diagrams)**.** We express the definition of functor using commutative diagrams. Let $\mathcal{C}$ and $\mathcal{D}$ be categories with sets of objects $\mathcal{C}_0$ and $\mathcal{D}_0$, sets of arrows $\mathcal{C}_1$ and $\mathcal{D}_1$, sets of composable pairs of arrows $\mathcal{C}_2$ and $\mathcal{D}_2$, and identity morphisms given by $\text{id} : \mathcal{C}_0 \longrightarrow \mathcal{C}_1$ and $\text{id} : \mathcal{D}_0 \longrightarrow \mathcal{D}_1$, respectively. A functor $\mathcal{F} : \mathcal{C} \longrightarrow \mathcal{D}$ consists of functions $\mathcal{F}_0 : \mathcal{C}_0 \longrightarrow \mathcal{D}_0$, $\mathcal{F}_1 : \mathcal{C}_1 \longrightarrow \mathcal{D}_1$ along with the uniquely determined function $\mathcal{F}_2 : \mathcal{C}_2 \longrightarrow \mathcal{D}_2$ (think of $\mathcal{F}_2((f, g)) = (\mathcal{F}_1(f), \mathcal{F}_1(g))$) such that

$$\begin{tikzcd} & \mathcal{C}_2 \arrow[dl, "\text{proj}_1"'] \arrow[d, "\mathcal{F}_2"] \arrow[dr, "\text{proj}_2"] & \\ \mathcal{C}_1 \arrow[d, "\mathcal{F}_1"'] & \mathcal{D}_2 \arrow[dl, "\text{proj}_1"'] \arrow[dr, "\text{proj}_2"] & \mathcal{C}_1 \arrow[d, "\mathcal{F}_1"] \\ \mathcal{D}_1 & & \mathcal{D}_1 \end{tikzcd}$$

commutes. In addition, the following diagrams must commute:

$$\begin{tikzcd} \mathcal{C}_1 \arrow[r, "\text{dom}"] \arrow[d, "\mathcal{F}_1"'] & \mathcal{C}_0 \arrow[d, "\mathcal{F}_0"'] & \mathcal{C}_1 \arrow[l, "\text{cod}"'] \arrow[d, "\mathcal{F}_1"] \\ \mathcal{D}_1 \arrow[r, "\text{dom}"'] & \mathcal{D}_0 & \mathcal{D}_1 \arrow[l, "\text{cod}"] \end{tikzcd} \qquad \begin{tikzcd} \mathcal{C}_2 \arrow[r, "\text{comp}"] \arrow[d, "\mathcal{F}_2"'] & \mathcal{C}_1 \arrow[d, "\mathcal{F}_1"] \\ \mathcal{D}_2 \arrow[r, "\text{comp}"'] & \mathcal{D}_1 \end{tikzcd} \qquad \begin{tikzcd} \mathcal{C}_0 \arrow[r, "\text{id}"] \arrow[d, "\mathcal{F}_0"'] & \mathcal{C}_1 \arrow[d, "\mathcal{F}_1"] \\ \mathcal{D}_0 \arrow[r, "\text{id}"'] & \mathcal{D}_1 \end{tikzcd}$$

where the second diagram represents $\mathcal{F}_1(f \circ g) = \mathcal{F}_1(f) \circ \mathcal{F}_1(g)$ and the third diagram represents $\mathcal{F}_1(1_A) = 1_{\mathcal{F}_0(A)}$.

1.7.20. **Remark** (Diagrams as functors). In much of the categorical literature, a diagram in a category $\mathcal{C}$ is a functor $\mathcal{D} : \mathcal{E} \longrightarrow \mathcal{C}$ where $\mathcal{E}$ is a category. Because of Proposition 1.6.8, a graph homomorphism into a category extends uniquely to a functor based on the free category generated by the graph, so that diagrams in our sense generate diagrams in the functorial sense. On the other hand, any functor is a graph homomorphism on the underlying graph of its domain (although not conversely!), so that every diagram in the sense of functor is a diagram in the sense of graph homomorphism.

21.05.04

1.7.6. *Natural transformations.*

We saw that diagrams in a category are graph homomorphisms to the category from a different point of view. Now we introduce a third way to look at graph homomorphisms to a category, namely as models. To give an example, we need a definition.

1.7.21. **Definition.** A *unary operation* on a set S is a function $u : S \longrightarrow S$.

This definition is by analogy with the concept of binary operation on a set. A set with a unary operation is a (very simple) algebraic structure, which we call a u-structure. If the set is S and the operation is $f : S \longrightarrow S$, we say that (S, f) is a u-structure, meaning (S, f) denotes a u-structure whose underlying set is S, and whose unary operation is f. This uses positional notation in much the same way as procedures in many computer languages: the first entry in the expression '(S, f)' is the name of the underlying set of the u-structure and the second entry is the name of its operation.

A homomorphism of u-structures should be a function which preserves the structure. There is really only one definition that is reasonable for this idea: if (S, u) and (T, v) are u-structures, $f : S \longrightarrow T$ is a *homomorphism of u-structures* if $f(u(s)) = v(f(s))$ for all $s \in S$. Thus this diagram must commute:

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "u"'] & T \arrow[d, "v"] \\ S \arrow[r, "f"'] & T \end{tikzcd}$$

It is not difficult to show that the composite of two homomorphisms of u-structures is another one, and that the identity map is a homomorphism, so that u-structures and homomorphisms form a category.

We now use the concept of u-structure to motivate the third way of looking at graph homomorphisms to a category.

Let $\mathcal{U}$ be the graph with one node u_0 and one arrow e:

$$\begin{tikzcd} u_0 \arrow[loop above, "e"] \end{tikzcd}$$

Let us define a graph homomorphism $D : \mathcal{U} \longrightarrow \mathbf{Set}$ as follows: $D(u_0) = \mathbb{R}$ and $D(e) = x \mapsto x^2$. Now $(\mathbb{R}, x \mapsto x^2)$ is a u-structure, and the notation we have introduced in 1.7.21 tells us that we have chosen $\mathbb{R}$ to be its underlying set and $x \mapsto x^2$ to be its unary operation. Except for the arbitrary names 'u_0' and 'e', the graph homomorphism D communicates the same information: '$\mathbb{R}$ is the particular set chosen to be the value of u_0, and $x \mapsto x^2$ is the particular function chosen to be the value of e.' In this sense, a u-structure is essentially the same thing as a diagram in $\mathbf{Set}$ of shape $\mathcal{U}$: a u-structure 'models' the graph $\mathcal{U}$. This suggests the following definition.

1.7.22. **Definition.** A *model M of a graph* $\mathcal{G}$ is a graph homomorphism $M : \mathcal{G} \longrightarrow \mathbf{Set}$.

A monoid can be defined as a model involving a graph homomorphism (and other ingredients) using the concept of finite product. We had to introduce u-structures here to have an example for which we had the requisite techniques.

1.7.23. **Example.** As another example, consider this graph:

$$a \underset{\text{target}}{\overset{\text{source}}{\rightrightarrows}} n$$

A model M of this graph consists of sets $\mathcal{G}_0 = M(n)$ and $\mathcal{G}_1 = M(a)$ together with functions source $= M(\text{source}) : \mathcal{G}_1 \longrightarrow \mathcal{G}_0$ and target $= M(\text{target}) : \mathcal{G}_1 \longrightarrow \mathcal{G}_0$. To understand what this structure is, imagine a picture in which there is a dot corresponding to each element of G_0 and an arrow corresponding to each element $a \in G_1$ which goes from the dot corresponding to source(a) to the one corresponding to target(a). It should be clear that the picture so described is a graph and thus the given graph is a graph whose models are graphs!

1.7.24. **Remark** (Models in arbitrary categories)**.** The concept of model can be generalized to arbitrary categories: if $\mathcal{C}$ is any category, a model of $\mathcal{G}$ in $\mathcal{C}$ is a graph homomorphism from $\mathcal{G}$ to $\mathcal{C}$. For example, a model of the graph for u-structures in the category of posets and monotone maps is a poset and a monotone map from the poset to itself. In these notes, the bare word 'model' always means a model in **Set**.

1.7.25. **Remark** (Natural transformations between models of a graph)**.** In a category, there is a natural notion of an arrow from one model of a graph to another. This usually turns out to coincide with the standard definition of homomorphism for that kind of structure.

1.7.26. **Definition.** Let $D, E : \mathcal{G} \longrightarrow \mathcal{C}$ be two models of the same graph in a category. A *natural transformation* $\alpha : D \longrightarrow E$ is given by a family of arrows αa of $\mathcal{C}$ indexed by the nodes of $\mathcal{G}$ such that:

NT-1 $\alpha a : Da \longrightarrow Ea$ for each node a of $\mathcal{G}$.
NT-2 For any arrow $s : a \longrightarrow b$ in $\mathcal{G}$, the diagram

$$\begin{tikzcd} Da \arrow[r, "\alpha a"] \arrow[d, "Ds"'] & Ea \arrow[d, "Es"] \\ Db \arrow[r, "\alpha b"'] & Eb \end{tikzcd} \tag{6}$$

commutes.

The commutativity of the diagram in NT-2 is referred to as the em naturality condition on α. The arrow αa for an object a is the component of the natural transformation α at a. Note that you talk about a natural transformation from D to E only if D and E have the same domain (here $\mathcal{G}$) as well as the same codomain (here $\mathcal{C}$) and if, moreover, the codomain is a category. In this situation, it is often convenient to write $\alpha : D \longrightarrow E : \mathcal{G} \longrightarrow \mathcal{C}$.

1.7.27. **Definition.** Let D, E and F be models of $\mathcal{G}$ in $\mathcal{C}$, and $\alpha : D \longrightarrow E$ and $\beta : E \longrightarrow F$ natural transformations. The composite $\beta \circ \alpha : D \longrightarrow F$ is defined componentwise: $(\beta \circ \alpha)a = \beta a \circ \alpha a$.

1.7.28. **Proposition.** *The composite of two natural transformations is also a natural transformation.*

Proof. The diagram that has to be shown commutative is the outer rectangle of

$$\begin{tikzcd} Da \arrow[r, "\alpha a"] \arrow[d, "Ds"'] & Ea \arrow[r, "\beta a"] \arrow[d, "Es"'] & Fa \arrow[d, "Fs"] \\ Db \arrow[r, "\alpha b"'] & Eb \arrow[r, "\beta b"'] & Fb \end{tikzcd} \tag{7}$$

for each arrow $s : a \longrightarrow b$ in $\mathcal{G}$. The rectangle commutes because the two squares do; the squares commute as a consequence of the naturality of α and β. □

It is interesting that categorists began using modes of reasoning like that in the preceding proof because objects of categories generally lacked elements; now one appreciates them for their own sake because they allow element-free (and thus variable-free) arguments.

It is even easier to show that there is an identity natural transformation between any model D and itself, defined by $1_D a = 1_{Da}$. We then have the following proposition, whose proof is straightforward.

1.7.29. **Proposition.** *The models of a given graph $\mathcal{G}$ in a given category $\mathcal{C}$, and the natural transformations between them, form a category. We denote this category by* $\mathbf{Mod}(\mathcal{G}, \mathcal{C})$.

1.7.30. **Example.** The natural transformations between models in **Set** of the u-structure graph $\mathcal{U}$ (one vertex and one non-trivial arrow) are exactly the homomorphisms of u-structures defined in 1.7.21. The graph $\mathcal{U}$ has one object u_0 and one arrow e, so that a natural transformation from a model D to a model E has only one component which is a function from $D(u_0)$ to $E(u_0)$. If we set $S = D(u_0)$, $u = D(e)$, $T = E(u_0)$, $v = E(e)$, and we define $\alpha u_0 = f$, this is the single component of a natural transformation from D to E.
Condition NT-2 in 1.7.26 coincides in this case with the diagram in 1.7.21: the naturality condition is the same as the definition of homomorphism of u-structures. It follows that the category of u-structures and homomorphisms is essentially $\mathbf{Mod}(\mathcal{U}, \mathbf{Set})$.

1.7.31. **Example.** A homomorphism of graphs is a natural transformation between models of the graph

$$\begin{tikzcd} a \arrow[r, shift left, "\text{source}"] \arrow[r, shift right, "\text{target}"'] & n \end{tikzcd}$$

The two graphs in diagram (4) are the two necessary instances (one for the source and the other for the target) of the diagram (6). In a similar way, the diagram (7), used to show that the composite of two natural transformations is a natural transformation, reduces in this case to the commutativity of diagram (5): specifically, the only possibilities (other than those in which s is an identity arrow) for a and b in diagram (7) are $a = a$ and $b = n$, giving two diagrams shaped like diagram (7), one for $s =$ source (that is diagram (5)) and the other for $s =$ target.

1.7.32. **Example.** A model of the graph

$$0 \xrightarrow{u} 1 \tag{8}$$

in an arbitrary category $\mathcal{C}$ is essentially the same as an arrow in $\mathcal{C}$ (see 5.2.23 below). A natural transformation from the model represented by the arrow $f : A \longrightarrow B$ to the one represented by $g : C \longrightarrow D$ is a pair of arrows $h : A \longrightarrow C$ and $k : B \longrightarrow D$ making a commutative diagram:

$$\begin{tikzcd} Da \arrow[r, "\alpha a"] \arrow[d, "Ds"'] & Ea \arrow[d, "Es"] \\ Db \arrow[r, "\alpha b"'] & Eb \end{tikzcd} \tag{9}$$

The component at 0 is h and the component at 1 is k. The category of models in $\mathcal{C}$ is called the *arrow category* of $\mathcal{C}$; it is often denoted $C^{\longrightarrow}$.

1.7.33. **Example.** Let $\mathcal{G}$ be the graph with two nodes and no arrows, and $\mathcal{C}$ any category. Then $\mathbf{Mod}(\mathcal{G}, \mathcal{C})$ is isomorphic to $C \times C$.

1.7.7. *Natural isomorphisms.*

1.7.34. **Definition.** A natural transformation $\alpha : F \longrightarrow G : \mathcal{G} \longrightarrow \mathcal{D}$ is called a *natural isomorphism* if there is a natural transformation $\beta : G \longrightarrow F$ which is an inverse to α in the category $\mathbf{Mod}(\mathcal{G}, \mathcal{D})$. Natural isomorphisms are often called *natural equivalences*.

1.7.35. **Example.** The arrow $(h, k) : f \longrightarrow g$ in the arrow category of a category $\mathcal{C}$, as shown in (9), is a natural isomorphism if and only if h and k are both isomorphisms in $\mathcal{C}$. This is a special case of an important fact about natural isomorphisms, which we now state.

1.7.36. **Theorem.** *Suppose $F : \mathcal{G} \longrightarrow \mathcal{D}$ and $G : \mathcal{G} \longrightarrow \mathcal{D}$ are models of $\mathcal{G}$ in $\mathcal{D}$ and $\alpha : F \longrightarrow G$ is a natural transformation of models. Then α is a natural isomorphism if and only if for each node a of $\mathcal{G}$, $\alpha a : F(a) \longrightarrow G(a)$ is an isomorphism of $\mathcal{D}$.*

Proof. Suppose α has an inverse $\beta : G \longrightarrow F$ in $\mathrm{Mod}(\mathcal{G}, \mathcal{D})$. Then for any node a, by definition 1.7.27 the definition of the identity natural transformation, and the definition of inverse,

$$\alpha a \circ \beta a = (\alpha \circ \beta)a = \mathrm{id}_G\, a = \mathrm{id}_{G(a)}$$

and

$$\beta a \circ \alpha a = (\beta \circ \alpha)a = \mathrm{id}_F\, a = \mathrm{id}_{F(a)}$$

which means that the arrow βa is the inverse of the arrow αa, so that αa is an isomorphism in $\mathcal{D}$ as required.

Conversely, suppose that for each node a of $\mathcal{G}$, $\alpha a : F(a) \longrightarrow G(a)$ is an isomorphism of $\mathcal{D}$. The component of the inverse β at a node a is defined by letting $\beta a = (\alpha a)^{-1}$. This is the only possible definition, but it must be shown to be natural. Let $f : a \longrightarrow b$ be an arrow of the domain of F and G. Then we have

$$\begin{aligned} Ff \circ (\alpha a)^{-1} &= (\alpha b)^{-1} \circ (\alpha b) \circ Ff \circ (\alpha a)^{-1} \\ &= (\alpha b)^{-1} \circ Gf \circ (\alpha a) \circ (\alpha a)^{-1} \\ &= (\alpha b)^{-1} \circ Gf. \end{aligned}$$

27.05.04 which says that β is natural. The second equality uses the naturality of α. □

1.7.37. **Remark.** In 1.7.32, we said that a model in an arbitrary category $\mathcal{C}$ of the graph (8) is 'essentially the same' as an arrow in $\mathcal{C}$. This is common terminology and usually refers implicitly to an equivalence of categories. We spell it out in this case.

Let us say that for a category $\mathcal{C}$, $\mathcal{C}'$ is the category whose objects are the arrows of $\mathcal{C}$ and for which an arrow from f to g is a pair (h, k) making diagram (9) commute.

A model M of the graph (8) in a category $\mathcal{C}$ specifies the objects $M(0)$ and $M(1)$ and the arrow $M(u)$. $M(u)$ has domain $M(0)$ and codomain $M(1)$. But the domain and codomain of an arrow in a category are uniquely determined by the arrow. So that the only necessary information is which arrow $M(u)$ is.
Now we can define a functor $F : \mathcal{C}^{\longrightarrow} \longrightarrow \mathcal{C}'$. On objects it takes M to $M(u)$. The remarks in the preceding paragraph show that this map on objects is bijective. If $M(u) = f$ and $N(u) = g$, an arrow from M to N in $\mathcal{C}^{\longrightarrow}$ and an arrow from f to g in $\mathcal{C}'$ are the same thing – a pair (h, k) making diagram (9) commute. So we say F is the identity on arrows. It is straightforward to see that F is actually an isomorphism of categories.

1.7.8. *Natural transformations between functors II.*

1.7.38. **Definition.** A functor is among other things a graph homomorphism, so a *natural transformation between two functors* is a natural transformation of the corresponding graph homomorphisms. The following proposition is an immediate consequence of 1.7.28.

1.7.39. **Proposition.** *If $\mathcal{C}$ and $\mathcal{D}$ are categories, the functors from $\mathcal{C}$ to $\mathcal{D}$ form a category with natural transformations as arrows.*

We denote this category by $\mathbf{Func}(\mathcal{C}, \mathcal{D})$. Other common notations for it are $\mathcal{D}^{\mathcal{C}}$ and $[\mathcal{C}, \mathcal{D}]$. Tennent [1986] provides an exposition of the use of functor categories for programming language semantics. Of course, the graph homomorphisms from $\mathcal{C}$ to $\mathcal{D}$, which do not necessarily preserve the composition of arrows in $\mathcal{C}$, also form a category $\mathbf{Mod}(\mathcal{C}, \mathcal{D})$ (see 1.7.29), of which $\mathbf{Func}(\mathcal{C}, \mathcal{D})$ is a full subcategory. A natural transformation from one functor to another is a special case of a natural transformation from one graph homomorphism to another, so the ideas we have presented concerning natural transformations between graph homomorphisms apply to natural transformations between functors as well. In particular, Theorems 1.7.28 and 1.7.36 are true of natural transformations of functors.
If $\mathcal{C}$ is not a small category (see 1.3.4), then $\mathbf{Func}(\mathcal{C}, \mathcal{D})$ may not be locally small (see 1.3.4). In fact functors will be large, too, so they cannot be elements (objects) of a class of objects. This is a rather esoteric question that will not concern us in these notes since we will have no occasion to form functor categories of that sort.
We motivated the concept of natural transformation by considering models of graphs, and most of the discussion in the rest of this section concerns that point of view. Historically, the concept first arose for functors and not from the point of view of models.

1.7.40. **Examples.** We have already described some examples of natural transformations, as summed up in the following propositions. In 1.6.7, we defined the graph homomorphism $\eta\mathcal{G} : \mathcal{G} \longrightarrow U(F(\mathcal{G}))$ which includes a graph $\mathcal{G}$ into $U(F(\mathcal{G}))$, the underlying graph of the free category $F(\mathcal{G})$.

1.7.41. **Proposition.** *The family of arrows $\eta\mathcal{G}$ form a natural transformation from the identity functor on* **Grf** *to $U \circ F$, where U is the underlying graph functor from* **Cat** *to* **Grf**.

The proof is left as an exercise.
Further examples of natural transformations are given by actions on sets. We begin with a definition.

1.7.42. **Definition.** Let M be a monoid with identity 1 and let S be a set. An *action of M on S* is a function $\alpha : M \times S \longrightarrow S$ for which

A-1 $\alpha(1, s) = s$ for all $s \in S$.

A-2 $\alpha(mn, s) = \alpha(m, \alpha(n, s))$ for all $m, n \in M$ and $s \in S$.

It is customary in mathematics to write ms for $\alpha(m, s)$; then the preceding requirements become

A'-1 $1s = s$ for all $s \in S$.
A'-2 $(mn)s = m(ns)$ for all $m, n \in M$ and $s \in S$.

When actions are written this way, S is also called an *M-set.* The same syntax ms for $m \in M$ and $s \in S$ is used even when different actions are involved. This notation is analogous to (and presumably suggested by) the notation cv for scalar multiplication, where c is a scalar and v is a vector.
It is useful to think of the set S as a state space and the elements of M as acting to induce transitions from one state to another.

1.7.43. **Example.** One major type of action by a monoid is the case when the state space is a vector space and M is a collection of linear transformations closed under multiplication. However, in that case the linear structure (the fact that states can be added and multiplied by scalars) is extra structure which the definition above does not require. Our definition also does not require that there be any concept of continuity of transitions. Thus, the definition is very general and can be regarded as a nonlinear, discrete approach to state transition systems. Less structure means, as always, that fewer theorems are true and fewer useful tools are available. On the other hand, less structure means that more situations fit the axioms, so that the theorems that are true and the tools that do exist work for more applications.

1.7.44. **Definition.** Let M be a monoid with actions on sets S and T. An *equivariant map* from S to T is a function $\varphi : S \longrightarrow T$ with the property that $m\varphi(s) = \varphi(ms)$ for all $m \in M$ and $s \in S$. The identity function is an equivariant map and the composite of two equivariant maps is equivariant. This means that for each monoid M, monoid actions and equivariant maps form a category M- **Act**.

1.7.45. **Remark.** Let α be an action of a monoid M on a set S. Let $\mathcal{C}(M)$ denote the category determined by M with one object and morphisms the elements of M. The action α determines a functor $F_\alpha : \mathcal{C}(M) \longrightarrow \mathbf{Set}$ defined by:

AF-1 $F_\alpha(*) = S$.
AF-2 $F_\alpha(m) = s \mapsto \alpha(m, s)$ for $m \in M$ and $s \in S$.

1.7.46. **Example.** Let $\alpha : M \times S \longrightarrow S$ and $\beta : M \times T \longrightarrow T$ be two actions by a monoid $\mathcal{M}$. Let $\varphi : S \longrightarrow T$ be an equivariant map. If F and G are the functors corresponding to α and β, as in 1.7.45, then φ is the (only) component of a natural transformation from F to G. Conversely, the only component of any natural transformation from F to G is an equivariant map between the corresponding actions.
If $\varphi : S \longrightarrow T$ is an isomorphism of actions by M then φ determines a natural isomorphism.

1.7.47. **Example.** Let $U : \mathbf{Mon} \longrightarrow \mathbf{Set}$ be the underlying functor from the category of monoids. Define $U \times U : \mathbf{Mon} \longrightarrow Set$ as follows:

(1) For a monoid M, $(U \times U)(M) = U(M) \times U(M)$.
(2) For a monoid homomorphism $h : M \longrightarrow N$,

$$(U \times U)(h)(m, n) = (h(m), h(n)).$$

Then monoid multiplication is a natural transformation from $U \times UtoU$.
Formally: Let $\mu : U \times U \longrightarrow U$ be the family of maps whose value at a monoid M is the function $\mu M : U(M) \times U(M) \longrightarrow U(M)$ defined by $\mu M(m, m') = mm'$, the product of m and m' in M. Then μ is a natural transformation. (The function μM is *not* in general a monoid homomorphism, unless M is commutative.)
It is instructive to see why μ is a natural transformation. Let $h : M \longrightarrow N$ be a monoid homomorphism. We must show that the following diagram commutes:

$$\begin{tikzcd}
(U \times U)(M) \arrow[r, "\mu M"] \arrow[d, "(U \times U)(h)"'] & U(M) \arrow[d, "h"] \\
(U \times U)(N) \arrow[r, "\mu N"'] & U(N)
\end{tikzcd} \tag{10}$$

The top route takes an element $(m, m') \in (U \times U)(M)$ to $h(mm')$. The lower route takes it to $h(m)h(m')$. The commutativity of the diagram then follows from the fact that h is a homomorphism.

1.7.48. **Definition.** Let $\mathcal{C}$ be a category. A *subfunctor* of a functor $F : \mathcal{C} \longrightarrow \mathbf{Set}$ is a functor $G : \mathcal{C} \longrightarrow \mathbf{Set}$ with the property that for each object C of $\mathcal{C}$, $G(C) \subseteq F(C)$ and such that for each arrow $f : C \longrightarrow C'$ and each element $x \in G(C)$, we have that $Gf(x) = Ff(x)$. It is straightforward to check that the inclusion function $i_C : G(C) \longrightarrow F(C)$ is a natural transformation.

1.7.49. **Example.** Let B be a fixed set. We define a functor $R : \mathbf{Set}^{op} \longrightarrow \mathbf{Set}$ such that for a set A, $R(A) = \mathbf{Rel}(A, B)$, the set of relations from A to B. (A *relation* from A to B is essentially set of ordered pairs in $A \times B$.) For a set function $f : A' \longrightarrow A$ and relation $\alpha \in \mathbf{Rel}(A, B)$, define $R(f)(\alpha)$ to be the relation $\alpha' \in \mathbf{Rel}(A', B)$ defined by $a'\alpha'b$ if and only if $f(a')\alpha b$. It is easy to see that this makes $R : \mathbf{Set}^{op} \longrightarrow \mathbf{Set}$ a functor. (Note that $R(A) = \mathbf{Rel}(A, B)$, but R is *not* $\mathrm{Mor}_{\mathbf{Rel}}(\text{-}, B)$.)
For each A, let $\varphi_A : \mathbf{Rel}(A, B) \longrightarrow \mathrm{Mor}(A, \mathcal{P}B)$ be the bijection that takes a relation α from A to B to the function that takes an element $a \in A$ to the set $\{b \in B | a\alpha b\}$. (You should check that this is a bijection.) If we check that the functions φ_A are the components of a natural transformation from R to $\mathrm{Mor}(A, \mathcal{P}B)$, the transformation will automatically be a natural isomorphism by Theorem 1.7.36. To show that it is natural, let $\alpha \in \mathbf{Rel}(A, B)$ and $a' \in A'$. Then

$$\mathrm{Mor}(f, \mathcal{P}B)(\varphi_A(\alpha)(a')) = \varphi_A(\alpha)(f(a')) = \{b | f(a')\alpha b\} = \{b | a'\alpha' b\} = \varphi_{A'}(R(f)(a'))$$

as required.
This natural isomorphism can be taken to be the defining property of a topos. A *topos* is a cartesian closed category with some extra structure which produces an object of subobjects for each object. This structure makes toposes very much like the category of sets.

1.7.50. **Example.** For each set S, let $\{\}S : S \longrightarrow \mathcal{P}S$ be the function which takes an element x of S to the singleton subset $\{x\}$. Then fg is a natural transformation from the identity functor on **Set** to the direct image powerset functor $\mathcal{P}$. (See 1.6.3.) However, fg is not a natural transformation from the identity functor on **Set** to the universal image powerset functor (see 1.6.11), and it does not even make sense to ask whether it is a natural transformation to the inverse image powerset functor.

1.7.9. *Natural transformations of graphs.*
We now consider some natural transformations involving the category **Grf** of graphs and homomorphisms of graphs.

1.7.51. **Example.** The map which takes an arrow of a graph to its source is a natural transformation from A to N. (See 1.6.2.) The same is true for targets. Actually, every operation in any multisorted algebraic structure gives a natural transformation. Example 1.7.47 was another example of this.

1.7.52. **Example.** In 1.6.2, we defined the functor $N : \mathbf{Grf} \longrightarrow \mathbf{Set}$. It takes a graph $\mathcal{G}$ to its set G_0 of nodes and a homomorphism φ to φ_0. Now pick a graph with one node $*$ and no arrows and call it $\mathcal{E}$. Let $V = \mathrm{Mor}_{\mathbf{Grf}}(\mathcal{E}, \text{-})$.
A graph homomorphism from the graph $\mathcal{E}$ to an arbitrary graph $\mathcal{G}$ is evidently determined by the image of $\mathcal{E}$ and that can be any node of $\mathcal{G}$. In other words, nodes of $\mathcal{G}$ are 'essentially the same thing' as graph homomorphisms from $\mathcal{E}$ to $\mathcal{G}$, that is, as the elements of the set $V(\mathcal{G})$.
We can define a natural transformation $\alpha : V \longrightarrow N$ by defining $\alpha\mathcal{G}(f) = f_0(*)$ where $\mathcal{G}$ is a graph and $f : \mathcal{E} \longrightarrow \mathcal{G}$ is a graph homomorphism (arrow of **Grf**). There must be a naturality diagram (6) for each arrow of the source category, which in this case is **Grf**. Thus to see that α is natural, we require that for each graph homomorphism $g : \mathcal{G}_1 \longrightarrow \mathcal{G}_2$, the diagram

$$\begin{tikzcd} V\mathcal{G}_1 \arrow[r, "Vg"] \arrow[d, "\alpha\mathcal{G}_1"'] & V\mathcal{G}_2 \arrow[d, "\alpha\mathcal{G}_2"] \\ N\mathcal{G}_1 \arrow[r, "Ng"'] & N\mathcal{G}_2 \end{tikzcd} \tag{11}$$

commutes. Now Ng is g_0 (the node map of g) by definition, and the value of V (which is a hom functor) at a homomorphism g composes g with a graph homomorphism from the graph $\mathcal{E}$. Then we have, for a homomorphism $f : E \longrightarrow \mathcal{G}_1$ (i.e., an element of the upper left corner of the diagram),

$$(\alpha\mathcal{G}_2 \circ Vg)(f) = \alpha\mathcal{G}_2(g \circ f) = (g \circ f)_0(*)$$

while

$$(Ng \circ \alpha\mathcal{G}_1)(f) = Ng(f_0(*)) = g_0(f_0(*))$$

and these are equal from the definition of composition of graph homomorphisms.
The natural transformation α is in fact a natural isomorphism. This shows that N is naturally isomorphic to a hom functor. Such functors are called *representable*, and are considered in greater detail in 2.1.1

1.7.53. **Remark** (Connected components)**.** A node a can be *connected* to the node b of a graph $\mathcal{G}$ if it is possible to get from a to b following a sequence of arrows of $\mathcal{G}$ in either direction. In order to state this more precisely, let us say that an arrow a 'has' a node n if n is the domain or the codomain (or both) of a. Then a is *connected* to b means that there is a sequence $(c_0, c_1, \ldots, c_n)$ of arrows of $\mathcal{G}$ with the property that a is a node (either the source or the target) of c_0, b is a node of c_n, and for $i = 1, \ldots, n$, c_{i-1} and c_i have a node in common. We call such a sequence an *undirected path between a and b*.
It is a good exercise to see that 'being connected to' is an equivalence relation. (For reflexivity: a node is connected to itself by the empty sequence.) An equivalence class of nodes

with respect to this relation is called a *connected component of the graph* $\mathcal{G}$, and the set of connected components is called $W\mathcal{G}$.
Connected components can be defined for categories in the same way as for graphs. In that case, each connected component is a full subcategory.
If $f : \mathcal{G} \longrightarrow \mathcal{H}$ is a graph homomorphism and if two nodes a and b are in the same component of $\mathcal{G}$, then $f(a)$ and $f(b)$ are in the same component of $\mathcal{H}$, this is because f takes an undirected path between a and b to an undirected path between $f(a)$ and $f(b)$. Thus the arrow f induces a function $Wf : W\mathcal{G} \longrightarrow W\mathcal{H}$, namely the one which takes the component of a to the component of $f(a)$; and this makes W a functor from **Grf** to **Set**.
For a graph $\mathcal{G}$, let $\beta\mathcal{G} : N\mathcal{G} \longrightarrow W\mathcal{G}$ be the set function which takes a node of $\mathcal{G}$ to the component of $\mathcal{G}$ that contains that node. (The component is the value of $\beta\mathcal{G}$ at the node, not the codomain.) Then $\beta : N \longrightarrow W$ is a natural transformation. It is instructive to check the commutativity of the requisite diagram.

1.7.10. *Combining natural transformations and functors.*

1.7.54. **Remark** (Composites of natural transformations)**.** Let $F : \mathcal{A} \longrightarrow \mathcal{B}$ and $G : \mathcal{B} \longrightarrow \mathcal{C}$ be functors. There is a composite functor $G \circ F : \mathcal{A} \longrightarrow \mathcal{C}$ defined in the usual way by $G \circ F(A) = G(F(A))$. Similarly, let H, K and L be functors from $\mathcal{A}$ to $\mathcal{B}$ and $\alpha : H \longrightarrow K$ and $\beta : K \longrightarrow L$ be natural transformations. Recall that this means that for each object A of $\mathcal{A}$, $\alpha A : HA \longrightarrow KA$ and $\beta A : KA \longrightarrow LA$. Then as in **??**.2.11, we define $\beta \circ \alpha : H \longrightarrow L$ by $(\beta \circ \alpha)A = \beta A \circ \alpha A$.

1.7.55. **Remark** (Functors and natural transformations)**.** Things get more interesting when we mix functors and natural transformations. For example, suppose we have three categories $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, four functors, two of them, $F, G : \mathcal{A} \longrightarrow \mathcal{B}$ and the other two $H, K : \mathcal{B} \longrightarrow \mathcal{C}$, and two natural transformations $\alpha : F \longrightarrow G$ and $\beta : H \longrightarrow K$. We picture this situation as follows:

(12)
$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{B} \arrow[r, bend left, "H"] \arrow[r, bend right, "K"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathcal{C} \end{tikzcd}$$

1.7.56. **Definition.** The natural transformation $\beta F : H \circ F \longrightarrow K \circ F$ is defined by the formula $(\beta F)A = \beta(FA)$ for an object A of $\mathcal{A}$. The notation $\beta(FA)$ means the component of the natural transformation β at the object FA. This is indeed an arrow $H(F(A)) \longrightarrow K(F(A))$ as required. To show that βF is natural requires showing that for an arrow $f : A \longrightarrow A'$ of $\mathcal{A}$, the diagram

(13)
$$\begin{tikzcd} H(F(A)) \arrow[r, "\beta FA"] \arrow[d, "H(F(f))"'] & K(F(A)) \arrow[d, "K(F(f))"] \\ H(F(A')) \arrow[r, "\beta FA'"'] & K(F(A')) \end{tikzcd}$$

commutes, but this is just the naturality diagram of β applied to the arrow $F(f) : F(A) \longrightarrow F(A')$.

1.7.57. **Definition.** The natural transformation $H\alpha : H \circ F \longrightarrow H \circ G$ is defined by letting $(H\alpha)A = H(\alpha A)$ for an object A of $\mathcal{A}$, that is the value of $\mathcal{H}$ applied to the arrow αA. To see that $H\alpha$ thus defined is natural requires showing that

$$\begin{tikzcd}
H(F(A)) \arrow[r, "H(\alpha A)"] \arrow[d, "H(F(f))"'] & H(G(A)) \arrow[d, "H(G(f))"] \\
H(F(A')) \arrow[r, "H(\alpha A')"'] & H(G(A'))
\end{tikzcd}$$

commutes. This diagram is obtained by applying the functor H to the naturality diagram of α. Since functors preserve commutative diagrams, the result follows. Note that the proofs of naturality for βF and for $H\alpha$ are quite different. For example, the second requires that H be a functor, while the first works if F is merely an object function.

2. Representable functors and the Yoneda Lemma

2.1. Representable functors.

For an arbitrary category $\mathcal{C}$, the functors from $\mathcal{C}$ to **Set** are special because the hom functors $\mathrm{Mor}(C,\text{-})$ for each object C of $\mathcal{C}$ are set-valued functors. In this section, we introduce the concept of representable functor and discuss how certain set theoretic concepts can be transferred to arbitrary categories by representable functors. The main tool for this is the Yoneda Lemma, and universal elements, all of which are based on these hom functors.
These ideas have turned out to be fundamental tools for categorists. They are also closely connected with the concept of adjunction, to be discussed later.
If you are familiar with group theory, it may be illuminating to realize that representable functors are a generalization of the regular representation, and the Yoneda embedding is a generalization of Cayley's Theorem. We have already considered set-valued functors as actions in 1.7.42.

2.1.1. **Definition.** A functor F from a category $\mathcal{C}$ to the category of sets **Set** (a set-valued functor) is said to be *representable* if there exists an object C of $\mathcal{C}$ and a natural isomorphism $\alpha : F \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C,\text{-})$ to the hom functor $\mathrm{Mor}_{\mathcal{C}}(C,\text{-})$; in this case one says that C *represents the functor.*
A *contravariant functor is representable* if there exists an object C of $\mathcal{C}$ and a natural isomorphism $\alpha : F \longrightarrow \mathrm{Mor}_{\mathcal{C}}(\text{-},C)$ to the hom functor $\mathrm{Mor}_{\mathcal{C}}(\text{-},C)$; again one says that C *represents the functor.*

We have already looked at one example of representable functor in some detail in 1.7.52, where we showed that the set-of-nodes functor for graphs is represented by the graph with one node and no arrows. The set-of-arrows functor is represented by the graph with two nodes and one arrow between them.

2.1.2. **Example.** The arrow functor $A : \mathbf{Grf} \longrightarrow \mathbf{Set}$ of 1.6.2 is a representable functor represented by the graph **2** which is pictured as

$$1 \xrightarrow{e} 2.$$

Representable functors are used to think of objects as if they were sets. The main aim is to write morphisms then as set maps.

2.1.3. **Definition.** In definition 2.5.2 and many like it, what replaces the concept of element of A is an arbitrary arrow into A. In this context, an arbitrary arrow $a : T \longrightarrow A$ is called a *variable element of A, parametrized by T*. The set of variable elements of A parametrized by T is $A(T) := \mathrm{Mor}_{\mathcal{C}}(T, A)$.
When a is treated as a variable element of A and f has source A, one may write $f(a)$ for $f \circ a$. So we may consider f as a map of variable elements:

$$f = \mathrm{Mor}_{\mathcal{C}}(T, f) : \mathrm{Mor}_{\mathcal{C}}(T, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T, B).$$

So f takes the variable elements of A with parameter T to the variable elements of B with the same parameter T.

2.1.4. **Proposition.** *If $f, g : A \longrightarrow B$ are morphisms in $\mathcal{C}$ then the following are equivalent:*
(1) $f = g$,
(2) *$f(a) = g(a)$ for all parameters T and all variable elements $a \in A(T)$.*

Proof. Clearly (1) implies (2). Assume that (2) holds. Take $T := A$ and $a = 1_A$. Then $f(a) = g(a)$ implies $f = f \circ 1_A = f(1_A) = g(1_A) = g \circ 1_A = g$. □

2.2. **Terminal and initial objects.**

One of the most fundamental objects in the category of sets is a *singleton set* $T = \{*\}$, a set with exactly one element. It has the categorical property, that there is exactly one arrow $A \longrightarrow T$ for each set A. The last property can be generalized to arbitrary categories as follows.

2.2.1. **Definition.** An object T of a category $\mathcal{C}$ is called a *terminal object* if there is exactly one arrow $A \longrightarrow T$ for each object A of $\mathcal{C}$. We usually denote the terminal object by 1 and the unique arrow $A \longrightarrow 1$ by $\langle\rangle$.
The dual notion (see 1.4.13), an object of a category that has a unique arrow to each object (including itself), is called an *initial object* and is often denoted 0.

2.2.2. **Examples.** In the category of sets, the empty set is initial and any one-element set is terminal. Thus the category of sets has a unique initial object but many terminal objects. The one-element monoid is both initial and terminal in the category of monoids.
In the category determined by a poset, an object is initial if and only if it is an absolute minimum for the poset, and it is terminal if and only if it is an absolute maximum. Since there is no largest or smallest whole number, the category determined by the set of integers with its natural order (there is an arrow from m to n if and only if $m \leq n$) gives an example of a category without initial or terminal object.
In the category of semigroups, the empty semigroup (see 1.2.9) is the initial object and any one- element semigroup is a terminal object. On the other hand, the category of nonempty semigroups does not have an initial object.
Warning: To prove this, it is not enough to say that the initial object in the category of semigroups is the empty semigroup and that semigroup is missing here! You have to show that no other object can be the initial object in the smaller category. One way to do this is to let U be the semigroup with two elements 1 and e with $1 \cdot e = e \cdot 1 = e$, $1 \cdot 1 = 1$ and $e \cdot e = e$. Then any nonempty semigroup S has two homomorphisms to U: the constant function taking everything to 1 and the one taking everything to e. Thus no nonempty semigroup S can be the initial object.
In the category of graphs and graph homomorphisms, the graph with one node and one arrow is the terminal object.

2.2.3. **Proposition.** *Any two terminal (respectively initial) objects in a category $\mathcal{C}$ are isomorphic.*

Proof. Suppose 1 and $1'$ are terminal objects. Since 1 is terminal, there is an arrow $f : 1' \longrightarrow 1$. Similarly, there is an arrow $g : 1 \longrightarrow 1'$. The arrow $f \circ g : 1 \longrightarrow 1$ is an arrow with target 1. Since 1 is a terminal object of the category, there can be only one arrow from 1 to 1. Thus it must be that $f \circ g$ is the identity of 1. An analogous proof shows that $g \circ f$ is the identity of $1'$. □

2.2.4. **Remark.** In **Set**, an element x of a set A is the image of a function from a singleton set to A that takes the unique element of the singleton to x. Thus if we pick a specific singleton $\{*\}$ and call it 1, the elements of the set A are in one to one correspondence with $\mathrm{Mor}(1, A)$, which is the set of functions from the terminal object to A. Moreover, if $f : A \longrightarrow B$ is a set function and x is an element of A determining the function $x : 1 \longrightarrow A$, then the element $f(x)$ of B is essentially the same thing as the composite $f \circ x : 1 \longrightarrow B$. Because of this, the categorist typically thinks of an element $x \in A$ as being the constant $x : 1 \longrightarrow A$.

2.2.5. **Definition.** An arrow $1 \longrightarrow A$ in a category, where 1 is the terminal object, is called a *constant of type* A.

Thus each element of a set is a constant in **Set**. On the other hand, each monoid M has just one constant $1 \longrightarrow M$ in the category of monoids, since monoid homomorphisms must preserve the identity. The more common name in the categorical literature for a constant is *global element of* A, a name that comes from sheaf theory.

2.2.6. **Example.** The identity functor on **Set** is a representable functor represented by a terminal object in **Set**; in other words, $\mathrm{Mor}_{\mathbf{Set}}(\mathbf{1}, \text{-})$ is naturally isomorphic to the identity functor. This can be described by saying that 'a set is its set of global elements.'

A terminal object is an object with exactly one arrow $\langle\rangle : A \longrightarrow 1$ to it from each object A. So the arrows to 1 are not interesting. Global elements are arrows from the terminal object. There may be none or many, so they are interesting.

If 1 and $1'$ are two terminal objects in a category and $x : 1 \longrightarrow A$ and $x' : 1' \longrightarrow A$ are two constants with the property that $x' \circ \langle\rangle = x$ (where $\langle\rangle$ is the unique isomorphism from 1 to $1'$), then we regard x and x' as the same constant. Think about this comment as it applies to elements in the category of sets, with two different choices of terminal object, and you will see why.

There is an interesting technique to generalize notions in the categories **Set** to arbitrary categories $\mathcal{C}$ by using representable functors. We apply this technique to terminal objects.

2.2.7. **Proposition.** *An object* 1 *in an arbitrary category* $\mathcal{C}$ *is a terminal object if and only if the (contravariant) representable functor* $\mathrm{Mor}_{\mathcal{C}}(\text{-}, 1) : \mathcal{C}^{op} \longrightarrow$ **Set** *maps every object* A *in* $\mathcal{C}$ *to a terminal object* $\mathrm{Mor}_{\mathcal{C}}(A, 1)$ *in* **Set**.

Proof. Since $\mathrm{Mor}_{\mathcal{C}}(A, 1)$ is a singleton for all A, this is nothing but the definition of the terminal object. □

2.2.8. **Remark.** The definition of an initial object I is dual to the definition of a terminal object, so it is a terminal object in $\mathcal{C}^{op}$. It is characterized by the fact that all $\mathrm{Mor}_{\mathcal{C}^{op}}(A, I) = \mathrm{Mor}_{\mathcal{C}}(I, A)$ are singleton sets.

28.05.04

2.3. The extension principle.

2.3.1. **Remark (Covariant extension of set-theoretic properties).** For the notion of terminal object we should proceed according to the following *principle of covariant extension of set-theoretic properties* or *extension principle*:

(1) We define a (set-theoretic) property $\mathcal{P}$ of a set A (or a map f or a diagram $D : I \longrightarrow$ **Set**) in such a way that this property holds for all sets (resp. maps resp. diagrams) of the isomorphism class of A (resp. f resp. D), i.e. it holds for all sets (resp. maps resp. diagrams) isomorphic to A (resp. f resp. D).

(2) We say that the *(categorical) property* $\mathcal{P}$ *holds* for an object C (resp. morphism f resp. diagram $D : I \longrightarrow \mathcal{C}$) of an arbitrary category $\mathcal{C}$ by requiring that the contravariant representable functor $\mathrm{Mor}_{\mathcal{C}}(\text{-}, C)$ (resp. the natural transformation $\mathrm{Mor}_{\mathcal{C}}(\text{-}, f)$ resp. the diagram functor $\mathrm{Mor}_{\mathcal{C}}(\text{-}, D) : \mathcal{C} \times I \longrightarrow$ **Set**) has as images sets $\mathrm{Mor}_{\mathcal{C}}(B, C)$ (resp. maps $\mathrm{Mor}_{\mathcal{C}}(B, f)$ resp. diagrams $\mathrm{Mor}_{\mathcal{C}}(B, D) : I \longrightarrow$ **Set**) with property $\mathcal{P}$ in **Set** for all objects B in $\mathcal{C}$.

(3) We check that the objects (resp. morphisms resp. diagrams) with (categorical) property $\mathcal{P}$ in **Set** are exactly the *sets* (resp. maps resp. diagrams) with (set-theoretic) property $\mathcal{P}$.

The last condition means in our example of a terminal object, that a set 1 is a singleton iff all $\mathrm{Mor}_{\mathbf{Set}}(B, 1)$ are singletons.
Very often the property $\mathcal{P}$ is given by a definition referring to specific elements. The construction of products is an example. Such a property might not hold for all objects of an isomorphism class. Then we will have to extend the definition to the whole isomorphism class (see for example section 2.7).

2.3.2. **Example.** The set of objects of a small category is 'essentially the same thing' as the set of global elements of the category (as an object of **Cat**). In other words, the *set of objects functor* is represented by the terminal object of **Cat**, which is the category with one object and one arrow.
The set of arrows of a small category is the object part of a functor that is represented by the category **2**, which is the graph **2** with the addition of two identity arrows.

2.4. **Isomorphisms.**

In 1.7.16 we have defined the notion of an isomorphism. This is another example of the principle of covariant extension of set-theoretic properties.
In general, the word 'isomorphic' is used in a mathematical context to mean indistinguishable in form.

2.4.1. **Definition.** A map $f : A \longrightarrow B$ in **Set** is called an *isomorphism* if it is bijective.

It is clear that $f : A \longrightarrow B$ is an isomorphism iff there is an inverse map $g : B \longrightarrow A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$.
In the category of arrows of **Set** two arrows $f : A \longrightarrow B$ and $f' : A' \longrightarrow B'$ are isomorphic if and only if there are maps $g : A \longrightarrow A'$, $h : B \longrightarrow B'$, $g' : A' \longrightarrow A$, and $h' : B' \longrightarrow B$ such that

$$\begin{aligned} h \circ f = f' \circ g, &\quad h' \circ f' = f \circ g', \\ g' \circ g = 1_A, &\quad g \circ g' = 1_{A'}, \\ h' \circ h = 1_B, &\quad h \circ h' = 1_{B'}. \end{aligned}$$

with

$$\begin{tikzcd} A \arrow[r, "g"] \arrow[d, "f"'] & A' \arrow[r, "g'"] \arrow[d, "f'"'] & A \arrow[r, "g"] \arrow[d, "f"] & A' \arrow[d, "f'"] \\ B \arrow[r, "h"'] & B' \arrow[r, "h'"'] & B \arrow[r, "h"'] & B' \end{tikzcd} \tag{14}$$

It is clear that f' is bijective if f is bijective, since $f' = f' \circ g \circ g' = h \circ f \circ g'$ with bijective maps h, f, and g'. Hence f is bijective if and only if all f' that are isomorphic to f are bijective maps. This is the extension principle 2.3.1 (1). The extension principle 2.3.1 (2) provides the following definition.

2.4.2. **Definition.** A morphism $f : A \longrightarrow B$ in an arbitrary category $\mathcal{C}$ is called *invertible* or an *isomorphism* if $\mathrm{Mor}_{\mathcal{C}}(C, f) : \mathrm{Mor}_{\mathcal{C}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C, B)$ is bijective for all objects C in $\mathcal{C}$.

This definition coincides with the original definition 1.7.16 of an isomorphism by the following Lemma:

2.4.3. **Lemma.** *Let $f : A \longrightarrow B$ be a morphism in $\mathcal{C}$. f is an isomorphism (as in definition 2.4.2) if and only if there exists a morphism $g : B \longrightarrow A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$.*

Proof. Let f be an isomorphism. Choose $C := B$. Then $\mathrm{Mor}_{\mathcal{C}}(B, f) : \mathrm{Mor}_{\mathcal{C}}(B, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(B, B)$ is a bijective map. Thus there is $g \in \mathrm{Mor}_{\mathcal{C}}(B, A)$ with $f \circ g = \mathrm{Mor}_{\mathcal{C}}(B, f)(g) = 1_B$. Furthermore $\mathrm{Mor}_{\mathcal{C}}(A, f) : \mathrm{Mor}_{\mathcal{C}}(A, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(A, B)$ is bijective hence $\mathrm{Mor}_{\mathcal{C}}(A, f)(g \circ f) = f \circ g \circ f = 1_B \circ f = f \circ 1_A = \mathrm{Mor}_{\mathcal{C}}(A, f)(1_A)$ implies $g \circ f = 1_A$.
Conversely assume that there is a morphism $g : B \longrightarrow A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$. Let C be an object in $\mathcal{C}$ and $h : C \longrightarrow A$. Then $\mathrm{Mor}_{\mathcal{C}}(C, g) \circ \mathrm{Mor}_{\mathcal{C}}(C, f)(h) = g \circ f \circ h = h$ hence $\mathrm{Mor}_{\mathcal{C}}(C, g) \circ \mathrm{Mor}_{\mathcal{C}}(C, f) = 1_{\mathrm{Mor}_{\mathcal{C}}(C,A)}$. Furthermore let $k : C \longrightarrow B$. Then $\mathrm{Mor}_{\mathcal{C}}(C, f) \circ \mathrm{Mor}_{\mathcal{C}}(C, g)(k) = f \circ g \circ k = k$ hence $\mathrm{Mor}_{\mathcal{C}}(C, f) \circ \mathrm{Mor}_{\mathcal{C}}(C, g) = 1_{\mathrm{Mor}_{\mathcal{C}}(C,B)}$. Thus $\mathrm{Mor}_{\mathcal{C}}(B, f) : \mathrm{Mor}_{\mathcal{C}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C, B)$ is a bijective map. □

We now have to show that this definition satisfies extension principle 2.3.1 (3).
Let $f : A \longrightarrow B$ be a (categorical) isomorphism in **Set**. By Lemma 2.4.3 f has an inverse map $g : B \longrightarrow A$, hence it is bijective.

2.4.4. **Proposition.** *If $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are isomorphisms in a category with inverses $f^{-1} : B \longrightarrow A$ and $g^{-1} : C \longrightarrow B$, then $g \circ f$ is an isomorphism with inverse $f^{-1} \circ g^{-1}$.*

The proof is omitted. This proposition is sometimes called the 'Shoe-Sock Theorem': to undo the act of putting on your socks, then your shoes, you have to take off your shoes, then your socks.

2.4.5. **Definition.** Let $\mathcal{C}$ be a category and suppose that A and B are two objects of $\mathcal{C}$. We say that A is *isomorphic to* B, written $A \cong B$, if there is an isomorphism $f : A \longrightarrow B$.

2.4.6. **Definition.** An arrow $f : A \longrightarrow A$ in a category (with the source and target the same) is called an *endomorphism*. If it is invertible, it is called an *automorphism*.

2.4.7. **Examples.** Identity arrows in a category are isomorphisms (hence an automorphisms).
In the category determined by a partially ordered set, the *only* isomorphisms are the identity arrows.
If, in the category determined by a monoid, every arrow is an isomorphism then the monoid is a group. Because of this, a category in which every arrow is an isomorphism is called a *groupoid*.
In the category of semigroups and semigroup homomorphisms, and likewise in the category of monoids and monoid homomorphisms, the isomorphisms are exactly the bijective homomorphisms. On the other hand, in the category of posets and monotone maps, there are bijective homomorphisms that have no inverse.

2.4.8. **Definition.** A *property* that objects of a category may have *is preserved by isomorphisms* if for any object A with the property, any object isomorphic to A must also have the property.

From the categorist's point of view there is no reason to distinguish between two isomorphic objects in a category, since the interesting fact about a mathematical object is the way it relates to other mathematical objects and two isomorphic objects relate to other objects in the same way. For this reason, the concept of wide subcategory (definition 1.4.10) is not in the spirit of category theory. What really should matter is whether the subcategory contains an isomorphic copy of every object in the big category. This motivates the following definition.

2.4.9. **Definition.** A subcategory $\mathcal{D}$ of a category $\mathcal{C}$ is said to be a *representative subcategory* if every object of $\mathcal{C}$ is isomorphic to some object of $\mathcal{D}$.

2.4.10. **Example.** Let $\mathcal{D}$ be the category whose objects are the empty set and all sets of the form $\{1, 2, \ldots, n\}$ for some positive integer n and whose arrows are all the functions between them. Then $\mathcal{D}$ is a representative subcategory of **Fin** (Definition 2.1.12), since there is a bijection from any nonempty finite set to some set of the form $\{1, 2, \ldots, n\}$. Note that $\mathcal{D}$ is also full in **Fin** and that $\mathcal{D}$ is a small category.

2.5. **Monomorphisms and epimorphisms.**
Recall that a function $f : A \longrightarrow B$ in **Set** is *injective* if for any $x, y \in A$, if $x \neq y$, then $f(x) \neq f(y)$.

2.5.1. **Definition.** A map $f : A \longrightarrow B$ in **Set** is called a *(set-theoretic) monomorphism* if it is injective.

Let the two arrows $f : A \longrightarrow B$ and $f' : A' \longrightarrow B'$ be isomorphic with bijective $g : A \longrightarrow A'$, $h : B \longrightarrow B'$, $g' : A' \longrightarrow A$, and $h' : B' \longrightarrow B$ as in diagram (14). Then f' is injective if f is injective. In fact assume $f'(x) = f'(y)$. Then $fg'(x) = h'f'(x) = h'f'(y) = fg'(y)$ implies $g'(x) = g'(y)$ and hence $x = y$ since g' is bijective and f is injective.
So the notion of monomorphism satisfies the extension principle 2.3.1 (1). The extension principle 2.3.1 (2) provides the following definition.

2.5.2. **Definition.** (1) A morphism $f : A \longrightarrow B$ in an arbitrary category $\mathcal{C}$ is called a *(categorical) monomorphism* if $\mathrm{Mor}_{\mathcal{C}}(C, f) : \mathrm{Mor}_{\mathcal{C}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C, B)$ is injective for all objects C in $\mathcal{C}$.
(2) A morphism $f : A \longrightarrow B$ in an arbitrary category $\mathcal{C}$ is called an *epimorphism* if $\mathrm{Mor}_{\mathcal{C}^{op}}(C, f) : \mathrm{Mor}_{\mathcal{C}^{op}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}^{op}}(C, B)$ is injective for all objects C in $\mathcal{C}$.

2.5.3. **Proposition.** (1) *A morphism $f : A \longrightarrow B$ is a monomorphism if and only if it satisfies the* left cancelation property*:*
- $\forall C \in \mathcal{C},\ \forall g, h : C \longrightarrow A$*:*

$$f \circ g = f \circ h \Longrightarrow g = h.$$

(2) *A morphism $f : A \longrightarrow B$ is an epimorphism if and only if it satisfies the* right cancelation property*:*
- $\forall C \in \mathcal{C},\ \forall g, h : B \longrightarrow C$*:*

$$g \circ f = h \circ f \Longrightarrow g = h.$$

Proof. The left cancelation property says that

$$\mathrm{Mor}_{\mathcal{C}}(C, f) : \mathrm{Mor}_{\mathcal{C}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C, B)$$

is injective for all C in $\mathcal{C}$.
The right cancelation properties follows by duality. □

03.06.04

We now have to show that the definition of a monomorphism satisfies the extension principle 2.3.1 (3).

2.5.4. **Proposition.** *A map in* **Set** *is a (categorical) monomorphism if and only if it is injective.*

Proof. Let $f : A \longrightarrow B$ be a monomorphism in **Set**. Then $\mathrm{Mor}_{\mathbf{Set}}(C, f) : \mathrm{Mor}_{\mathbf{Set}}(C, A) \longrightarrow \mathrm{Mor}_{\mathbf{Set}}(C, B)$ is injective for all sets C. Let $x, y \in A$ with $f(x) = f(y)$. Let C be a terminal object $\{*\}$ in **Set**. Then there are maps $g, h : \{*\} \longrightarrow A$ with $g(*) = x$ and $h(*) = y$. The composites satisfy $f \circ g(*) = f(x) = f(y) = f \circ h(*)$ hence $f \circ g = f \circ h$ and $g = h$. So we get $x = g(*) = h(*) = y$ which shows that f is injective.
Conversely assume that f is injective. Let $g, h : C \longrightarrow A$ maps with $f \circ g = f \circ h$. Then $f(g(x)) = f(h(x))$ for all $x \in C$, hence $g(x) = h(x)$ and $g = h$. Thus f is left cancelable and hence a monomorphism. □

Using the notation of variable elements of 2.1.3, f is a monomorphism if for any variable elements $x, y : T \longrightarrow A$, if $x \neq y$, then $f(x) \neq f(y)$, i.e. $f : A(T) \longrightarrow B(T)$ is injective for all parameters T.

2.5.5. **Remark.** Now that we have completed all three steps of the extension principle we may abbreviate the term 'categorical monomorphism' just to 'monomorphism'.
An arrow, that is a monomorphism in a category, is a monomorphism in any subcategory it happens to be in. However, an arrow can be a monomorphism in a subcategory of a category $\mathcal{C}$ without being a monomorphism in $\mathcal{C}$.

2.5.6. **Examples.** In most familiar categories of sets with structure and functions that preserve structure, the monomorphisms are exactly the injective functions. In particular, the monomorphisms in **Mon** are the injective homomorphisms (see 2.5.7). This is more evidence that definition 2.5.2 is the correct categorical definition generalizing the set-theoretic concept of injectivity.
In the category determined by a poset, every arrow is a monomorphism. A monomorphism of the category determined by a monoid is generally called left cancelable.
An isomorphism in any category is a monomorphism. For suppose f is an isomorphism and assume $f \circ x = f \circ y$. Then $x = f^{-1} \circ f \circ x = f^{-1} \circ f \circ y = y$.

2.5.7. **Proposition.** *A monomorphism in the category of monoids is an injective homomorphism, and conversely.*

Proof. Let $f : M \longrightarrow M'$ be a monoid homomorphism. Suppose it is injective. Let $g, h : V \longrightarrow M$ be homomorphisms for which $f \circ g = f \circ h$. For any $v \in V$, $f(g(v)) = f(h(v))$, so $g(v) = h(v)$ since f is injective. Hence $g = h$. It follows that f is a monomorphism.
Essentially the same proof works in other categories of structures and structure-preserving maps – if the map is injective it is a monomorphism for the same reason as in **Set** (see Exercise **??**).
However, the converse definitely does not work that way. The proof for **Set** in Theorem 2.5.4 above uses distinct global elements x and y, but a monoid need not have distinct global elements. For example, let N denote the monoid of nonnegative integers with addition as operation. Then the only global element of N on addition is 0. So we have to work harder to get a proof.
Suppose f is a monomorphism. Let $x, y \in M$ be distinct elements. Let $p_x : N \longrightarrow M$ take k to x^k and similarly define p_y; p_x and p_y are homomorphisms since for all x, $x^{k+n} = x^k x^n$. Since $x \neq y$, p_x and p_y are distinct homomorphisms.
If $f(x) = f(y)$ then for all positive integers k, $f(p_x(k)) = f(x^k) = f(x)^k = f(y)^k = f(y^k) = f(p_y(k))$ so that $f \circ p_x = f \circ p_y$ which would mean that f is not a monomorphism. Thus we must have $f(x) \neq f(y)$ so that f is injective. □

The trick in the preceding proof was to find an object (N here) that allows one to distinguish elements of the arbitrary monoid M. In **Set**, the corresponding object was the terminal object, but that does not work for **Mon**: each monoid has exactly one global element because a map from the one-element monoid must have the identity element as value.
We now state two propositions that give some elementary properties of monomorphisms.

2.5.8. **Proposition.** *Suppose $f : A \longrightarrow B$ and $g : B \longrightarrow C$ in a category $\mathcal{C}$. Then*

(1) *If f and g are monomorphisms, so is $g \circ f$.*
(2) *If $g \circ f$ is a monomorphism, so is f.*

Proof. (1) The extension principle and the definition 2.5.2 of monomorphisms give a short proof. Using $\mathrm{Mor}_{\mathcal{C}}(T, g) \circ \mathrm{Mor}_{\mathcal{C}}(T, f) = \mathrm{Mor}_{\mathcal{C}}(T, g \circ f)$ we get:

$$\begin{array}{l} f, g \text{ are monomorphisms} \iff \\ \mathrm{Mor}_{\mathcal{C}}(T, f), \mathrm{Mor}_{\mathcal{C}}(T, g) \text{ are injective for all objects } T \text{ in } \mathcal{C} \Longrightarrow \\ \mathrm{Mor}_{\mathcal{C}}(T, g \circ f) \text{ is injective for all objects } T \text{ in } \mathcal{C} \iff \\ g \circ f \text{ is a monomorphism.} \end{array}$$

(2) With the same arguments we get:

$$\begin{array}{l} g \circ f \text{ is a monomorphism} \iff \\ \mathrm{Mor}_{\mathcal{C}}(T, g \circ f) \text{ is injective for all objects } T \text{ in } \mathcal{C} \Longrightarrow \\ \mathrm{Mor}_{\mathcal{C}}(T, f) \text{ is injective for all objects } T \text{ in } \mathcal{C} \iff \\ f \text{ is a monomorphism.} \end{array}$$

□

2.5.9. **Proposition.** *Let $m : C \longrightarrow 0$ be a monomorphism into an initial object. Then m is an isomorphism.*

Proof. Let $i : 0 \longrightarrow C$ be the unique arrow given by definition of initial object. Then $m \circ i$ and 1_0 are both arrows from 0 to 0 and so must be the same. It remains to show that $i \circ m = 1_C$. This follows from the fact that $m \circ i \circ m = m \circ 1_C$ and that m is a monomorphism. □

2.5.10. **Remark** (**Properties of Cat).** We note some properties without proof. The initial category has no objects and, therefore, no arrows. The terminal category has just one object and the identity arrow of that object. To any category $\mathcal{C}$ there is just one functor that takes every object to that single object and every arrow to that one arrow. A functor is a monomorphism in **Cat** if and only if it is injective on both objects and arrows. The corresponding statement for epimorphisms is not true.

2.5.11. **Example.** Not all monomorphisms, however, coincide with injective maps between objects in $\mathcal{C}$, even if there is an underlying (faithful) functor from $\mathcal{C}$ to **Set**. For an example consider the category of divisible abelian groups.
An abelian group G is called *divisible* if $nG = G$ for each natural number $n \geq 1$, i.e. for each $g \in G$ and n there is an element $g' \in G$ with $ng' = g$.
Consider the residue class homomorphism $\nu : \mathbb{Q} \longrightarrow \mathbb{Q}/\mathbb{Z}$ from rational numbers to rational numbers modulo integers. This is a monomorphism in the category of divisible abelian groups (with group homomorphisms as morphisms). Let $f, g : A \longrightarrow \mathbb{Q}$ be homomorphisms with $f \neq g$. Then there is $a \in A$ with $0 \neq f(a) - g(a) = r/s \in \mathbb{Q}$. Let $b \in A$ with $rb = a$. Then $r(f(b) - g(b)) = f(a) - g(a) = r/s$, hence $f(a) - g(a) = 1/s$ and $\nu f(b) \neq \nu g(b)$. This implies $\nu f \neq \nu g$ hence ν is a monomorphism.

For the dual notion of epimorphisms we have similar properties.

2.5.12. **Proposition.** *A set function is an epimorphism in* **Set** *if and only if it is surjective.*

Proof. Suppose $f : S \longrightarrow T$ is surjective, and $g, h : T \longrightarrow X$ are two functions. If $g \neq h$, then there is some particular element $t \in T$ for which $g(t) \neq h(t)$. Since f is surjective, there is an element $s \in S$ for which $f(s) = t$. Then $g(f(s)) \neq h(f(s))$, so that $g \circ f \neq h \circ f$.
Conversely, suppose f is not surjective. Then there is some $t \in T$ for which there is no $s \in S$ such that $f(s) = t$. Now define two functions $g : T \longrightarrow \{0, 1\}$ and $h : T \longrightarrow \{0, 1\}$ as follows:

(i) $g(x) = h(x) = 0$ for all x in T except t.
(ii) $g(t) = 0$.
(iii) $h(t) = 1$.

Then $g \neq h$ but $g \circ f = h \circ f$, so f is not an epimorphism. □

04.06.04

2.5.13. **Proposition.** *Let $f : A \longrightarrow B$ and $g : B \longrightarrow C$. Then*

(1) *If f and g are epimorphisms, so is $g \circ f$.*
(2) *If $g \circ f$ is an epimorphism, so is g.*

Proof. This is the dual of Proposition 2.5.8. □

2.5.14. **Remark.** In **Set** an arrow that is both monic and epic is bijective, and hence an isomorphism. In general, this need not happen. One example is the inclusion of $\mathbb{Z}$ in $\mathbb{Q}$ in Ri described in **??** (an inverse would also have to be an inverse in **Set**, but there isn't one since the inclusion is not bijective).
An easier example is the arrow in

$$\begin{tikzcd} C \arrow[loop above] \arrow[r] & D \arrow[loop above] \end{tikzcd}$$

from C to D. It is both monic and epic (vacuously) but there is no arrow from D to C so it is not an isomorphism because there is no arrow in the category that could be its inverse.

If we try to generalize the notion of a surjective map according to the extension principle we obtain the notion of a split epimorphism. We first introduce this notion.
An arrow $f : A \longrightarrow B$ in a category is an isomorphism if it has an inverse $g : B \longrightarrow A$ which must satisfy both the equations $g \circ f = 1_A$ and $f \circ g = 1_B$. If it only satisfies the second equation, $f \circ g = 1_B$, then f is a *left inverse* of g and (as you might expect) g is a *right inverse* of f.

2.5.15. **Definition.** Suppose f has a right inverse g. Then f is called a *split epimorphism* (f is *split by* g) and g is called a *split monomorphism*.

2.5.16. **Lemma.** *Every surjection in* **Set** *is a split epimorphism.*

Proof. If $f : A \longrightarrow B$, then choose, for each $b \in B$, some element $a \in A$ such that $f(a) = b$. The existence of such an a is guaranteed by surjectivity. Define $g(b)$ to be a. Then $f(g(b)) = f(a) = b$ for any $b \in B$, so $f \circ g = 1_B$. □

The so-called axiom of choice is exactly what is required to make all those generally infinitely many choices.
And in fact, one possible formulation of the axiom of choice is that every epimorphism splits.

2.5.17. **Proposition.** *A morphism $f : A \longrightarrow B$ in $\mathcal{C}$ is a split epimorphism if and only if*

$$\mathrm{Mor}_{\mathcal{C}}(C, f) : \mathrm{Mor}_{\mathcal{C}}(C, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C, B)$$

is surjective for all objects C in $\mathcal{C}$.

Proof. Let f have the splitting $g : B \longrightarrow A$. Let $h \in \mathrm{Mor}_{\mathcal{C}}(C, B)$. Then $g \circ h \in \mathrm{Mor}_{\mathcal{C}}(C, A)$ and $f \circ (g \circ h) = (f \circ g) \circ h = h$ so that $\mathrm{Mor}_{\mathcal{C}}(C, f)$ is surjective.
Conversely assume that all $\mathrm{Mor}_{\mathcal{C}}(C, f)$ are surjective. Choose $C = B$ and let $g \in \mathrm{Mor}_{\mathcal{C}}(B, A)$ with $\mathrm{Mor}_{\mathcal{C}}(B, f)(g) = 1_B \in \mathrm{Mor}_{\mathcal{C}}(B, B)$. Then $f \circ g = 1_B$ so that f is a split epimorphism. $\square$

So we have checked all items for the extension principle and a *split epimorphism* is the covariant extension of the notion of surjective map in **Set**.
Epimorphisms in other categories may not be split. The function that includes the monoid of non-negative integers $(\mathbb{N}, +)$ on addition in the monoid of all the integers $(\mathbb{Z}, +)$ on addition certainly does not have a right inverse in the category of monoids, since it does not have a right inverse in the category of sets. There are plenty of examples of epimorphisms of monoids which are surjective which have no right inverse in the category of monoids, although of course they do in the category of sets. One such epimorphism of monoids is the function that takes the integers mod 4 on addition to the integers mod 2 on addition, with 0 and 2 going to 0 and 1 and 3 going to 1.
Unlike epis, which always split in the category of sets, monics in **Set** do not always split. Every arrow out of the empty set is monic and, save for the identity of $\emptyset$ to itself, is not split. On the other hand, every monic with nonempty source does split. We leave the details to the reader.

2.6. **Subobjects.**
The concept of subobject is intended to generalize the concept of subset of a set, submonoid of a monoid, subcategory of a category, and so on. This idea cannot be translated exactly into categorical terms, since the usual concept of subset violates the strict typing rules of category theory: to go from a subset to a set requires a change of type, so there is no feasible way to say that the same element x is in both a set and a subset of the set. Because of this, any categorical definition of subobject will not give exactly the concept of subset when applied to the category of sets. However, the usual definition of subobject produces, in **Set**, a concept that is naturally equivalent to the concept of subset. The definition, when applied to sets, defines subset in terms of the inclusion function.
We need a preliminary idea. Let $\mathcal{C}/C$ be a comma category. We call an object $f : A \longrightarrow C$ *monic* if f is a monomorphism in $\mathcal{C}$.

2.6.1. **Proposition.** *If $f : A \longrightarrow C$ and $f' : A' \longrightarrow C$ are monic, then there is at most one morphism $g : (A, f) \longrightarrow (A', f')$ in $\mathcal{C}/C$.*

Proof. Given $g, g' : (A, f) \longrightarrow (A', f')$. Then $f' \circ g = f = f' \circ g'$ in $\mathcal{C}$. Since f' is a monomorphism in $\mathcal{C}$ we get $g = g'$. $\square$

2.6.2. **Corollary.** *If there are morphisms $g : (A, f) \longrightarrow (A', f')$ and $h : (A', f') \longrightarrow (A, f)$, then g is an isomorphism and $h = g^{-1}$.*

Proof. By the proposition we have $h \circ g = 1_A : (A, f) \longrightarrow (A, f)$ and $g \circ h = 1_{A'} : (A', f') \longrightarrow (A', f')$ so that g and h are mutually inverse morphisms. $\square$

2.6.3. **Remark.** Isomorphisms between objects (A, f) in $\mathcal{C}$ define an equivalence relation. The equivalence classes are partially ordered by $\overline{(A, f)} \leq \overline{(A', f')}$ iff there is a morphism $g : (A, f) \longrightarrow (A', f')$.

2.6.4. **Definition.** In a category $\mathcal{C}$, a *subobject* of an object C is an equivalence or isomorphism class of monics in $\mathcal{C}/C$. The subobject is a *proper subobject* if it does not contain $1_C : C \longrightarrow C$.

Observe that it follows immediately from Proposition 2.5.9 that an initial object in a category has no proper subobjects.

2.6.5. **Remark.** Let $\alpha : F \longrightarrow G$ be a natural transformation between models of $\mathcal{G}$ in $\mathcal{D}$. Suppose each component of α is a monomorphism in $\mathcal{D}$. Then it is easy to prove that α is a monomorphism in $\mathbf{Mod}(\mathcal{G}, \mathcal{D})$. The converse is not true.

2.7. **Products.**

Another application of the extension principle relates to products. We first define the product of two sets.

2.7.1. **Definition.** If A and B are sets, the *Cartesian product* $A \times B$ is the set of all ordered pairs with first coordinate in A and second coordinate in B; in other words,

$$A \times B = \{(a, b) | a \in A \text{ and } b \in B\}.$$

The coordinates define functions $p_A = \mathrm{proj}_1 : A \times B \longrightarrow A$ and $\mathrm{proj}_2 = \mathrm{proj}_2 : A \times B \longrightarrow B$ called the *coordinate functions*, or the *projections*.

Now this definition does not satisfy the first property for the extension principle, a set X isomorphic to $A \times B$ will in general not consist of pairs of elements. So the sets $\{0, 1, 2, 3, 4, 5\}$ and $\{0, 1, \} \times \{0, 1, 2\}$ are clearly isomorphic, because both have 6 elements, but the first set does not consist of pairs. So here we have a set theoretic definition that refers to specific properties of the elements.
All the more it is interesting to find out that there is an example for the extension principle hidden behind their concept.

2.7.2. **Proposition.** *Let A, B be sets. The Cartesian product $A \times B$ together with the projections* $\mathrm{proj}_1 : A \times B \longrightarrow A$ *and* $\mathrm{proj}_2 : A \times B \longrightarrow B$ *satisfies the following universal property:*

(*) *for every set T and every pair of maps $f : T \longrightarrow A$, $g : T \longrightarrow B$ there is a unique map $(f, g) : T \longrightarrow A \times B$ such that the diagram*

$$\begin{tikzcd} & T \arrow[dl, "f"'] \arrow[d, "{(f,g)}"'] \arrow[dr, "g"] & \\ A & A \times B \arrow[l, "\mathrm{proj}_1"] \arrow[r, "\mathrm{proj}_2"'] & B \end{tikzcd}$$

commutes, i.e. $\mathrm{proj}_1 \circ (f, g) = f$ *and* $\mathrm{proj}_2 \circ (f, g) = g$.

Proof. We first show the uniqueness of the map $(f, g) : T \longrightarrow A \times B$: Given T, f, and g. If $h, k : T \longrightarrow A \times B$ are maps such that $\mathrm{proj}_1 \circ h = f = \mathrm{proj}_1 \circ k$ and $\mathrm{proj}_2 \circ h = g = \mathrm{proj}_2 \circ k$ then for any $t \in T$ we have $h(t) =: (a, b) \in A \times B$ and $k(t) =: (a', b') \in A \times B$. Then

we get $a = \mathrm{proj}_1(a,b) = (\mathrm{proj}_1 \circ h)(t) = f(t) = (\mathrm{proj}_1 \circ k)(t) = \mathrm{proj}_1(a',b') = a'$ and $b = \mathrm{proj}_2(a,b) = (\mathrm{proj}_2 \circ h)(t) = g(t) = (\mathrm{proj}_2 \circ k)(t) = \mathrm{proj}_2(a',b') = b'$, hence $h(t) = (a,b) = (a',b') = k(t)$ and $h = k$. This shows the uniqueness of the map h.
Existence: From the uniqueness proof we obtain a reasonable definition of (f,g). For $t \in T$ we define

$$(f,g)(t) := (f(t), g(t)).$$

This is a map from T to $A \times B$. Since it satisfies $(\mathrm{proj}_1 \circ (f,g))(t) = f(t)$ and $(\mathrm{proj}_2 \circ (f,g))(t) = g(t)$ we get $\mathrm{proj}_1 \circ (f,g) = f$ and $\mathrm{proj}_2 \circ (f,g) = g$. □

Obviously any triple $(P, p'_A : P \longrightarrow A, p'_B : P \longrightarrow B)$ isomorphic to $(A \times B, \mathrm{proj}_1, \mathrm{proj}_2)$ satisfies the same universal property. In fact let $h : P \longrightarrow A \times B$ be an isomorphism such that

$$\begin{tikzcd}
 & P \arrow[dl, "p'_A"'] \arrow[dd, "h"] \arrow[dr, "p'_B"] & \\
A & & B \\
 & A \times B \arrow[ul, "\mathrm{proj}_1"] \arrow[ur, "\mathrm{proj}_2"'] &
\end{tikzcd}$$

commutes. Let T be a set and let $f : T \longrightarrow A$, $g : T \longrightarrow B$ be maps. Then the map $h^{-1} \circ (f,g) : T \longrightarrow P$ satisfies $p'_A \circ (h^{-1} \circ (f,g)) = f$ and $p'_B \circ (h^{-1} \circ (f,g)) = g$. It is an easy exercise to show that this map is unique with this property. So the universal property given in the theorem satisfies the first property of the extension principle.
In arbitrary categories we cannot construct the Cartesian product of two objects since there are no elements and we thus cannot construct pairs of elements. So when we use the notation $A \times B$ in an arbitrary category this cannot mean a set of pairs of elements. It will stand for an arbitrary object with additional properties described by morphisms.
The extension principle 2.3.1 (2) provides the following definition.

2.7.3. **Definition.** Given objects A and B in a category $\mathcal{C}$. An object $A \times B$ together with morphisms $p_A : A \times B \longrightarrow A$ and $p_B : A \times B \longrightarrow B$ is called a (not the!) *(categorical) product* of A and B, and the morphisms p_A, p_B are called *projections*, if for all objects T in $\mathcal{C}$ there is an isomorphism of sets

$$\mathrm{Mor}_{\mathcal{C}}(T, A \times B) \cong \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$$

(where $\mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$is the Cartesian product) such that the diagram

$$\begin{tikzcd}
 & \mathrm{Mor}_{\mathcal{C}}(T, A \times B) \arrow[dl, "\mathrm{Mor}_{\mathcal{C}}(T, p_A)"'] \arrow[dd, "h"] \arrow[dr, "\mathrm{Mor}_{\mathcal{C}}(T, p_B)"] & \\
\mathrm{Mor}_{\mathcal{C}}(T, A) & & \mathrm{Mor}_{\mathcal{C}}(T, B) \\
 & \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B) \arrow[ul, "p_{\mathrm{Mor}_{\mathcal{C}}(T,A)}"] \arrow[ur, "p_{\mathrm{Mor}_{\mathcal{C}}(T,B)}"'] &
\end{tikzcd}$$

11.06.04 commutes.

2.7.4. **Proposition (Characterization of products by the universal mapping property).** *Given objects A and B in a category $\mathcal{C}$. An object $A \times B$ together with morphisms $p_A : A \times B \longrightarrow A$ and $p_B : A \times B \longrightarrow B$ is a (categorical) product,*

(**) *if and only if for each object $T \in \mathcal{C}$ and each pair of morphisms $f : T \longrightarrow A$, $g : T \longrightarrow B$ there is a unique morphism $(f, g) : T \longrightarrow A \times B$ such that the diagram*

$$\begin{tikzcd} & T \arrow[dl, "f"'] \arrow[d, "(f,g)"'] \arrow[dr, "g"] & \\ A & A \times B \arrow[l, "p_A"] \arrow[r, "p_B"'] & B \end{tikzcd}$$

commutes, i.e. $p_A \circ (f, g) = f$ *and* $p_B \circ (f, g) = g$.

Proof. $\Longleftarrow$: Let $(A \times B, p_A, p_B)$ be a product of A and B. Let T be an object in $\mathcal{C}$ and let $u \in \mathrm{Mor}_{\mathcal{C}}(T, A \times B)$. Define $f := p_A \circ u : T \longrightarrow A$ und $g := p_B \circ u : T \longrightarrow B$. Then $(f, g) \in \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$. Now we define $h(u) := (f, g) \in \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$ and get $\mathrm{Mor}_{\mathcal{C}}(T, p_A)(u) = p_A \circ u = f = p_{\mathrm{Mor}_{\mathcal{C}}(T,A)} \circ h(u)$ and $\mathrm{Mor}_{\mathcal{C}}(T, p_B)(u) = p_B \circ u = g = p_{\mathrm{Mor}_{\mathcal{C}}(T,B)} \circ h(u)$.
It remains to show that h is bijective. We construct the inverse map as

$$k : \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T, A \times B) \qquad k((f, g)) := u$$

where $(f, g) \in \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$. By the universal property of the product there is a unique $u : T \longrightarrow A \times B$ such that $p_A \circ u = f$ and $p_B \circ u = g$. We use this u for the definition of the map h.
Let $(f, g) \in \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$. Then $(h \circ k)((f, g)) = h(u) = (p_A \circ u, p_B \circ u) = (f, g)$ hence $(i \circ u) = 1_{\mathrm{Mor}_{\mathcal{C}}(T,A) \times \mathrm{Mor}_{\mathcal{C}}(T,B)}$.
Let $u \in \mathrm{Mor}_{\mathcal{C}}(T, A \times B)$. Then $(k \circ h)(u) = k((f, g)) = k((p_A \circ u, p_B \circ u)) = u'$ where $u' : T \longrightarrow A \times B$ is the unique morphism such that $p_A \circ u' = p_A \circ f$ and $p_B \circ u' = p_B \circ u$ hence $u = u'$ and $(k \circ h) = 1_{\mathrm{Mor}_{\mathcal{C}}(T,A \times B)}$.
$\Longrightarrow$: Given morphisms $f : T \longrightarrow A$ and $g : T \longrightarrow B$. Then (f, g) is an element of $\mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$.
So there is a unique morphism $u := h^{-1}((f, g))$ satisfying $p_{\mathrm{Mor}_{\mathcal{C}}(T,A)} \circ h(u) = f$ and $p_{\mathrm{Mor}_{\mathcal{C}}(T,B)} \circ h(u) = g$. Equivalently $\mathrm{Mor}_{\mathcal{C}}(T, p_A)(u) = f = p_A \circ u$ and $\mathrm{Mor}_{\mathcal{C}}(T, p_B)(u) = g = p_B \circ u$ with a unique u. Thus $(A \times B, p_A, p_B)$ is a product of A and B. □

2.7.5. **Proposition.** *The family of isomorphisms*

$$h_T : \mathrm{Mor}_{\mathcal{C}}(T, A \times B) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B)$$

such that

$$\begin{tikzcd} & \mathrm{Mor}_{\mathcal{C}}(T, A \times B) \arrow[dl, "\mathrm{Mor}_{\mathcal{C}}(T, p_A)"'] \arrow[dd, "h_T"] \arrow[dr, "\mathrm{Mor}_{\mathcal{C}}(T, p_B)"] & \\ \mathrm{Mor}_{\mathcal{C}}(T, A) & & \mathrm{Mor}_{\mathcal{C}}(T, B) \\ & \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B) \arrow[ul, "p_{\mathrm{Mor}_{\mathcal{C}}(T,A)}"] \arrow[ur, "p_{\mathrm{Mor}_{\mathcal{C}}(T,B)}"'] & \end{tikzcd}$$

commute for all objects T in $\mathcal{C}$, is a natural isomorphism in T, A, and B.

Proof. We show that the inverses

$$k_T : \mathrm{Mor}_{\mathcal{C}}(T, A) \times \mathrm{Mor}_{\mathcal{C}}(T, B) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T, A \times B)$$

as constructed in Proposition 2.7.4 form a natural transformation. So for $t : T' \longrightarrow T$ we have to show that the diagram

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{C}}(T,A) \times \mathrm{Mor}_{\mathcal{C}}(T,B) \arrow[r, "k_T"] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(t,A) \times \mathrm{Mor}_{\mathcal{C}}(t,B)"'] & \mathrm{Mor}_{\mathcal{C}}(T,A \times B) \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(t,A \times B)"] \\
\mathrm{Mor}_{\mathcal{C}}(T',A) \times \mathrm{Mor}_{\mathcal{C}}(T',B) \arrow[r, "k_{T'}"] & \mathrm{Mor}_{\mathcal{C}}(T',A \times B)
\end{tikzcd}$$

commutes. Let $(f,g) \in \mathrm{Mor}_{\mathcal{C}}(T,A) \times \mathrm{Mor}_{\mathcal{C}}(T,B)$. Then $[f,g] := k_T((f,g))$ is the unique morphism satisfying $p_A \circ [f,g] = f$ and $p_B \circ [f,g] = g$. The two sides of the square give two morphisms $[f \circ t, g \circ t] = (k_T \circ \mathrm{Mor}_{\mathcal{C}}(t,A) \times \mathrm{Mor}_{\mathcal{C}}(t,B))((f,g))$ and $[f,g] \circ t = (\mathrm{Mor}_{\mathcal{C}}(f, A \times B) \circ k_{T'})((f,g))$. We have to show that these are equal. Observe that a morphism $u \in \mathrm{Mor}_{\mathcal{C}}(T', A \times B)$ is uniquely determined by the morphisms $p_A \circ u$ and $p_B \circ u$. We get from $p_A \circ [f,g] \circ t = f \circ t = p_A \circ [f \circ t, g \circ t]$ and $p_B \circ [f,g] \circ t = g \circ t = p_B \circ [f \circ t, g \circ t]$ the result $[f,g] \circ t = [f \circ t, g \circ t]$.
Now let $a : A \longrightarrow A'$ and $b : B \longrightarrow B'$ be given. Then the diagram

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{C}}(T,A) \times \mathrm{Mor}_{\mathcal{C}}(T,B) \arrow[r, "k"] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T,a) \times \mathrm{Mor}_{\mathcal{C}}(T,b)"'] & \mathrm{Mor}_{\mathcal{C}}(T,A \times B) \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T,a \times b)"] \\
\mathrm{Mor}_{\mathcal{C}}(T,A') \times \mathrm{Mor}_{\mathcal{C}}(T,B') \arrow[r, "k"] & \mathrm{Mor}_{\mathcal{C}}(T,A' \times B')
\end{tikzcd}$$

commutes since $p_{A'} \circ (\mathrm{Mor}_{\mathcal{C}}(T, a \times b) \circ k)((f,g)) = p_{A'} \circ (a \times b) \circ [f,g]) = a \circ p_A \circ [f,g] = a \circ f = p'_A([a \circ f, b \circ f]) = p'_A \circ (k \circ (\mathrm{Mor}_{\mathcal{C}}(T,a) \times \mathrm{Mor}_{\mathcal{C}}(T,b)))((f,g))$ and $p_{B'} \circ (\mathrm{Mor}_{\mathcal{C}}(T, a \times b) \circ k)((f,g)) = p_{B'} \circ (a \times b) \circ [f,g]) = b \circ p_B \circ [f,g] = b \circ f = p'_B([a \circ f, b \circ f]) = p'_B \circ (k \circ (\mathrm{Mor}_{\mathcal{C}}(T,a) \times \mathrm{Mor}_{\mathcal{C}}(T,b)))((f,g))$ implies $\mathrm{Mor}_{\mathcal{C}}(T, a \times b) \circ k = k \circ (\mathrm{Mor}_{\mathcal{C}}(T,a) \times \mathrm{Mor}_{\mathcal{C}}(T,b))$. $\square$

Before we show that the definition of a product satisfies the extension principle 2.3.1 (3) we show a general uniqueness theorem.

2.7.6. **Theorem.** *Let A and B be objects of a category $\mathcal{C}$. If (P, p_A, p_B) and (P', p'_A, p'_B) are products of A and B in $\mathcal{C}$, then there is a unique isomorphism $k : P' \longrightarrow P$ such that $p_A \circ k = p'_A$ and $p_B \circ k = p'_B$.*

Proof. Since (P, p_A, p_B) satisfies the universal property (**) there is a unique morphism $k : P' \longrightarrow P$ such that $p_A \circ k = p'_A$ and $p_B \circ k = p'_B$. Symmetrically there is a unique morphism $h : P' \longrightarrow P$ such that $p'_A \circ h = p_A$ and $p'_B \circ h = p_B$. Consider the following commutative diagram:

$$\begin{tikzcd}
& P \arrow[ddl, "p_A"'] \arrow[ddr, "p_B"] \arrow[d, "h"'] \arrow[ddd, bend left, "1_P"] & \\
& P' \arrow[dl, "p'_A"'] \arrow[dr, "p'_B"] \arrow[dd, "k"] \arrow[ddd, bend right, "1_{P'}"'] & \\
A & & B \\
& P \arrow[ul, "p_A"] \arrow[ur, "p_B"'] \arrow[d, "h"] & \\
& P' \arrow[uul, "p'_A"] \arrow[uur, "p'_B"'] &
\end{tikzcd}$$

Since $p_A \circ k \circ h = p_A = p_A \circ 1_P$ and $p_B \circ k \circ h = p_B = p_B \circ 1_P$ we get by uniqueness $k \circ h = 1_P$. Symmetrically we get $h \circ k = 1_{P'}$, thus h is an isomorphism with the required properties. $\square$

We now have to show that the definition of a product satisfies the extension principle 2.3.1 (3).

2.7.7. **Proposition.** *Given sets A and B. A set P together with two maps $p'_A : P \longrightarrow A$ and $p'_B : P \longrightarrow B$ satisfies the universal property (*) of proposition 2.7.2 if and only if there is an isomorphism $h : P \longrightarrow A \times B$ such that $p_A \circ h = p'_A$ and $p_B \circ h = p'_B$.*

Proof. By proposition 2.7.2 the Cartesian product $A \times B$ and the projections $p_A : A \times B \longrightarrow A$ and $p_B : A \times B \longrightarrow B$ satisfy the universal property of proposition 2.7.4. Following the proof of proposition 2.7.2 we have shown that (P, p'_A, p'_B) also satisfies the universal property (*).
Conversely if a triple (P, p'_A, p'_B) satisfies the universal property (*) resp. (**) then by theorem 2.7.6 P and $A \times B$ are isomorphic by h with $p_A \circ h = p'_A$ and $p_B \circ h = p'_B$. $\square$

So in the future we will assume that products are given by and satisfy property (**).
All propositions and theorems of this section of products are as well valid for products with more than two factors, for products of arbitrary families of objects $(A_i|i \in I)$.

2.7.8. **Definition.** Let $(A_i|i \in I)$ be a family of objects in a category $\mathcal{C}$. An object $\prod_{i\in I} A_i$ together with a family of morphisms $(p_j : \prod_{i\in I} A_i \longrightarrow A_j|j \in I)$ is called a *product*, the morphisms $p_j : \prod_{i\in I} A_i \longrightarrow A_j$ are called *projections*, if for each object $T \in \mathcal{C}$ and each family of morphisms $(f_i : T \longrightarrow A_i|i \in I)$ there is a unique morphism $[f_i|i \in I] : T \longrightarrow \prod_{i\in I} A_i$ such that for all $j \in I$ the diagrams

$$\begin{tikzcd} T \arrow[d, "{[f_i]}"'] \arrow[dr, "f_j"] & \\ \prod_{i\in I} A_i \arrow[r, "p_j"] & A_j \end{tikzcd}$$

commute.

These products consist of an object and a family of projections and are denoted

$$(\prod_{i\in I} A_i, (p_j : \prod_{i\in I} A_i \longrightarrow A_j|j \in I)).$$

We leave the details to the reader.
If a category $\mathcal{C}$ has a product for any two objects in $\mathcal{C}$ then it is clear that it has products for any finite non-empty family of objects. If $\mathcal{C}$ has a product for any two objects in $\mathcal{C}$ and a terminal object then it has products for any finite family of objects.

2.7.9. **Definition.** (1) A category $\mathcal{C}$ is called a category with *finite products*, if there is a product in $\mathcal{C}$ for any finite family of objects (empty families included).
(2) A category $\mathcal{C}$ is called a category with *products*, if there is a product in $\mathcal{C}$ for any family of objects (empty families included). 17.06.04

The dual notion of a product the notion of a coproduct.

2.7.10. **Definition.** Let $(A_i|i \in I)$ be a family of objects in a category $\mathcal{C}$. An object $\coprod_{i\in I} A_i$ together with a family of morphisms $(\iota_j : A_j \longrightarrow \coprod_{i\in I} A_i|j \in I)$ is called a *coproduct*, the morphisms $\iota_j : A_j \longrightarrow \coprod_{i\in I} A_i$ are called *injections*, if for each object $T \in \mathcal{C}$ and each family of morphisms $(f_i : A_i \longrightarrow T | i \in I)$ there is a unique morphism $\langle f_i | i \in I \rangle : \coprod_{i \in I} A_i \longrightarrow T$ such that for all $j \in I$ the diagrams

$$\begin{tikzcd} A_j \arrow[r, "\iota_j"] \arrow[dr, "f_j"'] & \coprod_{i\in I} A_i \arrow[d, "\langle f_i \rangle"] \\ & T \end{tikzcd}$$

commute.

2.7.11. **Examples.** (1) The Cartesian products of semigroups, monoids, groups, abelian groups, modules, vector spaces, rings, algebras, Lie algebras with componentwise operations are products.

(2) The Cartesian product of topological spaces with the *product topology* is a product. Here and in the first example the more precise statement is that there exist products and that the underlying functor to sets maps these products and their projections to Cartesian products. So there is a product structure on the Cartesian products of the underlying sets, but the product structure has to be determined in each case.

(3) In the category of fields there are in general no products.

(4) Let $\mathcal{G}$ and $\mathcal{H}$ be two graphs. The product $G \times H$ in the category of graphs and homomorphisms is defined as follows: $(G \times H)_0 = G_0 \times H_0$. An arrow from (g, h) to $(g'; h')$ is a pair (a, b) with $a : g \longrightarrow g'$ in $\mathcal{G}$ and $b : h \longrightarrow h'$ in $\mathcal{H}$. The projections are the usual first and second projections.

(5) We have already seen in 1.3.8 that any poset (partially ordered set) has a corresponding category structure $\mathcal{C}(P)$. Let P be a poset and x and y two objects of $\mathcal{C}(P)$ (that is, elements of P). Let us see what, if anything, is their product. A product must be an element z together with a pair of arrows $z \longrightarrow x$ and $z \longrightarrow y$, which is just another way of saying that $z \leq x$ and $z \leq y$. The definition of product also requires that for any $w \in P$, given an arrow $w \longrightarrow x$ and one $w \longrightarrow y$, there is an arrow $w \longrightarrow z$.

This translates to $w \leq x$ and $w \leq y$ implies $w \leq z$ which, together with the fact that $z \leq x$ and $z \leq y$, characterizes z as the *infimum* of x and y, often denoted $x \wedge y$. Thus the existence of products in such a category is equivalent to the existence of infimums.

In particular, we see that products generalize a well- known construction in posets. Note that a poset that lacks infimums provides an easy example of a category without products.

(6) The product of categories as defined in 1.4.12 is the product of the categories in **Cat**.

(7) The coproduct of vector spaces is the direct sum.

(8) The coproduct of two abelian groups in the category of abelian groups is the direct sum or the product.

(9) The coproduct of two finite abelian groups ($\neq 0$) in the category of groups is an infinite nonabelian group.

2.7.12. **Proposition.** *If A is an object in a category with a terminal object 1, then*

$$1 \xleftarrow{\langle\rangle} A \xrightarrow{1_A} A$$

is a product diagram.

Proof. A cone over A and 1 has to have this form, where $f : B \longrightarrow A$ is any arrow.

$$\begin{tikzcd} & T \arrow[dl, "\langle\rangle"'] \arrow[d, "f"'] \arrow[dr, "f"] & \\ 1 & A \arrow[l, "\langle\rangle"] \arrow[r, "1_A"'] & A \end{tikzcd}$$

Clearly the only possible arrow in the middle is f. □

2.7.13. **Definition and Remark.** Our notation $A \times B$ means that $A \times B$ is the vertex of a product cone with base the discrete diagram D with $D(1) = A$ and $D(2) = B$. Then $B \times A$ denotes the product given by the diagram

$$B \xleftarrow{p_1} B \times A \xrightarrow{p_2} A$$

where we use p_1 and p_2 to avoid confusing them with the arrows proj_1 and $proj_2$ of the product diagram. (Of course, this is an ad hoc solution. If one had to deal with this situation a lot it would be necessary to introduce notation such as $\mathrm{proj}_1^{A,B}$ and $\mathrm{proj}_1^{B,A}$.) Then this is a product diagram:

$$A \xleftarrow{p_2} B \times A \xrightarrow{p_1} B$$

It follows from Theorem 2.7.6 that there is an isomorphism $[p_2, p_1] : B \times A \longrightarrow A \times B$ (called the *switch morphism*) that commutes with the projections. Its inverse is $[\mathrm{proj}_2, \mathrm{proj}_1] : A \times B \longrightarrow B \times A$.

2.8. **Equalizers.**

If A and B are sets and $f, g : A \longrightarrow B$ are functions, it is an important task to collect all solutions $a \in A$ of the equation

$$f(a) = g(a).$$

This is more general that solving an equation $f(a) = 0$ since B does not necessarily contain some element 0.
Again we want to apply the extension principle in this case.

2.8.1. **Definition.** If A and B are sets and $f, g : A \longrightarrow B$ are functions, the *equalizer* $\mathrm{Eq}(f, g)$ is the subset $\mathrm{Eq}(f, g) \subseteq A$ consisting of all elements $a \in A$ for which $f(a) = g(a)$; in short,

$$\mathrm{Eq}(f, g) = \{a \in A | f(a) = g(a)\}.$$

Together with the set $\mathrm{Eq}(f, g)$ there is the *inclusion* function $j : \mathrm{Eq}(f, g) \longrightarrow A$.

To make this into a categorical concept and apply the extension principle, we first observe that $\mathrm{Eq}(f, g)$ is a subset of A. So it has a property that cannot directly be transfered to isomorphic sets. But there is a universal property connected with this notion that will give us the correct concept.

2.8.2. **Proposition.** *Let A and B be sets and $f, g : A \longrightarrow B$ be functions. The equalizer $\mathrm{Eq}(f, g)$ together with the inclusion map $j : \mathrm{Eq}(f, g) \longrightarrow A$ satisfies the following universal property:*

- $f \circ j = g \circ j$;

- *for every set T and every map $h : T \longrightarrow A$, such that $f \circ h = g \circ h$, there is a unique map $k : B \longrightarrow \mathrm{Eq}(f,g)$ such that the diagram*

$$\begin{tikzcd} & T \arrow[dl, "k"'] \arrow[d, "h"] & \\ \mathrm{Eq}(f,g) \arrow[r, "j"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

commutes, i.e. $j \circ k = h$.

Proof. We first show the uniqueness of the map $k : T \longrightarrow \mathrm{Eq}(f,g)$: Suppose that $j : \mathrm{Eq}(f,g) \longrightarrow A$ is the inclusion function: $j(a) = a$ for $a \in \mathrm{Eq}(f,g)$. Given T and h. If $k, k' : T \longrightarrow \mathrm{Eq}(f,g)$ are maps such that $j \circ k = h = j \circ k'$ then for any $t \in T$ we have $k(t) =: a \in \mathrm{Eq}(f,g)$ and $k'(t) =: a' \in \mathrm{Eq}(f,g)$. Then we get $a = j(a) = (j \circ k)(t) = h(t) = (j \circ k')(t) = j(a') = a'$, hence $k(t) = a = h(t) = a' = k'(t)$ and $k = k'$. This shows the uniqueness of the map k.
Existence: From the uniqueness proof we obtain a reasonable definition of k. For $t \in T$ we define $k(t) := h(t)$. This is a map from T to $\mathrm{Eq}(f,g)$ since $h(t)$ satisfies $f(h(t)) = (f \circ h)(t) = (g \circ h)(t) = g(h(t))$. Then we get $j \circ k = h$. □

Obviously any pair (E, ι) isomorphic to $(\mathrm{Eq}(f,g), j)$ satisfies the same universal property. In fact given an isomorphism $i : E \longrightarrow \mathrm{Eq}(f,g)$ such that

$$\begin{tikzcd} & E \arrow[dl, "i"'] \arrow[d, "\iota"] \\ \mathrm{Eq}(f,g) \arrow[r, "j"] & A \end{tikzcd}$$

commutes. Let T be a set and $h : T \longrightarrow A$ be a map such that $f \circ h = g \circ h$. Let $k : T \longrightarrow \mathrm{Eq}(f,g)$ be the unique map that satisfies $j \circ k = h$. Then the map $i^{-1} \circ k : T \longrightarrow E$ satisfies $\iota \circ (i^{-1} \circ k) = j \circ i \circ i^{-1} \circ k = j \circ k = h$. It is an easy exercise to show that this map is unique with this property. So the universal property given in the theorem satisfies the first property of the extension principle. To define the extension of the property for arbitrary categories we use again the special description of equalizers in sets as a subsets of A.
In arbitrary categories we cannot construct an equalizer of two morphisms since there are no elements and we thus cannot construct the set of those elements on which the two maps coincide. But the extension principle 2.3.1 (2) provides the following definition.

2.8.3. **Definition.** Let A and B be objects in a category $\mathcal{C}$ and $f, g : A \longrightarrow B$ be morphisms. An object $\mathrm{Eq}(f,g)$ together with a morphisms $\iota : \mathrm{Eq}(f,g) \longrightarrow A$ is called a *(categorical) equalizer* of f and g, if for all objects T in $\mathcal{C}$ there is an isomorphism of sets

$$i : \mathrm{Mor}_{\mathcal{C}}(T, \mathrm{Eq}(f,g)) \cong \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f), \mathrm{Mor}_{\mathcal{C}}(T,g))$$

(where $\mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$ is the set-theoretic equalizer from 2.8.1) such that the diagram

$$\begin{tikzcd} & \mathrm{Mor}_{\mathcal{C}}(T,\mathrm{Eq}(f,g)) \arrow[dl, "i"'] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T,\iota)"] \\ \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g)) \arrow[r, "j"'] & \mathrm{Mor}_{\mathcal{C}}(T,A) \end{tikzcd}$$

commutes.

Frequently, $\mathrm{Eq}(f,g)$ is referred to as an equalizer of f and g without referring to j. Nevertheless, j is a crucial part of the data. 18.06.04

2.8.4. **Proposition (Characterization of equalizers by the universal mapping property).** *Given objects A and B and morphisms $f,g : A \longrightarrow B$ in a category $\mathcal{C}$. An object E together with a morphism $\iota : E \longrightarrow A$ is a (categorical) equalizer, if and only if for each object $T \in \mathcal{C}$ and each morphism $h : T \longrightarrow A$ such that such that $f \circ h = g \circ h$, there is a unique map $k : B \longrightarrow E$ such that the diagram*

$$\begin{tikzcd} & T \arrow[dl, "k"'] \arrow[d, "h"] & \\ E \arrow[r, "\iota"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

commutes, i.e. $\iota \circ k = h$.

Proof. $\Longleftarrow$: Let (E,ι) be an equalizer of f and g. Let T be an object in $\mathcal{C}$ and let $k \in \mathrm{Mor}_{\mathcal{C}}(T,E)$. Define $h := \iota \circ k : T \longrightarrow A$. Then $f \circ h = f \circ \iota \circ k = g \circ \iota \circ k = g \circ h$ and thus $\mathrm{Mor}_{\mathcal{C}}(T,f)(h) = \mathrm{Mor}_{\mathcal{C}}(T,g)(h)$ so that $h \in \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$. Now we can define $i(k) := h \in \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$ and get $\mathrm{Mor}_{\mathcal{C}}(T,\iota)(k) = \iota \circ k = h \in \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$.
It remains to show that i is bijective. We construct the inverse map

$$u : \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g)) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T,E) \qquad u(h) := k$$

where $h \in \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$ implies $\mathrm{Mor}_{\mathcal{C}}(T,f)(h) = \mathrm{Mor}_{\mathcal{C}}(T,f)(h)$. By the universal property of the equalizer there is a unique $k : T \longrightarrow L$ such that $\iota \circ k = h$. We use this k for the definition of the map u.
Let $h \in \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$. Then $(i \circ u)(h) = i(k) = \iota \circ k = h$ hence $(i \circ u) = 1_{\mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))}$.
Let $k \in \mathrm{Mor}_{\mathcal{C}}(T,E)$. Then $(u \circ i)(k) = u(h) = u(\iota \circ k) = k'$ where $k' : T \longrightarrow E$ is the unique morphism such that $\iota \circ k' = \iota \circ k$ hence $k = k'$ and $(u \circ i) = 1_{\mathrm{Mor}_{\mathcal{C}}(T,E)}$.
$\Longrightarrow$: If $h : T \longrightarrow A$ is a morphism in $\mathcal{C}$ such that $f \circ h = g \circ h$. Then h is an element of $\mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$, since $\mathrm{Mor}_{\mathcal{C}}(T,f)(h) = \mathrm{Mor}_{\mathcal{C}}(T,g)(h)$. So there is a unique morphism $k := i^{-1}(h)$ satisfying $(j \circ i)(k) = h$ or $\mathrm{Mor}_{\mathcal{C}}(T,\iota)(k) = h$ or $\iota \circ k = h$. Thus (E,ι) is an equalizer of f and g. $\square$

2.8.5. **Proposition.** *The family of isomorphisms*

$$i_T : \mathrm{Mor}_{\mathcal{C}}(T,\mathrm{Eq}(f,g)) \longrightarrow \mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f),\mathrm{Mor}_{\mathcal{C}}(T,g))$$

such that

$$\begin{tikzcd}
 & \mathrm{Mor}_{\mathcal{C}}(T, \mathrm{Eq}(f,g)) \arrow[dl, "i"'] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T,\iota)"] \\
\mathrm{Eq}(\mathrm{Mor}_{\mathcal{C}}(T,f), \mathrm{Mor}_{\mathcal{C}}(T,g)) \arrow[r, "j"'] & \mathrm{Mor}_{\mathcal{C}}(T,A)
\end{tikzcd}$$

commute for all objects T in $\mathcal{C}$, is a natural isomorphism in T and morphisms of (f,g).

Proof. is similar to the proof of Proposition 2.7.5. □

2.8.6. **Examples.** (1) In **Set**, an equalizer E of the functions $(x,y) \mapsto x^2+y^2$ and $(x,y) \mapsto 1$ from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$ is the circle $x^2+y^2 = 1$. The arrow $j : E \longrightarrow \mathbb{R} \times \mathbb{R}$ is the inclusion.
(2) Given a graph G , the inclusion of the set of loops in the graph is an equalizer of the source and target functions. This equalizer will be empty if the graph has no loops.

A theorem like 2.7.6 is true of equalizers as well.

2.8.7. **Proposition.** *If $j : E \longrightarrow A$ and $j' : E' \longrightarrow A$ are both equalizers of $f, g : A \longrightarrow B$, then there is a unique isomorphism $i : E \longrightarrow E'$ for which $j' \circ i = j$.*

Proof. We give two proofs. The first uses the concept of universal element. Let $\mathcal{C}$ be the category containing the equalizers given. Let $\mathcal{F} : \mathcal{C}^{op} \longrightarrow$ **Set** be the functor for which $\mathcal{F}(C) = \{u : C \longrightarrow A | f \circ u = g \circ u\}$, the set of arrows from C which equalize f and g. If $h : D \longrightarrow C$ let $\mathcal{F}(h)(u) = u \circ h$. This makes sense, because if $u \in F(C)$, then $f \circ u \circ h = g \circ u \circ h$, so $u \circ h \in F(D)$.
Note that this makes $\mathcal{F}$ a subfunctor of the contravariant hom functor Mor(-, A).
The definition of an equalizer of f and g can be restated this way: an equalizer of f and g is an element $j \in \mathcal{F}(E)$ for some object E such that for any $u \in F(C)$ there is a unique arrow $k : C \longrightarrow E$ such that $\mathcal{F}(k)(j) = u$. This means j is a universal element of $\mathcal{F}$, so by Proposition 2.9.3, any two equalizers $j \in F(E)$ and $j' \in F(E')$ are isomorphic by a unique arrow $i : E \longrightarrow E'$ such that $F(i)(j') = j$. That is, $j = j' \circ i$, as required.
Here is a direct proof not using universal elements: the fact that $f \circ j' = g \circ j'$ implies the existence of a unique arrow $h : E' \longrightarrow E$ such that $j' = j \circ h$. The fact that $f \circ j = g \circ j$ implies the existence of a unique arrow $h' : E \longrightarrow E'$ such that $j = j' \circ h'$. Then $j \circ h \circ h' = j' \circ h' = j = j \circ 1_E$ and the uniqueness part of the definition of equalizer implies that $h \circ h' = 1_E$. By symmetry, $h' \circ h = 1_{E'}$. □

We add some more properties of monomorphisms and equalizers.

2.8.8. **Proposition.** *Let $j : E \longrightarrow A$ be an equalizer of the pair of arrows $f, g : A \longrightarrow B$. Then j is a monomorphism. Moreover, any two equalizers of f and g belong to the same subobject of A.*

Proof. To see that j is a monomorphism, suppose $h, k : C \longrightarrow E$ with $j \circ h = j \circ k = l$. Then $f \circ l = f \circ j \circ k = g \circ j \circ k$ so there is a unique arrow $m : C \longrightarrow E$ with $j \circ m = l$. But both h and k are such arrows and so $h = k$.
Now suppose j and j' are equalizers of f and g. In the notation of Proposition 8.1.4, i and i^{-1} are the arrows required by the definition of subobject in 3.3.12, since $j' \circ i = j$ and $j \circ i^{-1} = j'$. □

8.1.6 More generally, if $f_1, \ldots, f_n$ are arrows from A to B, then an object E together with an arrow $j : E \longrightarrow A$ is an *equalizer* of $f_1, \ldots, f_n$ if it has the property that an arbitrary arrow $h : C \longrightarrow A$ factors uniquely through j if and only if $f_1 \circ h = \ldots = f_n \circ h$.
Having equalizers of parallel pairs implies having equalizers of all finite lists.

2.8.9. **Definition.** A monomorphism $e : S \longrightarrow T$ in a category is *regular* if e is an equalizer of a pair of arrows.

2.8.10. **Proposition.** *An arrow in a category that is both an epimorphism and a regular monomorphism is an isomorphism.*

Proof. Let $f : A \longrightarrow B$ be both an epimorphism and an equalizer of $g, h : B \longrightarrow C$. Since $g \circ f = h \circ f$ and f is epi, $g = h$. Then $g \circ 1_B = h \circ 1_B$ so there is a $k : B \longrightarrow A$ such that $f \circ k = 1_B$. But then $f \circ k \circ f = f = f \circ 1_A$. But f being a monomorphism can be canceled from the left to conclude that $k \circ f = 1_A$. □

2.8.11. **Proposition.** *Every monomorphism in* **Set** *is regular.*

Proof. A monomorphism in **Set** is an injective function (see Theorem 3.3.3), so let $f : A \longrightarrow B$ be an injective function. Let C be the set of all pairs $\{(b, i) | b \in B, i = 0, 1\}$ and impose an equivalence relation on these pairs forcing $(b, 0) = (b, 1)$ if and only if there is an $a \in A$ with $f(a) = b$ (and not forcing $(b, i) = (c, j)$ if b and c are distinct). Since f is injective, if such an a exists, there is only one. Let $g : B \longrightarrow C$ by $g(b) = (b, 0)$ and $h : B \longrightarrow C$ by $h(b) = (b, 1)$. Then clearly $g(b) = h(b)$ if and only if there is an $a \in A$ with $f(a) = b$. Now let $k : D \longrightarrow B$ with $g \circ k = h \circ k$. It must be that for all $x \in D$, there is an $a \in A$, and only one, such that $k(x) = f(a)$. If we let $l(x) = a$, then $l : D \longrightarrow A$ is the unique arrow with $f \circ l = k$. □

2.8.12. **Proposition.** *(1) Every split monomorphism in $\mathcal{C}$ is a regular monomorphism.*
(2) Every regular monomorphism in $\mathcal{C}$ is a monomorphism.

Proof. (1) Let $f : A \longrightarrow B$ be s split monomorphism with left inverse $g : B \longrightarrow A$, such that $g \circ f = 1_A$. We claim that (A, f) is an equalizer of $(f \circ g, 1_B)$. We have $(fg)f = f(gf) = f1_A = 1_B f$. Now let $h : T \longrightarrow B$ be a morphism with $fgh = 1_B h = h$. Then $h = f(gh)$ so that we have a factorization gh of h through f. To show uniqueness let $u : T \longrightarrow A$ satisfy $h = fu$. Then $gh = gfu = 1_A u = u$. (2) holds by Proposition 2.8.8. □

2.8.13. **Remark.** Functors preserve split monomorphisms. In general they do not preserve regular monomorphisms or monomorphisms.
The next definition can again be viewed as a generalization of a set theoretic notion. We only give a direct definition.

2.8.14. **Definition.** Let $f : A \longrightarrow B$ be a morphism in $\mathcal{C}$. A triple $(K, g : K \longrightarrow A, h : K \longrightarrow A)$ is a *kernel pair* of f if

(1) $f \circ g = f \circ h$,
(2) for each triple $(T, u : T \longrightarrow A, v : T \longrightarrow A)$ satisfying $f \circ u = f \circ v$ there is a unique $k : T \longrightarrow K$ such that $g \circ k = u$ and $h \circ k = v$, i.e. the following diagram commutes:

$$\begin{tikzcd}
 & T \arrow[dl, "k"'] \arrow[d, shift right, "u"'] \arrow[d, shift left, "v"] & \\
K \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & A \arrow[r, "f"] & B
\end{tikzcd}$$

2.8.15. **Proposition.** (1) *Let $f : B \longrightarrow C$ be a morphism in $\mathcal{C}$ that has a kernel pair $g, h : A \longrightarrow B$. If f is a coequalizer then f is a coequalizer of g and h.*

(2) *Let $g, h : A \longrightarrow B$ be a pair of morphisms that has a coequalizer $f : B \longrightarrow C$. If (A, g, h) is a kernel pair then it is a kernel pair of f.*

Proof. (1) In the diagram

$$\begin{tikzcd}
& X \arrow[dl, "x"'] \arrow[d, shift right, "u"'] \arrow[d, shift left, "v"] & \\
A \arrow[r, shift left, "h"] \arrow[r, shift right, "g"'] & B \arrow[r, "f"] \arrow[d, "k"'] & C \arrow[dl, "y"] \\
& Y &
\end{tikzcd}$$

let (C, f) be the coequalizer of (u, v). Let (A, g, h) be a kernel pair of f. Then there is a unique morphism x such that $gx = u$ and $hx = v$.
Given $k : B \longrightarrow Y$ with $kg = kh$. Then $ku = kgx = khx = kv$, so that there is a unique $y : C \longrightarrow Y$ such that $yf = k$. Thus f is a coequalizer of g and h.
(2) Let (A, g, h) be a kernel pair of k and let (C, f) be a coequalizer of (g, h). Then there is a unique morphism y such that $yf = k$.
Given $u, v : X \longrightarrow B$ with $fu = fv$. Then $ku = yfu = yfv = kv$, so that there is a unique $x : X \longrightarrow A$ such that $gx = u$ and $hx = v$. Thus (A, g, h) is a kernel pair of f. □

24.06.04

2.8.16. **Lemma.** *$f : A \longrightarrow B$ is a monomorphism if and only if $(A, 1_A, 1_A)$ is a kernel pair for f.*

Proof.
f monomorphism
$\iff \forall u, v : T \longrightarrow A : fu = fv \Longrightarrow u = v$
$\iff \forall u, v : T \longrightarrow A : fu = fv \Longrightarrow \exists_1 k : T \longrightarrow A : 1_A k = u, 1_A k = v$
$\iff (A, 1_A, 1_A)$ kernel pair of f.

□

2.8.17. **Corollary.** *If a functor preserves kernel pairs, the it preserves monomorphisms.*

Proof. Let $f : A \longrightarrow B$ be a monomorphism. Then $(A,_A, 1_A)$ is a kernel pair for f. Thus $(F(A), 1_{F(A)}, 1_{F(A)})$ is a kernel pair of $F(f)$, hence $F(f)$ is a monomorphism. □

2.9. **Universal elements.**
There is a second important way of defining a representable functor. We will show its impact with a number of examples.

2.9.1. **Definition.** Let $\mathcal{F} : \mathcal{C} \longrightarrow \mathbf{Set}$ be a covariant functor. A pair (A, x) with $A \in \mathcal{C}, x \in \mathcal{F}(A)$ is called a *universal (or generic) object* for $\mathcal{F}$, if for each $B \in \mathcal{C}$ and $y \in \mathcal{F}(B)$ there exists a unique $f \in \mathrm{Mor}_{\mathcal{C}}(A, B)$ such that $\mathcal{F}(f)(x) = y$:

$$\begin{tikzcd}
A \arrow[d, "f"'] & \mathcal{F}(A) \arrow[d, "\mathcal{F}(f)"'] \ni x \arrow[d, mapsto] \\
B & \mathcal{F}(B) \ni y
\end{tikzcd}$$

Let $\mathcal{F} : \mathcal{C} \longrightarrow \mathbf{Set}$ be a contravariant functor. A pair (A, x) with $A \in \mathcal{C}, x \in \mathcal{F}(A)$ is called a *(co-)universal (or (co-)generic) object* for $\mathcal{F}$, if for each $B \in \mathcal{C}$ and $y \in \mathcal{F}(B)$ there exists a unique $f \in \mathrm{Mor}_\mathcal{C}(B, A)$ such that $\mathcal{F}(f)(x) = y$:

$$\begin{tikzcd} A & \mathcal{F}(A) \ni x \arrow[d, "\mathcal{F}(f)"'] \arrow[d, mapsto, shift left=4] \\ B \arrow[u, "f"] & \mathcal{F}(B) \ni y \end{tikzcd}$$

The following proposition is a version of the Yoneda Lemma and could also be derived from it.

2.9.2. **Proposition.** *$\mathcal{F}$ has a universal element (A, a) if and only if $\mathcal{F}$ is representable, i.e. there is a natural isomorphism $\varphi : \mathcal{F} \cong \mathrm{Mor}_\mathcal{C}(A, -)$ (with $a = \varphi(A)^{-1}(1_A)$).*

Proof. $\Longrightarrow$: The map

$$\varphi(B) : \mathcal{F}(B) \ni y \mapsto f \in \mathrm{Mor}_\mathcal{C}(A, B) \text{ with } \mathcal{F}(f)(a) = y$$

is bijective with the inverse map

$$\psi(B) : \mathrm{Mor}_\mathcal{C}(A, B) \ni f \mapsto \mathcal{F}(f)(a) \in \mathcal{F}(B).$$

In fact we have $y \mapsto f \mapsto \mathcal{F}(f)(a) = y$ and $f \mapsto y := \mathcal{F}(f)(a) \mapsto g$ such that $\mathcal{F}(g)(a) = y$ but then $\mathcal{F}(g)(a) = y = \mathcal{F}(f)(a)$. By uniqueness we get $f = g$. Hence all $\varphi(B)$ are bijective with inverse map $\psi(B)$. It is sufficient to show that ψ is a natural transformation.
Given $g : B \longrightarrow C$. Then the following diagram commutes

$$\begin{tikzcd} \mathrm{Mor}_\mathcal{C}(A, B) \arrow[r, "\psi(B)"] \arrow[d, "\mathrm{Mor}_\mathcal{C}(A,g)"'] & \mathcal{F}(B) \arrow[d, "\mathcal{F}(g)"] \\ \mathrm{Mor}_\mathcal{C}(A, C) \arrow[r, "\psi(C)"'] & \mathcal{F}(C) \end{tikzcd}$$

since $\psi(C)\,\mathrm{Mor}_\mathcal{C}(A, g)(f) = \psi(C)(gf) = \mathcal{F}(gf)(a) = \mathcal{F}(g)\mathcal{F}(f)(a) = \mathcal{F}(g)\psi(B)(f)$.
$\Leftarrow$: Let A be given. Let $a := \varphi(A)^{-1}(1_A)$. For $y \in \mathcal{F}(B)$ we get $y = \varphi(B)^{-1}(f) = \varphi(B)^{-1}(f1_A) = \varphi(B)^{-1}\,\mathrm{Mor}_\mathcal{C}(A, f)(1_A) = \mathcal{F}(f)\varphi(A)^{-1}(1_A) = \mathcal{F}(f)(a)$ for a uniquely determined $f \in \mathrm{Mor}_\mathcal{C}(A, B)$. $\square$

2.9.3. **Proposition.** *Let (A, x) and (B, y) be universal elements for $\mathcal{F}$. Then there exists a unique isomorphism $h : A \longrightarrow B$ such that $\mathcal{F}(h)(x) = y$.*

Proof. The proof follows from the following commutative diagram as in the proof of Theorem 2.7.6:

$$\begin{tikzcd}
A \arrow[d, "h"] \arrow[dd, bend right=60, "1_A"'] & \mathcal{F}(A) \arrow[d, "\mathcal{F}(h)"'] \arrow[dd, bend left=60, "1_{\mathcal{F}(A)}"] & x \arrow[d, mapsto] \\
B \arrow[d, "k"] \arrow[dd, bend right=60, "1_A"'] & \mathcal{F}(B) \arrow[d, "\mathcal{F}(k)"'] \arrow[dd, bend left=60, "1_{\mathcal{F}(B)}"] & y \arrow[d, mapsto] \\
A \arrow[d, "h"] & \mathcal{F}(A) \arrow[d, "\mathcal{F}(h)"'] & x \arrow[d, mapsto] \\
B & \mathcal{F}(B) & y
\end{tikzcd}$$

$\square$

2.9.4. **Examples.** (1) Let R be a ring. Let $X \in \mathbf{Set}$ be a set. $\mathcal{F} : R\text{-}\mathrm{Mod} \longrightarrow \mathbf{Set}$, $\mathcal{F}(M) := \mathrm{Map}(X, M)$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the free R-module $(RX, x : X \longrightarrow RX)$ with the property, that for all $(M, y : X \longrightarrow M)$ there exists a unique $f \in \mathrm{Mor}_R(RX, M)$ such that $\mathcal{F}(f)(x) = \mathrm{Map}(X, f)(x) = fx = y$

$$\begin{tikzcd}
X \arrow[r, "x"] \arrow[dr, "y"'] & RX \arrow[d, "f"] \\
& M.
\end{tikzcd}$$

(2) Given modules M_R and ${}_RN$ (or vector spaces M_K and ${}_KN$). Define $\mathcal{F} : \mathrm{Ab} \longrightarrow \mathbf{Set}$ by $\mathcal{F}(A) := \mathrm{Bil}_R(M, N; A)$. Then $\mathcal{F}$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the tensor product $(M \otimes_R N, \otimes : M \times N \longrightarrow M \otimes_R N)$ with the property that for all $(A, f : M \times N \longrightarrow A)$ there exists a unique $g \in \mathrm{Mor}(M \otimes_R N, A)$ such that $\mathcal{F}(g)(\otimes) = \mathrm{Bil}_R(M, N; g)(\otimes) = g\otimes = f$

$$\begin{tikzcd}
M \times N \arrow[r, "\otimes"] \arrow[dr, "f"'] & M \otimes_R N \arrow[d, "g"] \\
& A.
\end{tikzcd}$$

(3) Given a $\mathcal{K}$-module V. Define $\mathcal{F} : \mathcal{K}\text{-}\mathrm{Alg} \longrightarrow \mathbf{Set}$ by $\mathcal{F}(A) := \mathrm{Mor}(V, A)$. Then $\mathcal{F}$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the tensor algebra $(T(V), \iota : V \longrightarrow T(V))$ with the property that for all $(A, f : V \longrightarrow A)$ there exists a unique $g \in \mathrm{Mor}_{\mathrm{Alg}}(T(V), A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Mor}(V, g)(\iota) = g\iota = f$

$$\begin{tikzcd}
V \arrow[r, "\iota"] \arrow[dr, "f"'] & T(V) \arrow[d, "g"] \\
& A.
\end{tikzcd}$$

(4) Given a $\mathcal{K}$-module V. Define $\mathcal{F} : \mathcal{K}\text{-}\mathrm{cAlg} \longrightarrow \mathbf{Set}$ by $\mathcal{F}(A) := \mathrm{Mor}(V, A)$. Then $\mathcal{F}$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the symmetric algebra $(S(V), \iota : V \longrightarrow S(V))$ with the property that for all $(A, f : V \longrightarrow A)$ there exists a unique $g \in \mathrm{Mor}_{\mathrm{cAlg}}(S(V), A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Mor}(V, g)(\iota) = g\iota = f$

$$\begin{tikzcd} V \arrow[r, "\iota"] \arrow[dr, "f"'] & S(V) \arrow[d, "g"] \\ & A. \end{tikzcd}$$

(5) Given a $\mathcal{K}$-module V. Define $\mathcal{F} : \mathcal{K}\text{-}\mathrm{Alg} \longrightarrow \mathbf{Set}$ by

$$\mathcal{F}(A) := \{f \in \mathrm{Mor}(V, A) | \forall v, v' \in V : f(v)f(v') = f(v')f(v)\}.$$

Then $\mathcal{F}$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the symmetric algebra $(S(V), \iota : V \longrightarrow S(V))$ with the property that for all $(A, f : V \longrightarrow A)$ such that $f(v)f(v') = f(v')f(v)$ for all $v, v' \in V$ there exists a unique $g \in \mathrm{Mor}_{\mathrm{Alg}}(S(V), A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Mor}(V, g)(\iota) = g\iota = f$

$$\begin{tikzcd} V \arrow[r, "\iota"] \arrow[dr, "f"'] & S(V) \arrow[d, "g"] \\ & A. \end{tikzcd}$$

(6) Given a $\mathcal{K}$-module V. Define $\mathcal{F} : \mathcal{K}\text{-}\mathrm{Alg} \longrightarrow \mathbf{Set}$ by

$$\mathcal{F}(A) := \{f \in \mathrm{Mor}(V, A) | \forall v \in V : f(v)^2 = 0\}.$$

Then $\mathcal{F}$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the exterior algebra $(E(V), \iota : V \longrightarrow E(V))$ with the property that for all $(A, f : V \longrightarrow A)$ such that $f(v)^2 = 0$ for all $v \in V$ there exists a unique $g \in \mathrm{Mor}_{\mathrm{Alg}}(E(V), A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Mor}(V, g)(\iota) = g\iota = f$

$$\begin{tikzcd} V \arrow[r, "\iota"] \arrow[dr, "f"'] & E(V) \arrow[d, "g"] \\ & A. \end{tikzcd}$$

(7) Let $\mathcal{K}$ be a commutative ring. Let $X \in \mathbf{Set}$ be a set. $\mathcal{F} : \mathcal{K}\text{-}\mathrm{cAlg} \longrightarrow \mathbf{Set}$, $\mathcal{F}(A) := \mathrm{Map}(X, A)$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the polynomial ring $(\mathcal{K}[X], \iota : X \longrightarrow \mathcal{K}[X])$ with the property, that for all $(A, f : X \longrightarrow A)$ there exists a unique $g \in \mathrm{Mor}_{\mathrm{cAlg}}(\mathcal{K}[X], A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Map}(X, g)(x) = g\iota = f$

$$\begin{tikzcd} X \arrow[r, "\iota"] \arrow[dr, "f"'] & \mathcal{K}[X] \arrow[d, "g"] \\ & A. \end{tikzcd}$$

(8) Let $\mathcal{K}$ be a commutative ring. Let $X \in \mathbf{Set}$ be a set. $\mathcal{F} : \mathcal{K}\text{-}\mathrm{Alg} \longrightarrow \mathbf{Set}$, $\mathcal{F}(A) := \mathrm{Map}(X, A)$ is a covariant functor. A representing object for $\mathcal{F}$ is given by the noncommutative polynomial ring $(\mathcal{K}\langle X\rangle, \iota : X \longrightarrow \mathcal{K}\langle X\rangle)$ with the property, that for all $(A, f : X \longrightarrow A)$

there exists a unique $g \in \mathrm{Mor}_{\mathrm{Alg}}(\mathcal{K}\langle X\rangle, A)$ such that $\mathcal{F}(g)(\iota) = \mathrm{Map}(X,g)(x) = g\iota = f$

$$\begin{tikzcd} X \arrow[r, "\iota"] \arrow[dr, "f"'] & \mathcal{K}\langle X\rangle \arrow[d, "g"] \\ & A. \end{tikzcd}$$

(9) Let S be a set. A function $f : \mathbb{R} \longrightarrow S$ is called *of period* 2π if $f(x+2\pi) = f(x)$ for all $x \in \mathbb{R}$. Define $\mathcal{F} : \mathbf{Set} \longrightarrow \mathbf{Set}$ by $\mathcal{F}(S) = \{f : \mathbb{R} \longrightarrow S | f \text{ is of period } 2\pi\}$. Obviously $\mathcal{F}$ is a functor. We want a ”‘universal function of period 2π”’. Let $S^1 := \mathbb{R}/2\pi\mathbb{Z}$ be the unit circle and $w : \mathbb{R} \longrightarrow S^1$ be the ”‘wrapping”’ of $\mathbb{R}$ onto S^1. Each periodic function $f : \mathbb{R} \longrightarrow S$ can be written uniquely as $\mathbb{R} \xrightarrow{w} S^1 \xrightarrow{\overline{f}} S$. This means $w \in \mathcal{F}(S^1)$ and for each $f \in \mathcal{F}(S)$ there is a unique map $\overline{f} : S^1 \longrightarrow S$ such that $\mathcal{F}(\overline{f})(w) = \overline{f} \circ w = f$. So (S^1, w) is a universal element for $\mathcal{F}$:

$$\begin{tikzcd} \mathbb{R} \arrow[r, "w"] \arrow[dr, "f"'] & S^1 \arrow[d, "\overline{f}"] \\ & S. \end{tikzcd}$$

(10) Let $\mathcal{F}$ be as in (9). Let $J := \{x \in \mathbb{R} | 0 \le x < 2\pi\}$ and $t : \mathbb{R} \longrightarrow J$ be given by $t(y) := x \iff y - x = 2\pi n$ for some $n \in \mathbb{Z}$. Obviously t is a function of period 2π. For a function $f : \mathbb{R} \longrightarrow S$ of period 2π there is a unique $\widetilde{f} : J \longrightarrow S$ such that $\mathcal{F}(\widetilde{f})(t) = f \in \mathcal{F}(S)$. Thus (J, t) is a universal element for $\mathcal{F}$:

$$\begin{tikzcd} \mathbb{R} \arrow[r, "t"] \arrow[dr, "f"'] & J \arrow[d, "\widetilde{f}"] \\ & S. \end{tikzcd}$$

By Proposition 2.9.3 we get $(S^1, w) \cong (J, t)$.

25.06.04

2.9.5. **Proposition.** *Let $\mathcal{D}$ be a category. Given a representable functor $\mathcal{F}_X : \mathcal{C} \longrightarrow$ **Set** for each $X \in \mathcal{D}$. Given a natural transformation $\mathcal{F}_g : \mathcal{F}_Y \longrightarrow \mathcal{F}_X$ for each $g : X \longrightarrow Y$ (contravariant!) such that $\mathcal{F}$ depends functorially on X, i.e. $\mathcal{F}_{1_X} = 1_{\mathcal{F}_X}, \mathcal{F}_{hg} = \mathcal{F}_g\mathcal{F}_h$. Then the representing objects (A_X, a_X) for $\mathcal{F}_X$ depend functorially on X, i.e. for each $g : X \longrightarrow Y$ there is a unique morphism $A_g : A_X \longrightarrow A_Y$ (with $\mathcal{F}_X(A_g)(a_X) = \mathcal{F}_g(A_Y)(a_Y)$) and the following identities hold $A_{1_X} = 1_{A_X}, A_{hg} = A_hA_g$. So we get a covariant functor $\mathcal{D} \ni X \longrightarrow A_X \in \mathcal{C}$.*

Proof. Choose a representing object (A_X, a_X) for $\mathcal{F}_X$ for each $X \in \mathcal{D}$ (by the axiom of choice). Then there is a unique morphism $A_g : A_X \longrightarrow A_Y$ with

$$\mathcal{F}_X(A_g)(a_X) = \mathcal{F}_g(A_Y)(a_Y) \in \mathcal{F}_X(A_Y),$$

for each $g : X \longrightarrow Y$ because $\mathcal{F}_g(A_Y) : \mathcal{F}_Y(A_Y) \longrightarrow \mathcal{F}_X(A_Y)$ is given. We have $\mathcal{F}_X(A_1)(a_X) = \mathcal{F}_1(A_X)(a_X) = a_X = \mathcal{F}_X(1)(a_X)$ hence $A_1 = 1$, and $\mathcal{F}_X(A_{hg})(a_X) = \mathcal{F}_{hg}(A_Z)(a_Z) = \mathcal{F}_g(A_Z)\mathcal{F}_h(A_Z)(a_Z) = \mathcal{F}_g(A_Z)\mathcal{F}_Y(A_h)(a_Y) = \mathcal{F}_X(A_h)\mathcal{F}_g(A_Y)(a_Y) = \mathcal{F}_X(A_h)\mathcal{F}_X(A_g)(a_X) = \mathcal{F}_X(A_hA_g)(a_X)$ hence $A_hA_g = A_{hg}$ for $g : X \longrightarrow Y$ and $h : Y \longrightarrow Z$ in $\mathcal{D}$. $\square$

2.9.6. **Corollary.** (1) $\mathrm{Map}(X, M) \cong \mathrm{Mor}_R(RX, M)$ *is a natural transformation in* M *(and in* X*!). In particular* $\mathbf{Set} \ni X \mapsto RX \in R\text{-}\mathrm{Mod}$ *is a functor.*
(2) $\mathrm{Bil}_R(M, N; A) \cong \mathrm{Mor}(M \otimes_R N, A)$ *is a natural transformation in* A *(and in* $(M, N) \in \mathrm{Mod}\text{-}R \times R\text{-}\mathrm{Mod}$*). In particular* $\mathrm{Mod}\text{-}R \times R\text{-}\mathrm{Mod} \ni M, N \mapsto M \otimes_r N \in \mathrm{Ab}$ *is a functor.*
(3) $R\text{-}\mathrm{Mod}\text{-}S \times S\text{-}\mathrm{Mod}\text{-}T \ni (M, N) \mapsto M \otimes_S N \in R\text{-}\mathrm{Mod}\text{-}T$ *is a functor.*

2.10. **The Yoneda Lemma.**

If $F : \mathcal{C} \longrightarrow \mathcal{D}$ and $G : \mathcal{C} \longrightarrow \mathcal{D}$ are functors then we denote with $\mathrm{Nat}(F, G)$ the set of natural transformations from F to G. Actually this is an illegal set, it may be too large. In our applications, however, it will be isomorphic to a set so that we don't run into problems. If $\mathcal{C}$ is a small category then there are no problems either, since each natural transformation is a family of morphisms indexed by a set.

2.10.1. **Theorem.** *(Yoneda Lemma) Let* $\mathcal{C}$ *be a category. Given a covariant functor* $\mathcal{F} : \mathcal{C} \longrightarrow \mathbf{Set}$ *and an object* $A \in \mathcal{C}$*. Then the map*

$$\pi : \mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A, \text{-}), \mathcal{F}) \ni \phi \mapsto \phi(A)(1_A) \in \mathcal{F}(A)$$

is bijective with the inverse map

$$\pi^{-1} : \mathcal{F}(A) \ni a \mapsto h^a \in \mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A, \text{-}), \mathcal{F}),$$

where $h^a(B)(f) = \mathcal{F}(f)(a)$*.*

Proof. For $\phi \in \mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A, \text{-}), \mathcal{F})$ we have a map $\phi(A) : \mathrm{Mor}_{\mathcal{C}}(A, A) \longrightarrow \mathcal{F}(A)$, hence π with $\pi(\phi) := \phi(A)(1_A)$ is a well defined map. For π^{-1} we have to check that h^a is a natural transformation. Given $f : B \longrightarrow C$ in $\mathcal{C}$. Then the diagram

$$\begin{tikzcd} \mathrm{Mor}_{\mathcal{C}}(A, B) \arrow[r, "\mathrm{Mor}(A,f)"] \arrow[d, "h^a(B)"'] & \mathrm{Mor}_{\mathcal{C}}(A, C) \arrow[d, "h^a(C)"] \\ \mathcal{F}(B) \arrow[r, "\mathcal{F}(f)"'] & \mathcal{F}(C) \end{tikzcd}$$

is commutative. In fact if $g \in \mathrm{Mor}_{\mathcal{C}}(A, B)$ then $h^a(C)\,\mathrm{Mor}_{\mathcal{C}}(A, f)(g) = h^a(C)(fg) = \mathcal{F}(fg)(a) = \mathcal{F}(f)\mathcal{F}(g)(a) = \mathcal{F}(f)h^a(B)(g)$. Thus π^{-1} is well defined.
Let $\pi^{-1}(a) = h^a$. Then $\pi\pi^{-1}(a) = h^a(A)(1_A) = \mathcal{F}(1_A)(a) = a$. Let $\pi(\phi) = \phi(A)(1_A) = a$. Then $\pi^{-1}\pi(\phi) = h^a$ and $h^a(B)(f) = \mathcal{F}(f)(a) = \mathcal{F}(f)(\phi(A)(1_A)) = \phi(B)\,\mathrm{Mor}_{\mathcal{C}}(A, f)(1_A) = \phi(B)(f)$, hence $h^a = \phi$. $\square$

2.10.2. **Corollary.** *Given* $A, B \in \mathcal{C}$*. Then the following hold*
(1) $\mathrm{Mor}_{\mathcal{C}}(A, B) \ni f \mapsto \mathrm{Mor}_{\mathcal{C}}(f, \text{-}) \in \mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(B, \text{-}), \mathrm{Mor}_{\mathcal{C}}(A, \text{-}))$ *is a bijective map.*
(2) *Under the bijective map from (1) the isomorphisms in* $\mathrm{Mor}_{\mathcal{C}}(A, B)$ *correspond to natural isomorphisms in* $\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(B, \text{-}), \mathrm{Mor}_{\mathcal{C}}(A, \text{-}))$*.*
(3) *For contravariant functors* $\mathcal{F} : \mathcal{C} \longrightarrow \mathbf{Set}$ *we have* $\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(\text{-}, A), \mathcal{F}) \cong \mathcal{F}(A)$*.*
(4) $\mathrm{Mor}_{\mathcal{C}}(A, B) \ni f \mapsto \mathrm{Mor}_{\mathcal{C}}(\text{-}, f) \in \mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(\text{-}, A), \mathrm{Mor}_{\mathcal{C}}(\text{-}, B))$ *is a bijective map that defines a one-to-one correspondence between the isomorphisms in* $\mathrm{Mor}_{\mathcal{C}}(A, B)$ *and the natural isomorphisms in* $\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(\text{-}, A), \mathrm{Mor}_{\mathcal{C}}(\text{-}, B))$*.*

Proof. (1) follows from the Yoneda Lemma with $\mathcal{F} = \mathrm{Mor}_{\mathcal{C}}(A, \text{-})$.
(2) Observe that $h^f(C)(g) = \mathrm{Mor}_{\mathcal{C}}(A, g)(f) = gf = \mathrm{Mor}_{\mathcal{C}}(f, C)(g)$ hence $h^f = \mathrm{Mor}_{\mathcal{C}}(f, \text{-})$. Since we have $\mathrm{Mor}_{\mathcal{C}}(f, \text{-})\,\mathrm{Mor}_{\mathcal{C}}(g, \text{-}) = \mathrm{Mor}_{\mathcal{C}}(gf, \text{-})$ and $\mathrm{Mor}_{\mathcal{C}}(f, \text{-}) = \mathrm{id}_{\mathrm{Mor}_{\mathcal{C}}(A, \text{-})}$ if and only if $f = 1_A$ we get the one-to-one correspondence between the isomorphisms from (1).

(3) and (4) follow by dualizing. □

2.10.3. **Remark.** The map π is a natural transformation in the arguments A and $\mathcal{F}$. More precisely: if $f : A \longrightarrow B$ and $\phi : \mathcal{F} \longrightarrow \mathcal{G}$ are given then the following diagrams commute

$$\begin{tikzcd}
\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A,\text{-}),\mathcal{F}) \arrow[r, "\pi"] \arrow[d, "{\mathrm{Nat}(\mathrm{Mor}(A,\text{-}),\phi)}"'] & \mathcal{F}(A) \arrow[d, "\phi(A)"] \\
\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A,\text{-}),\mathcal{G}) \arrow[r, "\pi"'] & \mathcal{G}(A)
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A,\text{-}),\mathcal{F}) \arrow[r, "\pi"] \arrow[d, "{\mathrm{Nat}(\mathrm{Mor}(f,\text{-}),\mathcal{F})}"'] & \mathcal{F}(A) \arrow[d, "\mathcal{F}(f)"] \\
\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(B,\text{-}),\mathcal{F}) \arrow[r, "\pi"'] & \mathcal{F}(B).
\end{tikzcd}$$

This can be easily checked. Indeed we have for $\psi : \mathrm{Mor}_{\mathcal{C}}(A,\text{-}) \longrightarrow \mathcal{F}$

$$\pi\,\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(A,\text{-}),\phi)(\psi) = \pi(\phi\psi) = (\phi\psi)(A)(1_A) = \phi(A)\psi(A)(1_A) = \phi(A)\pi(\psi)$$

and

$$\begin{aligned}
&\pi\,\mathrm{Nat}(\mathrm{Mor}_{\mathcal{C}}(f,\text{-}),\mathcal{F})(\psi) = \pi(\psi\,\mathrm{Mor}_{\mathcal{C}}(f,\text{-})) = (\psi\,\mathrm{Mor}_{\mathcal{C}}(f,\text{-}))(B)(1_B) = \psi(B)(f) \\
&= \psi(B)\,\mathrm{Mor}_{\mathcal{C}}(A,f)(1_A) = \mathcal{F}(f)\psi(A)(1_A) = \mathcal{F}(f)\pi(\psi).
\end{aligned}$$

2.10.4. **Remark.** By the previous corollary the representing object A is uniquely determined up to isomorphism by the isomorphism class of the functor $\mathrm{Mor}_{\mathcal{C}}(A,\text{-})$.

2.10.5. **Proposition.** *Let $\mathcal{G} : \mathcal{C} \times \mathcal{D} \longrightarrow$* **Set** *be a covariant bifunctor such that the functor $\mathcal{G}(C,\text{-}) : \mathcal{D} \longrightarrow$* **Set** *is representable for all $C \in \mathcal{C}$. Then there exists a contravariant functor $\mathcal{F} : \mathcal{C} \longrightarrow \mathcal{D}$ such that $\mathcal{G} \cong \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}\text{-},\text{-})$ holds. Furthermore $\mathcal{F}$ is uniquely determined by $\mathcal{G}$ up to isomorphism.*

Proof. For each $C \in \mathcal{C}$ choose an object $\mathcal{F}(C) \in \mathcal{D}$ and an isomorphism $\xi_C : \mathcal{G}(C,\text{-}) \cong \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(C),\text{-})$. Given $f : C \longrightarrow C'$ in $\mathcal{C}$ then let $\mathcal{F}(f) : \mathcal{F}(C') \longrightarrow \mathcal{F}(C)$ be the uniquely determined morphism (by the Yoneda Lemma) in $\mathcal{D}$ such that the diagram

$$\begin{tikzcd}
\mathcal{G}(C,\text{-}) \arrow[r, "\xi_C"] \arrow[d, "{\mathcal{G}(f,\text{-})}"'] & \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(C),\text{-}) \arrow[d, "{\mathrm{Mor}(\mathcal{F}(f),\text{-})}"] \\
\mathcal{G}(C',\text{-}) \arrow[r, "\xi_{C'}"'] & \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(C'),\text{-})
\end{tikzcd}$$

commutes. Because of the uniqueness of $\mathcal{F}(f)$ and because of the functoriality of $\mathcal{G}$ it is easy to see that $\mathcal{F}(fg) = \mathcal{F}(g)\mathcal{F}(f)$ and $\mathcal{F}(1_C) = 1_{\mathcal{F}(C)}$ hold and that $\mathcal{F}$ is a contravariant functor.

If $\mathcal{F}' : \mathcal{C} \longrightarrow \mathcal{D}$ is given with $\mathcal{G} \cong \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}'\text{-},\text{-})$ then $\phi : \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}\text{-},\text{-}) \cong \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}'\text{-},\text{-})$. Hence by the Yoneda Lemma $\psi(C) : \mathcal{F}(C) \cong \mathcal{F}'(C)$ is an isomorphism for all $C \in \mathcal{C}$. With these isomorphisms induced by ϕ the diagram

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(\mathcal{F}'(C), \text{-}) \arrow[r, "\mathrm{Mor}(\psi(C),\text{-})"] \arrow[d, "\mathrm{Mor}(\mathcal{F}'(f),\text{-})"'] & \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(C), \text{-}) \arrow[d, "\mathrm{Mor}(\mathcal{F}(f),\text{-})"] \\
\mathrm{Mor}_{\mathcal{D}}(\mathcal{F}'(C'), \text{-}) \arrow[r, "\mathrm{Mor}(\psi(C'),\text{-})"'] & \mathrm{Mor}_{\mathcal{D}}(\mathcal{F}(C'), \text{-})
\end{tikzcd}$$

commutes. Hence the diagram

$$\begin{tikzcd}
\mathcal{F}(C') \arrow[r, "\psi(C')"] \arrow[d, "\mathcal{F}'(f)"'] & \mathcal{F}'(C') \arrow[d, "\mathcal{F}(f)"] \\
\mathcal{F}(C) \arrow[r, "\psi(C)"'] & \mathcal{F}'(C)
\end{tikzcd}$$

commutes. Thus $\psi : \mathcal{F} \longrightarrow \mathcal{F}'$ is a natural isomorphism. $\square$

2.10.6. **Remark.** The Yoneda Lemma says in principle that we can recover (all properties of) an object A from the contravariant functor $\mathrm{Mor}_{\mathcal{C}}(\text{-}, A)$. Corollary 2.10.2 says that we can also recover a morphism f from the natural transformation $\mathrm{Mor}_{\mathcal{C}}(\text{-}, f)$. More importantly any natural transformation $\varphi : \mathrm{Mor}_{\mathcal{C}}(\text{-}, A) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(\text{-}, B)$ is of the form $\varphi = \mathrm{Mor}_{\mathcal{C}}(\text{-}, f)$ for some $f : A \longrightarrow B$. So we get the following Theorem.

2.10.7. **Theorem** (**The Yoneda Embedding**). *Let $\mathcal{C}$ be a (small) category. Then the functor $h : \mathcal{C} \longrightarrow \mathbf{Func}(\mathcal{C}^{op}, \mathbf{Set})$ with $h(A) = \mathrm{Mor}_{\mathcal{C}}(\text{-}, A)$ and $h(f) = \mathrm{Mor}_{\mathcal{C}}(\text{-}, f)$ is full and faithful. In particular $\mathcal{C}$ is equivalent to a full subcategory of $\mathbf{Func}(\mathcal{C}^{op}, \mathbf{Set})$.*

2.10.8. **Theorem** (Cayley). *Every group G is isomorphic to a group of permutations.*

Sketch of Proof.
1. First, define the Cayley representation $\overline{G}$ of G to be the following group of permutations: the underlying set of $\overline{G}$ is just G, and for each $g \in G$, we have the permutation $\overline{g}$, defined for all $h \in G$ by: $\overline{g}(h) = g \cdot h$. Now check that $\overline{g} = \overline{h}$ implies $g = h$.
2. Next define homomorphisms $i : G \longrightarrow \overline{G}$ by $i(g) = \overline{g}$, and $j : \overline{G} \longrightarrow G$ by $j(\overline{g}) = g$.
3. Finally show that $i \circ j = 1_{\overline{G}}$ and $j \circ i = 1_G$. $\square$

2.10.9. **Remark.** Warning: Note the two different levels of isomorphisms that occur in the proof of Cayley's theorem. There are permutations of the set of elements of G, which are isomorphisms in **Set**, and there is the isomorphism between G and $\overline{G}$, which is in the category **Groups** of groups and group homomorphisms.
Cayley's theorem says that any abstract group can be represented as a "concrete" one, i.e. a group of permutations of a set. The theorem can be generalized to show that any category can be represented as one that is "concrete", i.e. a category of sets and functions.

2.10.10. **Theorem.** *Every category $\mathcal{C}$ is isomorphic to one in which the objects are sets and the arrows are functions. (One should really require here that $\mathcal{C}$ is small.)*

Sketch of Proof. Define the Cayley representation of $\mathcal{C}$ to be the following concrete category:

- objects are sets of the form:
$$\overline{C} = \{f \in \mathcal{C} | \operatorname{cod}(f) = C\}$$
for all $C \in \mathcal{C}$,

- arrows are functions
$$\overline{g} : \overline{C} \longrightarrow \overline{D},$$
for $g : C \longrightarrow D$ in $\mathcal{C}$, defined by $\overline{g}(f) = g \circ f$.

$\square$

2.11. **Algebraic Structures.**

Algebraic structures are defined by operations, i.e. by composition maps from products of a set to the set itself, together with additional equations. These operations lead to natural transformations. Again monoids are the first interesting example.
We consider the category **Mon** of monoids and the underlying functor $U : \mathbf{Mon} \longrightarrow \mathbf{Set}$. First we translate the definition of a monoid into a category theoretical version.

2.11.1. **Definition.** Let $\mathcal{C}$ be a category with finite products. A *monoid* in $\mathcal{C}$ is a triple (M, mult, e_M) where

(1) M is an object in $\mathcal{C}$,
(2) $\mathrm{mult} : M \times M \longrightarrow M$ is a morphism in $\mathcal{C}$,
(3) $e_M : 1 \longrightarrow M$ is a morphism in $\mathcal{C}$ with 1 a terminal object.

such that the following diagrams commute:

$$\begin{tikzcd}
M \times M \times M \arrow[r, "M \times \mathrm{mult}"] \arrow[d, "\mathrm{mult} \times M"'] & M \times M \arrow[d, "\mathrm{mult}"] \\
M \times M \arrow[r, "\mathrm{mult}"] & M
\end{tikzcd}$$

and

$$\begin{tikzcd}
1 \times M \cong M \cong M \times 1 \arrow[r, "1_M \times e_M"] \arrow[d, "e_M \times 1_M"'] \arrow[dr, "1_M"] & M \times M \arrow[d, "\mathrm{mult}"] \\
M \times M \arrow[r, "\mathrm{mult}"] & M
\end{tikzcd}$$

We note that the unit element is replaced by a variable element $e_M : 1 \longrightarrow A$. In the category **Set** of sets a terminal object is a one-point set (or singleton). An element in $m \in M$ defines a unique map $m_M : 1 \longrightarrow M$, in fact M and $M(1)$ are isomorphic sets. In case of the unit element $m = e$ this leads to $e_M : \{*\} \longrightarrow M$ with $e_M(*) = e \in M$. So in the case of sets this
01.07.04 map makes the last diagram commutative.
The conditions for a (homo-)morphism of monoids $f : M \longrightarrow N$ translate to commutative diagrams

$$(15) \qquad \begin{tikzcd}
M \times M \arrow[r, "\mathrm{mult}"] \arrow[d, "f \times f"'] & M \arrow[d, "f"] \\
N \times N \arrow[r, "\mathrm{mult}"] & N
\end{tikzcd} \quad \text{and} \quad \begin{tikzcd}
1 \arrow[r, "e_M"] \arrow[d, "1_1 = f^0"'] & M \arrow[d, "f"] \\
1 \arrow[r, "e_N"] & N
\end{tikzcd}$$

It is easy to see that the composition of morphisms of monoids is again a morphism of monoids. One also has that the identity morphism 1_M in $\mathcal{C}$ is a morphism of monoids for all monoids (M, mult, e_M) in $\mathcal{C}$.
So we can define the *category of monoids* $\mathbf{Mon}_{\mathcal{C}}$ in an arbitrary category $\mathcal{C}$.

All other algebraic structures as long as they are defined by operations and equations (*equationally defined algebraic structures*) can be treated in the same way. Examples abound: semigroups, groups, abelian groups, rings, vector spaces over a (set-theoretic) field $\mathbb{K}$, $\mathbb{K}$-algebras, R-modules, Lie algebras, Jordan algebras, semirings, etc.
An easy example of an algebraic structure is the case of unary operations $u : S \longrightarrow S$ as discussed in 1.7.21. These operations play an important role in the study of endomorphisms of vector spaces, i.e. of square matrices.

2.11.2. **Remark** (Vector spaces)**.**
The case of $\mathbb{K}$-vector spaces and R-modules contains an interesting variant. Since $\mathbb{K}$ will not be an object in $\mathcal{C}$ in general the question arises how to define the action of $\mathbb{K}$ on vector spaces. This is done by considering each element $\alpha \in \mathbb{K}$ as an endomorphism for the vector space object V in $\mathcal{C}$, so each $\alpha \in \mathbb{K}$ induces $\alpha\cdot : V \longrightarrow V$ in $\mathcal{C}$. The relevant equations for a vector space are then written as equations of morphisms. For this purpose we first consider the structure of an abelian group on V.

2.11.3. **Definition.** Let $\mathcal{C}$ be a category with finite products. A *group in a category* $\mathcal{C}$ is a quadruple $(V, +, 0, -)$ where

(1) V is an object in $\mathcal{C}$,
(2) $+ : V \times V \longrightarrow V$ is a morphism, the *addition*,
(3) $0 : 1 \longrightarrow V$ is a morphism, the *neutral elements*,
(4) $- : V \longrightarrow V$ is a morphism, the *inverse*,

such that

$$\begin{tikzcd}
V \times V \times V \arrow[r, "V \times +"] \arrow[d, "+ \times V"'] & V \times V \arrow[d, "+"] \\
V \times V \arrow[r, "+"'] & V
\end{tikzcd}$$

$$\begin{tikzcd}
1 \times V \arrow[d, "0 \times 1_V"'] & \cong V \cong V \times 1 \arrow[r, "1_V \times 0"] \arrow[dr, "1_V"] & V \times V \arrow[d, "+"] \\
V \times V \arrow[rr, "+"] & & V
\end{tikzcd}$$

$$\begin{tikzcd}
V \arrow[r, "\langle\rangle"] \arrow[d, "{[1_V, 1_V]}"'] & 1 \arrow[r, "0"] & V \\
V \times V \arrow[rr, "1_V \times -"'] & & V \times V \arrow[u, "+"']
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
V \arrow[r, "\langle\rangle"] \arrow[d, "{[1_V, 1_V]}"'] & 1 \arrow[r, "0"] & V \\
V \times V \arrow[rr, "- \times 1_V"'] & & V \times V \arrow[u, "+"']
\end{tikzcd}$$

A group is called *abelian* or *commutative* if in addition the following diagram commutes

$$\begin{tikzcd}
V \times V \arrow[rr, "{[p_2, p_1]}"] \arrow[dr, "+"'] & & V \times V \arrow[dl, "+"] \\
& V &
\end{tikzcd}$$

where $[p_2, p_1]$ is the switch morphism of 2.7.13.

This definition allows us to define groups in many categories. Examples are groups in Top, called *topological groups*, groups in the category of analytical manifolds, called *analytical groups*, groups in the dual category of the category of commutative algebras, called *commutative Hopf algebras*. We denote the category of groups resp. abelian groups in $\mathcal{C}$ by $\mathrm{Gr}-\mathcal{C}$ resp. $\mathrm{Ab}-\mathcal{C}$.

2.11.4. **Lemma.** *Let $V = (V, +, 0, -)$ be an Abelian group in $\mathcal{C}$. Then $\mathrm{Mor}_{\mathrm{Ab}-\mathcal{C}}(V, V)$ is a ring with unit.*

Proof. The addition is given by

$$f + g := (V \xrightarrow{[1_V, 1_V]} V \times V \xrightarrow{f \times g} V \times V \xrightarrow{+} V).$$

The morphism $[1_V, 1_V]$ is usually abbreviated by Δ. It makes the following diagram commutative:

$$\begin{tikzcd} V \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & V \times V \arrow[d, "\Delta \times 1_V"] \\ V \times V \arrow[r, "1_V \times \Delta"'] & V \times V \times V \end{tikzcd}$$

i.e. it is *coassociative*. Then

$$\begin{aligned} (f+g)+h &= + \circ ((f+g) \times h) \circ \Delta \\ &= + \circ ((+ \circ (f \times g) \circ \Delta) \times h) \circ \Delta \\ &= + \circ (+ \times 1_V) \circ (f \times g \times h) \circ (\Delta \times 1_V) \circ \Delta \\ &= + \circ (1_V \times +) \circ (f \times g \times h) \circ (1_V \times \Delta) \circ \Delta \\ &= f + (g+h). \end{aligned}$$

Since we will prove a more general result in ??? we leave it to the reader to check that $V \xrightarrow{\langle\rangle} 1 \xrightarrow{0} V$ is the neutral element for the addition on $\mathrm{Mor}_{\mathrm{Ab}-\mathcal{C}}(V, V)$ and that $V \xrightarrow{f} V \xrightarrow{-} V$ is the additive inverse of f.
The multiplication on $\mathrm{Mor}_{\mathrm{Ab}-\mathcal{C}}(V, V)$ is defined to be the composition of morphisms, so it is associative with unit element.
The multiplication is distributive since

$$\begin{aligned} f \circ (g+h) &= f \circ + \circ (g \times h) \circ \Delta \\ &= + \circ (f \times f) \circ (g \times h) \circ \Delta \\ &= + \circ ((f \circ g) \times (f \circ h)) \circ \Delta \\ &= (f \circ g) + (f \circ h), \end{aligned}$$

by the fact that f is a morphism of abelian groups, and

$$\begin{aligned} (f+g) \circ h &= + \circ (f \times g) \circ \Delta \circ h \\ &= + \circ (f \times g) \circ (h \times h) \circ \Delta \\ &= + \circ ((f \circ h) \times (g \circ h)) \circ \Delta \\ &= (f \circ h) + (g \circ h), \end{aligned}$$

by an easy property of Δ. □

So a $\mathbb{K}$-vector space in $\mathcal{C}$ can be defined as follows:

2.11.5. **Definition.** Let $\mathbb{K}$ be a field. A *vector space* in a category $\mathcal{C}$ with finite products consists of an abelian group $V \in \mathcal{C}$ together with a ring homomorphism $\cdot : \mathbb{K} \longrightarrow \mathrm{Mor}_{\mathrm{Ab}-\mathcal{C}}(V, V)$.

So we get a unary operation for each element $\alpha \in \mathbb{K}$. This example shows that there may be infinitely many different operations on an algebraic structure in $\mathcal{C}$.

There is another point of view that produces many natural transformations. Again we use the example of monoids. Observe first that there is an underlying functor $U : \mathbf{Mon}_{\mathcal{C}} \longrightarrow \mathcal{C}$ like in the case of sets and ordinary monoids. Then the diagrams 15 can be rewritten as

$$\begin{tikzcd} U(M) \times U(M) \arrow[r, "\mathrm{mult}"] \arrow[d, "U(f) \times U(f)"'] & U(M) \arrow[d, "U(f)"] \\ U(N) \times U(N) \arrow[r, "\mathrm{mult}"] & U(N) \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} 1 \arrow[r, "e_M"] \arrow[d, "1_1 = f^0"'] & U(M) \arrow[d, "U(f)"] \\ 1 \arrow[r, "e_N"] & U(N) \end{tikzcd}$$

These diagrams commute for any choice of monoid morphism $f : M \longrightarrow N$. They show that mult and e are natural transformations. mult is a natural transformation from the functor $U \times U : \mathbf{Mon}_{\mathcal{C}} \longrightarrow \mathcal{C}$ to the underlying functor $U : \mathbf{Mon}_{\mathcal{C}} \longrightarrow \mathcal{C}$. It is easy the check that $U \times U$ is a functor in *one* variable. The interpretation of the second diagram is a bit more subtle. As first functor we use the one point functor $E : \mathbf{Mon}_{\mathcal{C}} \longrightarrow \mathcal{C}$ with $E((M, \mathrm{mult}, e)) := 1$ for any monoid in $\mathcal{C}$ and $E(f) := 1_1$, the identity. Then it is clear that $e : 1 \longrightarrow M$ defines also a natural transformation $e : E \longrightarrow U$.
Again any operation of any equationally defined algebraic structure can be considered as a natural transformation.
We close this section with a theorem that leads us back to the extension principle.

2.11.6. **Theorem.** *Let $\mathcal{C}$ be a category with finite products. An object G in $\mathcal{C}$ is a group if and only if the contravariant functor* $\mathrm{Mor}_{\mathcal{C}}(\text{-}, G) : \mathcal{C} \longrightarrow \mathbf{Set}$ *can be factored through the category of groups, i.e. there is a contravariant functor* $\mathcal{G} : \mathcal{C} \longrightarrow \mathrm{Gr}$ *from* $\mathcal{C}$ *to the category* Gr *of groups such that* $\mathrm{Mor}_{\mathcal{C}}(\text{-}, G) = U \circ \mathcal{G}$, *where* $U : \mathrm{Gr} \longrightarrow \mathbf{Set}$ *is the underlying functor.*

The construction given in the following proof will show more, namely that there is a bijection between the group structures on G and the different possible factorizations $\mathcal{G}$.
The statement of the Theorem also holds for any other equationally defined algebraic structures.
A given factorization $\mathcal{G}$ of $\mathrm{Mor}_{\mathcal{C}}(\text{-}, G)$ can also be viewed as a group structure for G, so we see another example of the extension principle. This variant of a definition for a group structure can also be given without any finite products, so that it is in fact more general than the definition, we have given for a group in a category.

Proof. The Yoneda Lemma defines a bijection between the set of morphisms $f : X \longrightarrow Y$ and the set of natural transformations $f(\text{-}) : X(\text{-}) \longrightarrow Y(\text{-})$ by $f(Z) = \mathrm{Mor}_{\mathcal{C}}(Z, f)$. In particular we have $m_X = \mathrm{Mor}_{\mathcal{C}}(X, m) = m(X)$. The diagram

$$\begin{tikzcd} G(\text{-}) \times G(\text{-}) \times G(\text{-}) \arrow[r, "m_{\text{-}} \times 1"] \arrow[d, "1 \times m_{\text{-}}"'] & G(\text{-}) \times G(\text{-}) \arrow[d, "m_{\text{-}}"] \\ G(\text{-}) \times G(\text{-}) \arrow[r, "m_{\text{-}}"] & G(\text{-}) \end{tikzcd}$$

commutes if and only if $\mathrm{Mor}_{\mathcal{C}}(\text{-}, m(m \times 1)) = \mathrm{Mor}_{\mathcal{C}}(\text{-}, m)(\mathrm{Mor}_{\mathcal{C}}(\text{-}, m) \times 1) = m_{\text{-}}(m_{\text{-}} \times 1) = m_{\text{-}}(1 \times m_{\text{-}}) = \mathrm{Mor}_{\mathcal{C}}(\text{-}, m)(1 \times \mathrm{Mor}_{\mathcal{C}}(\text{-}, m)) = \mathrm{Mor}_{\mathcal{C}}(\text{-}, m(1 \times m))$ if and only if $m(m \times 1) =$

$m(1 \times m)$ if and only if the diagram

$$\begin{tikzcd}
G \times G \times G \arrow[r, "m\times 1"] \arrow[d, "1\times m"'] & G \times G \arrow[d, "m"] \\
G \times G \arrow[r, "m"] & G
\end{tikzcd}$$

commutes. In a similar way one shows the equivalence of the other diagram(s). $\square$

02.07.04

2.11.7. **Remark.** Let $\mathcal{C}$ be a category with finite products. A morphism $f : G \longrightarrow G'$ in $\mathcal{C}$ is a homomorphism of groups if and only if

$$\begin{tikzcd}
G \times G \arrow[r, "m"] \arrow[d, "f\times f"'] & G \arrow[d, "f"] \\
G' \times G' \arrow[r, "m'"] & G'
\end{tikzcd}$$

commutes.

2.11.8. **Definition.** A *cogroup* (*comonoid*) G in $\mathcal{C}$ is a group (monoid) in $\mathcal{C}^{op}$, i.e. an object $G \in \operatorname{Ob}\mathcal{C} = \operatorname{Ob}\mathcal{C}^{op}$ together with a natural transformation $m(X) : G(X) \times G(X) \longrightarrow G(X)$ where $G(X) = \operatorname{Mor}_{\mathcal{C}^{op}}(X, G) = \operatorname{Mor}_{\mathcal{C}}(G, X)$, such that $(G(X), m(X))$ is a group (monoid) for each $X \in \mathcal{C}$.

2.11.9. **Remark.** Let $\mathcal{C}$ be a category with finite (categorical) coproducts. An object G in $\mathcal{C}$ carries the structure $m : G(\text{-}) \times G(\text{-}) \longrightarrow G(\text{-})$ of a cogroup in $\mathcal{C}$ if and only if there are morphisms $\Delta : G \longrightarrow G \amalg G$, $\varepsilon : G \longrightarrow I$, and $S : G \longrightarrow G$ such that the diagrams

$$\begin{tikzcd}
G \arrow[r, "\Delta"] \arrow[d, "\Delta"'] & G \amalg G \arrow[d, "1\amalg\Delta"] \\
G \amalg G \arrow[r, "\Delta\amalg 1"] & G \amalg G \amalg G
\end{tikzcd}
\qquad
\begin{tikzcd}
G \arrow[rr, "\Delta"] \arrow[d, "\Delta"'] \arrow[drr, "1"] & & G \amalg G \arrow[d, "1\amalg\varepsilon"] \\
G \amalg G \arrow[r, "\varepsilon\amalg 1"] & I \amalg G \cong & G \cong G \amalg I
\end{tikzcd}$$

$$\begin{tikzcd}
G \arrow[r] \arrow[d, "\Delta"'] & I \arrow[r, "\varepsilon"] & G \\
G \amalg G \arrow[rr, "1\amalg S", "S\amalg 1"'] & & G \amalg G \arrow[u, "\nabla"']
\end{tikzcd}$$

commute where ∇ is dual to the morphism Δ defined in [?] **??**. The multiplications are related by $\Delta_X = \operatorname{Mor}_{\mathcal{C}}(\Delta, X) = \Delta(X)$.

Let $\mathcal{C}$ be a category with finite coproducts and let G and G' be cogroups in $\mathcal{C}$. Then a homomorphism of groups $f : G' \longrightarrow G$ is a morphism $f : G \longrightarrow G'$ in $\mathcal{C}$ such that the diagram

$$\begin{tikzcd}
G \arrow[r, "\Delta"] \arrow[d, "f"'] & G \times G \arrow[d, "f\times f"] \\
G' \arrow[r, "\Delta'"] & G' \times G'
\end{tikzcd}$$

commutes. An analogous result holds for comonoids.

2.11.10. **Remark.** Obviously similar observations and statements can be made for other algebraic structures in a category $\mathcal{C}$. So one can introduce vector spaces and covector spaces, monoids and comonoids, rings and corings in a category $\mathcal{C}$. The structures can be described by morphisms in $\mathcal{C}$ if $\mathcal{C}$ is a category with finite (co-)products.

2.11.11. **Proposition.** *Let $G \in \mathcal{C}$ be a group with multiplication $a * b$, unit e, and inverse a^{-1} in $G(X)$. Then the morphisms $m : G \times G \longrightarrow G$, $u : E \longrightarrow G$, and $S : G \longrightarrow G$ are given by*

$$m = p_1 * p_2, \qquad u = e_E, \qquad S = \mathrm{id}_G^{-1}.$$

Proof. By the Yoneda Lemma 2.10.1 these morphisms can be constructed from the natural transformation as follows. Under $\mathrm{Mor}_{\mathcal{C}}(G\times G, G\times G) = G\times G(G\times G) \cong G(G\times G)\times G(G\times G) \xrightarrow{*} G(G\times G) = \mathrm{Mor}_{\mathcal{C}}(G\times G, G)$ the identity $\mathrm{id}_{G\times G} = (p_1, p_2)$ is mapped to $m = p_1 * p_2$. Under $\mathrm{Mor}_{\mathcal{C}}(E, E) = E(E) \longrightarrow G(E) = \mathrm{Mor}_{\mathcal{C}}(E, G)$ the identity of E is mapped to the neutral element $u = e_E$. Under $\mathrm{Mor}_{\mathcal{C}}(G, G) = G(G) \longrightarrow G(G) = \mathrm{Mor}_{\mathcal{C}}(G, G)$ the identity is mapped to its $*$-inverse $S = \mathrm{id}_G^{-1}$. $\square$

2.11.12. **Corollary.** *Let $G \in \mathcal{C}$ be a cogroup with multiplication $a * b$, unit e, and inverse a^{-1} in $G(X)$. Then the morphisms $\Delta : G \longrightarrow G \amalg G$, $\varepsilon : G \longrightarrow I$, and $S : G \longrightarrow G$ are given by*

$$\Delta = \iota_1 * \iota_2, \qquad \varepsilon = e_I, \qquad S = \mathrm{id}_G^{-1}.$$

2.12. Limits.

Equalizers and products are special cases of the general notion of a (categorical) limit. Limits can also be described as a generalization of a set theoretic construction with the extension principle. We will, however, use the short way and give a direct definition and cover the relation to limits on sets by an extra proposition.

2.12.1. **Definition.** (1) Let $\mathcal{D}$ be a (shape) graph and $F : \mathcal{D} \longrightarrow \mathcal{C}$ be a diagram. A *limit* or *projective limit* of F consists of

- an object $\varprojlim F$ in $\mathcal{C}$ together with
- a family of morphisms $(p_A : \varprojlim F \longrightarrow F(A) |$ for all vertices $A \in \mathcal{D})$ called *projections*, with commuting triangles

$$\begin{tikzcd} & \varprojlim F \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & \\ F(A) \arrow[rr, "F(f)"'] & & F(B) \end{tikzcd}$$

  for all edges $f : A \longrightarrow B$ in $\mathcal{D}$,

if the following universal property is satisfied:

(*) for each
  - object T in $\mathcal{C}$ and each
  - family of morphisms $(f_A : T \longrightarrow F(A) |$ for all vertices $A \in \mathcal{D})$, with commuting triangles

$$\begin{tikzcd} & T \arrow[dl, "f_A"'] \arrow[dr, "f_B"] & \\ F(A) \arrow[rr, "F(f)"'] & & F(B) \end{tikzcd}$$

for all edges $f : A \longrightarrow B$ in $\mathcal{D}$,
there is a unique morphism $g : T \longrightarrow \varprojlim F$ such that the triangles

$$\begin{tikzcd} T \arrow[rr, "g"] \arrow[dr, "f_A"'] & & \varprojlim F \arrow[dl, "p_A"] \\ & F(A) & \end{tikzcd}$$

commute for all vertices $A \in \mathcal{D}$.
(2) If there is a limit for each diagram $F : \mathcal{D} \longrightarrow \mathcal{C}$ of shape $\mathcal{D}$ then $\mathcal{C}$ is called *$\mathcal{D}$-complete*. We also say that $\mathcal{C}$ *has limits of shape* $\mathcal{D}$.
(3) If there is a limit for all diagrams $F : \mathcal{D} \longrightarrow \mathcal{C}$ in $\mathcal{C}$ (for all $\mathcal{D}$), then $\mathcal{C}$ is called *complete*.

2.12.2. **Remark.** Limits are unique up to isomorphism. This can be proved essentially the same way as the uniqueness of products up to isomorphism (Proposition 2.7.6) or the uniqueness of equalizers up to isomorphism (Proposition 2.8.7). It is indeed a special case of the uniqueness of a universal element (Proposition 2.9.3). The connection with a universal element will be given by the next theorem.

2.12.3. **Examples.** (1) Let $\mathcal{D}$ be an empty graph. Then a limit of a diagram consists of an object E, such that for each object T there is a unique morphism $g : T \longrightarrow E$. This is the definition of a terminal object E.
(2) Let $\mathcal{D}$ be a set of two vertices without any edges. Then a diagram consists of two objects A and B. A limit consists of an object $A \times B$ and two morphisms $p_A : A \times B \longrightarrow B$ and $p_B : A \times B \longrightarrow B$. The universal property says that for each object T and any two morphisms $f_A : T \longrightarrow A$ and $f_B : T \longrightarrow A \times B$ there is a unique morphism $(f, g) : T \longrightarrow A \times B$ such that $p_A \circ (f_A, f_B) = f_A$ and $p_B \circ (f_A, f_B) = f_B$. So this gives the definition of a product of two objects.
If $\mathcal{D}$ has an arbitray set I of vertices and an empty set of edges, then a limit of $F : \mathcal{D} \longrightarrow \mathcal{C}$
08.07.04 is the product $\prod_{i \in I} A_i$ with $A_i := F(i)$.
(3) Let $\mathcal{D}$ consist of two vertices 1 and two and two edges

$$\begin{tikzcd} 1 \arrow[r, shift left, "i"] \arrow[r, shift right, "j"'] & 2. \end{tikzcd}$$

Then a limit of a diagram

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

consists of an object $K := \varprojlim F$ and two morphisms $\iota : K \longrightarrow A$ and $\lambda : KtoB$ such that $f \circ \iota = \lambda = g \circ \iota$ which is universal. Obviously λ is uniquely determined by ι, so it is superfluous and we have the additional equation

$$f \circ \iota = g \circ \iota.$$

So in this case we obtain the notion of an equalizer.
(4) Let

$$\mathcal{D} \quad = \quad \begin{tikzcd} & 1 \arrow[d, "i"] \\ 2 \arrow[r, "j"'] & 3. \end{tikzcd}$$

A limit of the diagram

$$\begin{tikzcd} & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

consists of an object $A \times_C B := \varprojlim F$ and two morphisms $\mathfrak{p}_A : A \times_C B \longrightarrow A$ and $p_B : A \times_C B \longrightarrow B$ such that the diagram

$$\begin{tikzcd} A \times_C B \arrow[r, "p_A"] \arrow[d, "p_B"'] & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

The third morphism $p_C : A \times_C B \longrightarrow C$ is given by the property $p_C = f \circ p_A = g \circ p_B$, so it is uniquely determined. The triple $(A \times_C B, p_A, p_B)$ satisfies the fowwlowing universal property. Given an object T in $\mathcal{C}$ and two morphisms $q_A : T \longrightarrow A$ and $q_B : T \longrightarrow B$ such that $f \circ q_A = g \circ q_B$ (again we need not give the third morphism $q_C = f \circ q_A = g \circ q_B$) then there is a unique morphism $q : T \longrightarrow A \times_C B$ such that

$$\begin{tikzcd} T \arrow[dr, "q"] \arrow[drr, "q_A"] \arrow[ddr, "q_B"'] & & \\ & A \times_C B \arrow[r, "p_A"'] \arrow[d, "p_B"] & A \arrow[d, "f"] \\ & B \arrow[r, "g"'] & C \end{tikzcd}$$

commutes. This limit $(A \times_C B, p_A, p_B)$ is called a *fibre product* or a *pullback*.

(5) The limit $F : \mathcal{D} \longrightarrow \mathbf{Set}$ of a shape graph $\mathcal{D}$ with vertex set $V = V(\mathcal{D})$ and edge set $E = E(\mathcal{D})$ can be described as follows. The set $\varprojlim F$ is

$$\varprojlim F = \{\ (a_i) \mid \text{ for all } i \in V : a_i \in F(i) \text{ and for all } (u : i \longrightarrow j) \in E : F(u)(a_i) = a_j\}.$$

The projections $p_i : \varprojlim F \longrightarrow F(i)$ are defined by $p_i((a_j)) := a_i$.

2.12.4. **Proposition** (**Characterization of limits).** *Let $F : \mathcal{D} \longrightarrow \mathcal{C}$ be a diagram. An object L together with a family of morphisms $(p_A : L \longrightarrow F(A))$ for all $A \in \mathcal{D}$ is a limit of $F : \mathcal{D} \longrightarrow \mathcal{C}$ if and only if*

- *for all objects T in $\mathcal{C}$ there is an isomorphism of sets*

$$i : \mathrm{Mor}_{\mathcal{C}}(T, L) \cong \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T, F) : \mathcal{D} \longrightarrow \mathbf{Set})$$

  such that the tringles

$$\begin{tikzcd} & \mathrm{Mor}_{\mathcal{C}}(T, L) \arrow[dl, "i"'] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T, p_A)"] \\ \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T, F)) \arrow[r, "\widetilde{p}_A"'] & \mathrm{Mor}_{\mathcal{C}}(T, A) \end{tikzcd}$$

  commute for all $A \in \mathcal{D}$.

Frequently, $\varprojlim F$ is referred to as a limit of $F : \mathcal{D} \longrightarrow \mathcal{C}$ without referring to (p_A). Nevertheless, (p_A) is a crucial part of the data.

Proof. $\Longrightarrow$: Let $(L,(p_A))$ be a limit of $F : \mathcal{D} \longrightarrow \mathcal{C}$. Let T be an object in $\mathcal{C}$ and let $f \in \mathrm{Mor}_{\mathcal{C}}(T,L)$. Define $f_A := p_A \circ f : T \longrightarrow A$. Then $\mathrm{Mor}_{\mathcal{C}}(T,F)(u)(f_A) = \mathrm{Mor}_{\mathcal{C}}(T,F(u))(p_A \circ f) = F(u) \circ p_A \circ f = p_B \circ f = f_B$ for all $(u : A \longrightarrow B) \in \mathcal{D}$ so that $(f_A) \in \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F)$. Now we can define $i(f) := (f_A) \in \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F)$ and get $\mathrm{Mor}_{\mathcal{C}}(T,p_A)(f) = p_A \circ f = f_A = \widetilde{p}_A \circ i(f)$.

It remains to show that i is bijective. We construct the inverse map as

$$u : \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F)) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(T,L) \qquad u((f_A)) := f$$

where $(f_A) \in \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F))$ implies $\mathrm{Mor}_{\mathcal{C}}(T,F)(u)(f_A) = f_B = F(u) \circ f_A$ for all $(u : A \longrightarrow B) \in \mathcal{D}$. By the universal property of the limit there is a unique $f : T \longrightarrow L$ such that $p_A \circ f = f_A$. We use this f for the definition of the map u.

Let $(f_A) \in \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F))$. Then $(i \circ u)((f_A)) = i(f) = (p_A \circ f) = (f_A)$ hence $(i \circ u) = 1_{\varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F))}$.

Let $f \in \mathrm{Mor}_{\mathcal{C}}(T,L)$. Then $(u \circ i)(f) = u((f_A)) = u((p_A) \circ f) = f'$ where $f' : T \longrightarrow L$ is the unique morphism such that $p_A \circ f' = p_A \circ f$ for all $A \in \mathcal{D}$ hence $f = f'$ and $(u \circ i) = 1_{\mathrm{Mor}_{\mathcal{C}}(T,L)}$.

$\Longleftarrow$: If $(f_A : T \longrightarrow F(A) \mid A \in \mathcal{D})$ is a family of morphisms such that $F(u) \circ f_A = f_B$ for all $(u : A \longrightarrow B) \in \mathcal{D}$. Then (f_A) is an element of $\varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F))$, since $\mathrm{Mor}_{\mathcal{C}}(T,F)(u)(f_a) = F(u) \circ f_A = f_B$ for all $(u : A \longrightarrow B) \in \mathcal{D}$. It is the unique element with $\widetilde{p}_A((f_A)) = f_A$ for all $A \in \mathcal{D}$.
So there is a unique morphism $f := i^{-1}((f_A))$ satisfying $\widetilde{p}_A \circ i(f) = f_A$ or $\mathrm{Mor}_{\mathcal{C}}(T,p_A)(f) = f_A$ or $p_A \circ f = f_A$ for all $A \in \mathcal{D}$. Thus $(L,(p_A))$ is a limit of F. $\square$

2.12.5. **Proposition.** *The family of isomorphisms*

$$i_T : \mathrm{Mor}_{\mathcal{C}}(T, \varprojlim F) \longrightarrow \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F)$$

such that the tringles

$$\begin{tikzcd} & \mathrm{Mor}_{\mathcal{C}}(T,\varprojlim F) \arrow[dl, "i"'] \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(T,p_A)"] \\ \varprojlim(\mathrm{Mor}_{\mathcal{C}}(T,F)) \arrow[r, "\widetilde{p}_A"'] & \mathrm{Mor}_{\mathcal{C}}(T,A) \end{tikzcd}$$

commute for all objects T in $\mathcal{C}$, is a natural isomorphism in T and morphisms of diagrams F.

Proof. is similar to the proof of Proposition 2.7.5. $\square$

2.12.6. **Remark.** Thus the most important properties of the extension principle are now verified. Observe that Proposition 2.12.4 implies

$$\mathrm{Mor}_{\mathcal{C}}(T, \varprojlim F) \cong \varprojlim \mathrm{Mor}_{\mathcal{C}}(T,F).$$

The dual notion of a limit is a *colimit* or *direct limit*. You can define it as the limit in the dual category $\mathcal{C}^{op}$ of $\mathcal{C}$. Then for colimits the following isomorphism holds:

$$\mathrm{Mor}_{\mathcal{C}}(\varinjlim F, T) \cong \varprojlim \mathrm{Mor}_{\mathcal{C}}(F, T).$$

These isomorphisms can be interpreted as

- a covariant representable functor preserves limits;
- a contravariant representable functor maps colimits to limits.

2.12.7. **Example.** Let $(V_i | i \in I)$ be a family of vector spaces. Then

$$\mathrm{Hom}_{\mathbb{K}}(W, \prod_{i \in I} V_i) \cong \prod_{i \in I} \mathrm{Hom}_{\mathbb{K}}(W, V_i)$$

and

$$\mathrm{Hom}_{\mathbb{K}}(\bigoplus_{i \in I} V_i, W) \cong \prod_{i \in I} \mathrm{Hom}_{\mathbb{K}}(V_i, W)$$

2.12.8. **Theorem.** *Let $F : \mathcal{D} \longrightarrow \mathcal{C}$ be a diagram in a category $\mathcal{C}$. If $(L, (p_A))$ and $(L', (p'_A))$ are limits of F, then there is a unique isomorphism $h : L \longrightarrow L'$ such that $p'_A \circ h = p_A$ for all $A \in \mathcal{D}$.*

Proof. Since $(L', (p'_A))$ satisfies the universal property of a limit there is a unique morphism $h : L \longrightarrow L'$ such that $p'_A \circ h = p_A$ for all $A \in \mathcal{D}$. Symmetrically there is a unique morphism $k : L' \longrightarrow L$ such that $p_A \circ k = p'_A$ for all $A \in \mathcal{D}$. Consider the following commutative diagram:

$$\begin{tikzcd}
 & L \arrow[d, "h"] \arrow[ddl, "p_A"'] \arrow[ddd, bend left, "1_L"] \\
 & L' \arrow[dd, "k"] \arrow[dl, "p'_A"] \arrow[ddd, bend right, "1_{L'}"'] \\
A & \\
 & L \arrow[ul, "p_A"'] \arrow[d, "h"] \\
 & L' \arrow[uul, "p'_A"]
\end{tikzcd}$$

Since $p_A \circ k \circ h = p_A = p_A \circ 1_L$ for all $A \in \mathcal{D}$ we get by uniqueness $k \circ h = 1_L$. Symmetrically we get $h \circ k = 1_{L'}$, thus h is an isomorphism with the required properties. $\square$ 09.07.04

3. Adjoint functors and monads

3.1. **Adjoint functors.**

3.1.1. **Definition.** Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Let $F : \mathcal{C} \longrightarrow \mathcal{D}$ and $G : \mathcal{D} \longrightarrow \mathcal{C}$ be covariant functors. F is called *left adjoint to* G and G is called *right adjoint* to F if there is a natural isomorphism (natural in $C \in \mathcal{C}$ and $D \in \mathcal{D}$)

$$\varphi(C, D) : \mathrm{Mor}_{\mathcal{D}}(FC, D) \cong \mathrm{Mor}_{\mathcal{C}}(C, GD)$$

or simply

$$\mathrm{Mor}_{\mathcal{D}}(F\text{-}, \text{-}) \cong \mathrm{Mor}_{\mathcal{C}}(\text{-}, G\text{-}) : \mathcal{C}^{op} \times \mathcal{D} \longrightarrow \mathbf{Set}\,.$$

3.1.2. **Proposition.** *Let $F : \mathcal{C} \longrightarrow \mathcal{D}$ be left adjoint to $G : \mathcal{D} \longrightarrow \mathcal{C}$. Then G determines F uniquely up to a natural isomorphism.*

Proof. Assume that also $F' : \mathcal{C} \longrightarrow \mathcal{D}$ is left adjoint to G. Then we have natural isomorphisms

$$\mathrm{Mor}_{\mathcal{D}}(F\text{-}, \text{-}) \cong \mathrm{Mor}_{\mathcal{C}}(\text{-}, G\text{-}) \cong \mathrm{Mor}_{\mathcal{D}}(F'\text{-}, \text{-}).$$

By the Yoneda Lemma and Proposition 2.10.5 we get $F \cong F'$. □

3.1.3. **Corollary.** *Let $F_i : \mathcal{C} \longrightarrow \mathcal{D}$ be left adjoint to $G_i : \mathcal{D} \longrightarrow \mathcal{C}$ for $i = 1, 2$. Let $\alpha : G_1 \longrightarrow G_2$ be a natural transformation. Then there is a unique natural transformation $\beta : F_2 \longrightarrow F_1$ such that the diagram*

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(F_1\text{-}, \text{-}) \arrow[r, "\varphi_1(\text{-},\text{-})"] \arrow[d, "\mathrm{Mor}_{\mathcal{D}}(\beta\text{-}, \text{-})"'] & \mathrm{Mor}_{\mathcal{C}}(\text{-}, G_1\text{-}) \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(\text{-}, \alpha\text{-})"] \\
\mathrm{Mor}_{\mathcal{D}}(F_2\text{-}, \text{-}) \arrow[r, "\varphi_2(\text{-},\text{-})"'] & \mathrm{Mor}_{\mathcal{C}}(\text{-}, G_2\text{-})
\end{tikzcd}$$

Proof. Given $A, B \in \mathcal{D}$. Then there is a unique morphism $\beta(A) : F_2A \longrightarrow F_1A$ such that the following diagram commutes

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(F_1A, \text{-}) \arrow[r, "\varphi_1(A,\text{-})"] \arrow[d, "\mathrm{Mor}_{\mathcal{D}}(\beta(A), \text{-})"'] & \mathrm{Mor}_{\mathcal{C}}(A, G_1\text{-}) \arrow[d, "\mathrm{Mor}_{\mathcal{C}}(A, \alpha\text{-})"] \\
\mathrm{Mor}_{\mathcal{D}}(F_2A, \text{-}) \arrow[r, "\varphi_2(A,\text{-})"'] & \mathrm{Mor}_{\mathcal{C}}(A, G_2\text{-})
\end{tikzcd}$$

We have to show that $\beta : F_2 \longrightarrow F_1$ is a natural transformation. Let $f : A \longrightarrow B$ be a morphism in $\mathcal{D}$. Then

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(F_1B,\text{-}) \arrow[rr, "\varphi_1(B,\text{-})"] \arrow[dr, "\mathrm{Mor}_{\mathcal{D}}(F_1f,\text{-})"] \arrow[dd, "\mathrm{Mor}_{\mathcal{D}}(\beta(B),\text{-})"'] & & \mathrm{Mor}_{\mathcal{C}}(B,G_1\text{-}) \arrow[dr, "\mathrm{Mor}_{\mathcal{C}}(f,G_1\text{-})"] \arrow[dd, "\mathrm{Mor}_{\mathcal{C}}(B,\alpha\text{-})"] & \\
& \mathrm{Mor}_{\mathcal{D}}(F_1A,\text{-}) \arrow[rr, "\varphi_1(A,\text{-})"] \arrow[dd, "\mathrm{Mor}_{\mathcal{D}}(\beta(A),\text{-})"'] & & \mathrm{Mor}_{\mathcal{C}}(A,G_1\text{-}) \arrow[dd, "\mathrm{Mor}_{\mathcal{C}}(A,\alpha\text{-})"] \\
\mathrm{Mor}_{\mathcal{D}}(F_2B,\text{-}) \arrow[rr, "\varphi_2(B,\text{-})"'] \arrow[dr, "\mathrm{Mor}_{\mathcal{D}}(F_2f,\text{-})"'] & & \mathrm{Mor}_{\mathcal{C}}(B,G_2\text{-}) \arrow[dr, "\mathrm{Mor}_{\mathcal{C}}(f,G_2\text{-})"'] & \\
& \mathrm{Mor}_{\mathcal{D}}(F_2A,\text{-}) \arrow[rr, "\varphi_2(A,\text{-})"'] & & \mathrm{Mor}_{\mathcal{C}}(A,G_2\text{-})
\end{tikzcd}$$

□

There is an equivalent definition of adjoint functors that we want to give now. First we need some preparation.

3.1.4. **Lemma.** *Let $F : \mathcal{C} \longrightarrow \mathcal{D}$ and $G : \mathcal{D} \longrightarrow \mathcal{C}$ be functors. Then the map*

$$\mathrm{Nat}(\mathrm{Id}_{\mathcal{C}}, GF) \ni \Phi \mapsto G\text{-} \circ \Phi\text{-} \in \mathrm{Nat}(\mathrm{Mor}_{\mathcal{D}}(F\text{-},\text{-}), \mathrm{Mor}_{\mathcal{C}}(\text{-},G\text{-}))$$

is bijective with inverse map

$$\mathrm{Nat}(\mathrm{Mor}_{\mathcal{D}}(F\text{-},\text{-}), \mathrm{Mor}_{\mathcal{C}}(\text{-},G\text{-})) \ni \varphi \mapsto \varphi(\text{-},F\text{-})(1_{F\text{-}}) \in \mathrm{Nat}(\mathrm{Id}_{\mathcal{C}}, GF).$$

Given $\varphi : \mathrm{Mor}_{\mathcal{D}}(F\text{-},\text{-}) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(\text{-},G\text{-})$ as a family $\varphi(C,D) : \mathrm{Mor}_{\mathcal{D}}(FC,D) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(C,GD)$. Take $D := FC$. Then we get $\Phi(C) := \varphi(C,FC)(1_{FC}) : C \longrightarrow GFC$. We show that $\Phi : \mathrm{Id}_{\mathcal{C}} \longrightarrow GF$ is a natural transformation.
For $f : C \longrightarrow C'$ the diagram commutes:

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(FC,FC) \arrow[r, "\mathrm{Mor}_{\mathcal{D}}(FC,Ff)"] \arrow[d, "\varphi(C,FC)"'] & \mathrm{Mor}_{\mathcal{D}}(FC,FC') \arrow[d, "\varphi(C,FC')"'] & \mathrm{Mor}_{\mathcal{D}}(FC',FC') \arrow[l, "\mathrm{Mor}_{\mathcal{D}}(Ff,FC')"'] \arrow[d, "\varphi(C',FC')"] \\
\mathrm{Mor}_{\mathcal{C}}(C,GFC) \arrow[r, "\mathrm{Mor}_{\mathcal{C}}(C,GFf)"] & \mathrm{Mor}_{\mathcal{C}}(C,GFC') & \mathrm{Mor}_{\mathcal{C}}(C',GFC') \arrow[l, "\mathrm{Mor}_{\mathcal{C}}(f,GFC')"']
\end{tikzcd}$$

Hence we get

$$\begin{aligned}
\Phi(C') \circ f \quad &= (\mathrm{Mor}_{\mathcal{C}}(f,GFC') \circ \varphi(C',FC'))(1_{FC'}) \\
&= (\varphi(C,FC') \circ \mathrm{Mor}_{\mathcal{D}}(Ff,FC'))(1_{FC'}) \\
&= \varphi(C,FC')(Ff) \\
&= (\varphi(C,FC') \circ \mathrm{Mor}_{\mathcal{D}}(FC,Ff))(1_{FC}) \\
&= (\mathrm{Mor}_{\mathcal{C}}(C,GFf) \circ \varphi(C,FC))(1_{FC}) \\
&= GFf \circ \Phi(C).
\end{aligned}$$

Conversely let $\Phi : \mathrm{Id}_{\mathcal{C}} \longrightarrow GF$ be a natural transformation and let

$$\varphi(C,D)(f) := Gf \circ \Phi(C) = (\mathrm{Mor}_{\mathcal{C}}(\Phi(C),GD) \circ G)(f) : C \stackrel{\Phi(C)}{\longrightarrow} GFC \stackrel{Gf}{\longrightarrow} GD$$

for $f : FC \longrightarrow D$. We show that $\varphi : \mathrm{Mor}_{\mathcal{D}}(F\text{--}) \longrightarrow \mathrm{Mor}_{\mathcal{C}}(\text{-}, G\text{-})$ is a natural transformation. Given $g : C \longrightarrow C'$ then

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(FC', D) \arrow[r, "\mathrm{Mor}_{\mathcal{D}}(Fg, D)"] \arrow[d, "\varphi(C', D)"'] & \mathrm{Mor}_{\mathcal{D}}(FC, D) \arrow[d, "\varphi(C, D)"] \\
\mathrm{Mor}_{\mathcal{C}}(C', GD) \arrow[r, "\mathrm{Mor}_{\mathcal{C}}(g, GD)"] & \mathrm{Mor}_{\mathcal{C}}(C, GD)
\end{tikzcd}$$

commutes since

$$\begin{aligned}
(\varphi(C, D) \circ \mathrm{Mor}_{\mathcal{D}}(Fg, D))(f) &= \varphi(C, D)(f \circ Fg) \\
&= G(f \circ Fg) \circ \Phi(C) \\
&= Gf \circ GFg \circ \Phi(C) \\
&= Gf \circ \Phi(C') \circ g \\
&= \mathrm{Mor}_{\mathcal{C}}(g, GD)(Gf \circ \Phi(C') \\
&= (\mathrm{Mor}_{\mathcal{C}}(g, GD) \circ \varphi(C', D))(f).
\end{aligned}$$

Now let $g : D \longrightarrow D'$ be given. The the diagram

$$\begin{tikzcd}
\mathrm{Mor}_{\mathcal{D}}(FC, D) \arrow[r, "\mathrm{Mor}_{\mathcal{D}}(FC, g)"] \arrow[d, "\varphi(C, D)"'] & \mathrm{Mor}_{\mathcal{D}}(FC, D') \arrow[d, "\varphi(C, D')"] \\
\mathrm{Mor}_{\mathcal{C}}(C, GD) \arrow[r, "\mathrm{Mor}_{\mathcal{C}}(C, Gg)"] & \mathrm{Mor}_{\mathcal{C}}(C, GD')
\end{tikzcd}$$

commutes since

3.2. **Monads or Triples.**

We now describe a structure based on an endofunctor which has turned out to be an important technical tool in studying toposes and related topics. It is an abstraction of the concept of adjoint and in a sense an abstraction of universal algebra (see the remarks in fine print at the end of 10.2.3 below).

3.2.1. **Definition.** Let $\mathcal{A}$ be a category and $R : \mathcal{A} \longrightarrow \mathcal{A}$ be an endofunctor. An *R-algebra* is a pair (A, a) where $a : RA \longrightarrow A$ is an arrow of $\mathcal{A}$. A *homomorphism between R-algebras* (A, a) and (B, b) is an arrow $f : A \longrightarrow B$ of $\mathcal{A}$ such that

$$\begin{tikzcd}
RA \arrow[r, "a"] \arrow[d, "Rf"'] & A \arrow[d, "f"] \\
RB \arrow[r, "b"'] & B
\end{tikzcd}$$

commutes. This construction gives a category $(R : \mathcal{A})$ of R-algebras.

3.2.2. **Definition.** A *monad* $\mathbb{T} = (T, \eta, \mu)$ on a category $\mathcal{A}$ consists of a functor $T : \mathcal{A} \longrightarrow \mathcal{A}$, together with two natural transformations $\eta : \mathrm{id} \longrightarrow T$ and $\mu : T^2 \longrightarrow T$ for which the following diagrams commute

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

Here, ηT and $T\eta$ are defined as in 1.7.56and 1.7.57.The transformation η is the *unit* of the monad and μ is the *multiplication*. The left diagram constitutes the (left and right) unary identities and the right one the associative identity. The reason for these names comes from the analogy between monads and monoids. This will be made clear in 3.2.4.Another widely used name for monad is 'triple'.
An adjoint pair gives rise to a monad on the domain of the left adjoint.

3.2.3. **Proposition.** *Let $U : \mathcal{B} \longrightarrow \mathcal{A}$ and $F : \mathcal{A} \longrightarrow \mathcal{B}$ be functors such that F is left adjoint to U with $\eta : id \longrightarrow UF$ and $\varepsilon : FU \longrightarrow id$ the unit and counit, respectively. Then $(UF, \eta, U\varepsilon F)$ is a monad on $\mathcal{A}$.*

Note that $U\varepsilon F : UFUF \longrightarrow UF$, as required for the multiplication of a monad with functor UF.
Conversely, every monad arises in that way out of some (generally many) adjoint pair. See Section 3.3 for two ways of constructing such adjoints.

3.2.4. **Example.** Let M be a monoid. The *representation monad* $T = (T, \eta, \mu)$ on the category of sets is given by letting $T(S) = M \times S$ for a set S. $\eta S : S \longrightarrow T(S) = M \times S$ takes an element $s \in S$ to the pair $(1, s)$. We define μS by $(\mu S)(m1, m2, s) = (m1m2, s)$ for $s \in S$, $m1, m2 \in M$. Thus the unit and multiplication of the monad arise directly from that of the monoid. The unitary and associativity equations can easily be shown to follow from the corresponding equations for the monoid.
The standard way of getting this monad from an adjoint pair is by using the underlying and free functors on M-sets (see 4.2.1???). If S and T are M-sets, then a function $f : S \longrightarrow T$ is said to be *M-equivariant* if $f(ms) = mf(s)$ for $m \in M$, $s \in S$. For a fixed monoid M, the M-sets and the M-equivariant functions form a category, called the *category of M-sets*.
The free M-set generated by the set S is the set $M \times S$ with action given by $m'(m, s) = (m'm, s)$. Using Theorem 9.3.5???, one can show immediately that this determines a functor left adjoint to the underlying set functor on the category of M-sets. The monad associated to this adjoint pair is the one described above.

3.2.5. **Example.** Let $T : \mathbf{Set} \longrightarrow \mathbf{Set}$ be the functor which takes a set A to the Kleene closure A^* and a function $f : A \longrightarrow B$ to the function $f^* : A^* \longrightarrow B^*$ defined in Section 2.5.7???. Let $\eta A : A \longrightarrow A$ take an element a to the one-element string (a), and let $\mu A : A^{**} \longrightarrow A^*$ be the *flattening operation* taking a string $(s_1, s_2, \ldots, s_n)$ of strings to the concatenated string $s_1 s_2 \ldots s_n$ in A^* obtained in effect by erasing inner brackets: thus $((a, b), (c, d, e), (), (a, a))$ goes to $(a, b)(c, d, e)()(a, a) = (a, b, c, d, e, a, a)$.
In particular, $\mu A((a, b)) = (a, b)$. Then $\eta : id \longrightarrow T$ and $\mu : T \circ T \longrightarrow T$ are natural transformations, and (T, η, μ) is a monad.

3.2.6. **Definition.** A comonad $G = (G, \varepsilon, \delta)$ in a category $\mathcal{A}$ is a monad in A^{op}. Thus G is an endofunctor of $\mathcal{A}$ and $\varepsilon : G \longrightarrow id$ and $\delta : G \longrightarrow G2$ are natural transformations such that

$$\begin{tikzcd} & G \arrow[dl, "="'] \arrow[d, "\delta"] \arrow[dr, "="] & \\ G & G^2 \arrow[l, "\varepsilon G"] \arrow[r, "G\varepsilon"'] & G \end{tikzcd} \qquad \begin{tikzcd} G \arrow[r, "\delta"] \arrow[d, "\delta"'] & G^2 \arrow[d, "\delta G"] \\ G^2 \arrow[r, "G\delta"'] & G^3 \end{tikzcd}$$

3.3. **Factorizations of a monad.**

3.3.1. **Remark.** Let $\mathbb{T} = (T, \eta, \mu)$ be a monad on $\mathcal{C}$. We describe here a construction which exhibits the monad as coming from an adjoint. This construction, which is due to Kleisli [1965], has proven to be quite useful in theoretical computer science.
We define a category $\mathcal{K} = \mathcal{K}(\mathbb{T})$ which has the same objects as $\mathcal{C}$. If A and B are objects of $\mathcal{C}$, then an arrow in $\mathcal{K}$ from A to B is an arrow $A \longrightarrow TB$ in $\mathcal{C}$. The composition of arrows is as follows. If $f : A \longrightarrow TB$ and $g : B \longrightarrow TC$ are arrows in $\mathcal{C}$, we let their composite in $\mathcal{K}$ be the following composite in $\mathcal{C}$:

$$A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu C} TC.$$

The identity of the object A is the arrow $\eta A : A \longrightarrow TA$. It can be shown that this defines a category. Moreover, there are functors $U : \mathcal{K}(\mathbb{T}) \longrightarrow \mathcal{C}$ and $F : \mathcal{C} \longrightarrow \mathcal{K}(\mathbb{T})$ defined by $UA = TA$ and $Uf = \mu B \circ \mu Tf$, where B is the codomain of f, and $FA = A$ and for $g : A \longrightarrow B$, $Fg = Tg \circ \eta A$. Then F is left adjoint to U and $T = U \circ F$.

Here is a second way, due to Eilenberg and Moore [1965] of factoring every monad as an adjoint pair of functors. In mathematics, this construction has been much more interesting than the Kleisli construction, but in computer science it has been quite the opposite.

3.3.2. **Definition.** Let $\mathbb{T} = (T, \eta, \mu)$ be a monad on $\mathcal{A}$. A *T-algebra* (A, a) is called a $\mathbb{T}$-algebra if the following two diagrams commute:

$$\begin{tikzcd} T^2A \arrow[r, "\mu A"] \arrow[d, "Ta"'] & TA \arrow[d, "a"] \\ TA \arrow[r, "a"] & A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\eta A"] \arrow[dr, "="'] & TA \arrow[d, "a"] \\ & A \end{tikzcd}$$

An *arrow (homomorphism) between $\mathbb{T}$-algebras* is the same as an arrow between the corresponding T-algebras. With these definitions, the $\mathbb{T}$-algebras form a category traditionally denoted $\mathcal{A}^{\mathbb{T}}$ and called the category of $\mathbb{T}$-algebras.
There is an obvious underlying functor $U : \mathcal{A}^{\mathbb{T}} \longrightarrow \mathcal{A}$ with $U(A, a) = A$ and $Uf = f$. This latter makes sense because an arrow of $\mathcal{A}^{\mathbb{T}}$ is an arrow of $\mathcal{A}$ with special properties. There is also a functor $F : \mathcal{A} \longrightarrow \mathcal{A}^{\mathbb{T}}$ given by $FA = (TA, \mu A)$ and $Ff = Tf$. Some details have to be checked, these are included in the following.

3.3.3. **Proposition.** *The function F above is a functor left adjoint to U. The monad associated to the adjoint pair FU is precisely T.*

> By a theorem of Linton's, every equationally defined category of one-sorted algebraic structures is in fact equivalent to the category of Eilenberg-Moore algebras for some monad in Set ([Barr and Wells, 1985], Theorem 5 of Section 4.3). In fact, the converse is true if infinitary operations are allowed (but then a hypothesis has to be added on the direct part of the theorem that there is only a set of operations of any given arity).

3.3.4. **Example.** Let (T, η, u) be the monad in **Set** defined in 10.1.6. An algebra for this monad is a monoid: specifically, if $ff : T(A) \longrightarrow A$ is an algebra, then the definition $ab = ff(a, b)$ makes A a monoid, and up to isomorphisms every monoid arises this way. Moreover, algebra homomorphisms are monoid homomorphisms, and every monoid homomorphism arises this way. The proof requires lengthy but not difficult verifications.

The Kleisli category of a monad $\mathbb{T} = (T, \eta, \mu)$ is equivalent to the full subcategory of free $\mathbb{T}$-algebras. Its definition makes it clear that the arrows are substitutions.

3.3.5. **Example.** As an example, consider the list monad of 10.1.6???. An arrow $f : A \longrightarrow B$ (here A and B are sets) of the Kleisli category is a set function $A \longrightarrow TB$, so that it associates a string of elements of B to each element of A. Suppose $A = \{a, b\}$ and $B = \{c, d, e\}$, and that $f(a) = cddc$ and $f(b) = ec$. Then $Tf : TA \longrightarrow TTB$ takes, for example, the string *abba* to $(cddc)(ec)(ec)(cddc)$, the result of substituting *cddc* for a and *ec* for b in *abba*. Then μ takes that string to *cddcececcddc*. It is instructive in this situation to think of μ as carrying out a computation. In this case the computation is carried out using the (only) monoid operation, since in fact the algebras for this monad are monoids. Thus one can think of the objects of the Kleisli category as computations. This is more compelling if one uses a monad arising from algebraic structures such as rings that abstract some of the properties of numerical addition and multiplication, then the objects of the free algebra are polynomial expressions and μ evaluates the polynomial.

3.4. **Algebraic Categories.**

3.5. **More facts about adjoint functors.**

3.5.1. **Theorem.** *Let $F : C \longrightarrow \mathcal{D}$ be a covariant functor that has a left adjoint. Then F preserves limits.*

Idea of proof. Let $D : I \longrightarrow \mathcal{C}$ be a diagram. Consider the isomorphisms

$$\begin{aligned}\mathrm{Mor}_{\mathcal{D}}(T, F \varprojlim D) &\cong \mathrm{Mor}_{\mathcal{C}}(GT, \varprojlim D) \cong \varprojlim \mathrm{Mor}(GT, D) \cong \\ &\cong \varprojlim \mathrm{Mor}(T, FD) \cong \mathrm{Mor}(T, \varprojlim FD)\end{aligned}$$

which implies $F \varprojlim D \cong \varprojlim FD$. □

3.5.2. **Definition.** Let $F : \mathcal{C} \longrightarrow \mathcal{D}$ be a functor. Let $D \in \mathcal{D}$. A *set* $\mathfrak{L}_D$ of objects in $\mathcal{C}$ is called a *solution set of* D with respect to F, if for each $C \in \mathcal{C}$ and for each morphism $D \longrightarrow F(C)$ there is an object $C' \in \mathfrak{L}_D$ and morphisms $f : \mathcal{C}' \longrightarrow C$ and $D \longrightarrow F(C')$ such that the triangle

$$\begin{tikzcd} D \arrow[r] \arrow[dr] & F(C') \arrow[d, "F(f)"] \\ & F(C) \end{tikzcd}$$

commutes.

3.5.3. **Theorem.** *Let $\mathcal{C}$ be a complete, locally small category and let $F : \mathcal{C} \longrightarrow \mathcal{D}$ be a (covariant) functor. Assume that $\mathcal{C}$ has a cogenerator.*
F has a left adjoint if and only if

- *F preserves limits and*
- *F has solutions sets.*

3.5.4. **Definition.** An object $C \in \mathcal{C}$ is called a cogenerator, if for any two morphisms $f, g : A \longrightarrow B$ with $f \neq g$ there is a morphism $h : B \longrightarrow C$ such that $h \circ f \neq h \circ g$.

3.5.5. **Theorem.** *Let $\mathcal{C}$ be a complete, locally small category and let $F : \mathcal{C} \longrightarrow \mathcal{D}$ be a (covariant) functor. Assume that $\mathcal{C}$ has a cogenerator.*
F has a left adjoint if and only if F preserves limits.

References

[Backus, 1981a] Backus, J.: *The algebra of functional programs: Function level reasoning, linear equations, and extended definitions.* In: Formalization of Programming Concepts, J. Diaz and I. Ramos, editors, volume 107 of Lecture Notes in Computer Science. Springer-Verlag **1981**.

[Backus, 1981b] Backus, J.: *Is computer science based on the wrong fundamental concept of program?* In: Algorithmic Languages, J. W. deBakker and J. C. van Vliet, editors. North-Holland **1981**.

[Barr and Wells, 1999] Barr, M. and Wells, C.: *Category Theory for Computing Science,* third edition. Les publications Centre de recherches mathématiques **1999**.

[Cockett, 1989] Cockett, J. R. B.: *On the decidability of objects in a locos.* In: Categories in Computer Science and Logic, J. W. Gray and A. Scedrov, editors, volume 92 of Contemporary Mathematics. American Mathematical Society **1989**.

[Jacobson, 1974] Jacobson, N.: *Basic Algebra.* I. W. H. Freeman **1974**.

[Pareigis, 1969] Pareigis, B.: *Kategorien und Funktoren.* Teubner Verlag **1969**.

[Pareigis, 2000] Pareigis, B.: *Lineare Algebra für Informatiker.* Springer Verlag **2000**.

[Pitt, 1986] Pitt, D., Abramsky, S., Poigné, A., and Rydeheard, D., editors. *Category Theory and Computer Programming*, volume 240 of Lecture Notes in Computer Science. Springer-Verlag **1986**.

[Wagner, 1986] Wagner, E. G.: *Algebraic theories, data types and control constructs.* Fundamenta Informatica, volume 9, pages 343-370 **1986**.

[Williams, 1982] Williams, J. H.: *Notes on the fp style of functional programming.* In: Functional Programming and its Applications, P. H. Darlington, J. and D. Turner, editors. Cambridge University Press **1982**.