Elements of Basic Category Theory

Alfio Martini

Bericht-Nr. 96-5

Elements of Basic Category Theory

e-mail:alfio@cs.tu-berlin.de

Abstract

Category theory provides an elegant and powerful means of expressing relationships across a wide area of mathematics. But further than this it has had a considerable impact on the conceptual basis both of mathematics and many parts of theoretical computer science. Important connections in computer science include the design of both functional and imperative programming languages, semantic models of programming languages, semantics of concurrency, specification and development of algorithms, type theory and polymorphism, specification languages, algebraic semantics, constructive logic and automata theory. The purpose of this text is to provide a soft stairway to this infectious and attractive field of mathematics. We provide here a careful and detailed explanation of "basic elements", or more precisely, from the elementary definitions to adjoint situations. The general approach used here is to provide a careful motivation for the majority of constructions as well as a detailed presentation of examples and proofs that both illustrate and provide a deeper understanding of the abstract constructions. A careful study of this text should provide the reader with the necessary fluency to accompany main parts of current research in theoretical computer science in connection to category theory, where in many instances, the latter plays now the role of being an essential tool.

Contents

Chapter 1

Introduction

1.1 About categories

Category Theory is a relatively new branch of mathematical enquiry. The subject was initiated in the early 1940's by Samuel Eilenberg and Saunders Mac lane [EM 45]. Its origins lie in algebraic topology, where constructions are developed that connect the domain of topology with that of algebra, specifically group theory.

The basic idea lies in the observation that several areas of mathematics involve the study of *objects* and *mappings* between this objects, e.g., sets and functions, vector spaces and linear transformations, groups and group homomorphisms. Such uniformity of structure can be explored by getting rid of internal details of sets, groups or vector spaces, and focusing only on the functions, homomorphisms or transformations and the means to combine them.

As a consequence, one of the primary perspectives offered by category theory is that the concept of *morphism* abstracted from that of *function* or *mapping*, may be used instead of the set membership relation as the basic building block for developing mathematical constructions, and expressing properties of mathematical entities. Instead of defining properties of a collection by reference to its members, i.e., *internal structure*, one can proceed by reference to its *external* relationships with other collections.

Besides, categorists have developed a symbolism that allows one quickly to visualize quite complicated facts by means of *diagrams*. Also, the concept of category is well balanced, which allows an economical and useful *duality*. Thus, in category theory, the "two for the price of one" principle holds: every concept is two concepts, and every result is two results.

1.2 Motivation and objectives of this work

Category theory provides an elegant and powerful means of expressing relationships across a wide area of mathematics. But further than this it has had a considerable impact on the conceptual basis both of mathematics and many parts of theoretical computer science. Important connections in computer science include the design of both functional and imperative programming languages, semantic models of programming languages, semantics of concurrency, specification and development of algorithms, type theory and polymorphism, specification languages, algebraic semantics, constructive logic and automata theory.

Especially, there are already a number of textbooks on category theory both for mathematicians and computer scientists, e.g., ([ADA 90], [ARB 75], [BAR 90], [POI 92]) just to name a few, and so one may find himself asking about the very purpose of writing another text on the subject. The reason lies on the fact that an appropriate understanding of the most important and useful categorical constructions may require both a strenuous effort and quite a long time from a student (or a beginner), so as to allow him, at least, to appreciate the current research papers applying category theory to computer science. In principle, the properties of categorical constructions are no different from familiar algebraic structures such as associativity, commutativity and distributivity. The real problem is just that categorical definitions and their properties are so much more complicated than computer scientists are used to. For example, the adjunction is one of the most important concepts of category theory, and one of the most relevant to applications in computer science. However, one of the simpler definitions of an adjunction requires six free variables (two categories, two functors, and two natural transformations) and four bound variables (two objects and two morphisms). Even apparently simpler definitions usually contain an alternation of universal and existential quantifiers. The complication and unfamiliarity of the formulae make it difficult to acquire and develop the kind of instinctive skill at pattern matching that makes algebraic calculations such an effective method of mathematical reasoning.

Therefore, in attempting to provide a reasonable introduction to the main concepts of category theory we have taken the approach of trying to move always from the particular to the general, following through the steps of the abstraction steps until the the abstract concepts emerges naturally. The starting points are elementary, and at the finish it would be quite appropriate for the reader to feel that (s)he had just arrived at the subject, rather than reached the end of the story. Moreover, the level of detail of the whole presentation might suggest that it could be used for self-study for undergraduate computer scientists and mathematicians as well. A careful study of this text should provide the reader with the necessary fluency to accompany main parts of current research in theoretical computer science in connection to category theory, where in many instances, the latter plays now the role of being an essential tool.

1.3 Topics

Chapter 2 begins with a brief account of the concept of function and function composition to motivate the introduction of the definition of a category. After we develop in detail a extensive number of examples of very well known mathematical structures in order to show that they really define a category.

In chapter 3 we introduce the concepts of diagrams, formal constructions to obtain new categories from old ones and basic types of morphisms. Especially, as a construction to obtain a new category from a old one, we have the duality construction. With this construction we are able to present a very simple yet powerful result, i.e., the *Principle of Duality*, which will be very valuable for us throughout this text.

In chapter 4 we present the basic constructions of category theory, i.e., terminal objects, products, equalizers and pullbacks, as well as their dual ones, i.e., initial objects, coproducts, coequalizers and pushouts. Later, we will see that all these constructions are just special cases of the more general concept of limit (dually colimit).

In chapter 5 we introduce the concept of limit and completeness. The concept of a limit (dually colimit) generalizes all the constructions presented in chapter 4 and the concept of completeness deals with the problem of how to verify if each diagram in a specific category has a limit (dually colimit).

In chapter 6 we introduce the concepts of functors and natural transformations. First we introduce the concept of functor and present in detail many examples of this construction. After we introduce the basic types of functors, preservation of properties by functors and the comma-category construction, although a special case of comma-category was already introduced in example 3.4.3.4. In the end, we introduce natural transformations and show in detail some examples and basic properties of this construction.

In chapter 7 we present the concepts of free construction (dually cofree construction) and adjoint functors. This chapter is considerably more difficult than the preceding ones. However, many of the constructions in the preceding chapters are examples of free and cofree constructions and thus, the emphasis will be on presentation of several examples rather than on giving formal proofs of theorems (of which there are many).

1.4 Note for the reader

- The end of a proof of each proposition or theorem is indicated by the symbol $\Box$.
- The bibliographic notes are done at the end of each chapter and compiled at the end of this work.
- Perhaps, a explanation for the title " Elements of Basic Category Theory " should be given. "Elements" warn the reader that not everything is here and "basic" is a remind that this work is concerned with the smallest subset of categorical notions which are necessary for a "natural" understanding of more advanced material.
- Although some constructions presented here are far from being trivial, the reader should not be discouraged if he finds that his reading rate is considerably slower that normal. What is true is that the concepts are very general and very abstract, and that, therefore, they may take some time getting used to. The reader's task in learning category theory is to steep himself in unfamiliar but essentially shallow generalities until they become so familiar that they can be used with almost no conscious effort. In other words, if you need "some" category theory, here it is; read it, absorb it, and forget it.

1.5 Acknowledgments

This work was developed within a German-Brazil cooperation on Information Technology, in the context of GRAPHIT project, and partially supported by a CNPq-grant number 200529/94-3 for Álfio Martini. The introduction of this work was initiated by Prof. Dr. Harmut Ehrig's Kloosterman Lecture "On the role of Category Theory in Computer Science" in Leiden, Holland, September, 1993, as well as the encouragement of Prof. Dr. Daltro Nunes concerning categorical methods in computer science. Especial thanks to Martin Korff, Jan Nicklisch, Martin Grosse-Rhode, Luis Lamb and Uwe Wolter for innumerous (and fruitful) discussions on this subject. A very special "thank you" goes again to Uwe Wolter, who carefully read this report, found numerous significant errors, and made very useful suggestions on how to correct them. Last but not least, I am "thankfull" to the very precise Felix Cornelius's internal manuscript "Adjunktionen", from which I have profited extensively while writing the last chapter of this report.

Chapter 2

Categories

This chapter consists, basically, in the introduction of the concept of a category as well as in the presentation of a number of examples. Firstly we make (a brief) revision of the concept of a function and of its main properties in order to motivate the introduction of the definition of a category. The rest of the chapter worries itself in presenting and verifying a great set of mathematical structures that can be understood as defining a category.

2.1 Motivation

Together with the concept of a set, the concept of a function is one of the most fundamentals for mathematics. Informally, a **function** (also called **mapping** or **map**) f from a set A (the *domain* of f) to a set B (the *codomain* of f) is a correspondence which assigns for each element $a \in A$ (*input*) a unique $b = f(a) \in B$ (the *image* of a under f, or the *output* produced by f according to the input a). We denote this situation by $f : A \to B$.

Before presenting a more precise mathematical definition for the concept of a function, we need the following

Definition 2.1.1 *A* **relation** *ρ from a set A to a set B is a subset from the cartesian product $A \times B$, i.e., $\rho \subseteq A \times B$. If $\langle a, b\rangle \in \rho$ then sometimes we use the notation $a\,\rho\,b$ to denote that a is ρ-related to b.*

Now, we can visualize a function as a (static) set-theoretic object through the following formal

Definition 2.1.2 *A* **function** *$f : A \to B$ is a special type of relation from A to B (i.e., $f \subseteq A \times B$), which satisfies:*

1. *For all $a \in A$, $a\,f\,b$ for some $b \in B$ (f is totally defined).*
2. *$a\,f\,b$ and $a\,f\,b' \Rightarrow b = b'$ (f is well-defined or single-valued).*

Remarks 2.1.3

1. If we relax condition 1 in definition 2.1.2, we obtain the concept of a **partial function**. In this case, with A and B sets, a partial function from A to B (denoted $f : A \rightharpoonup B$) is specified by determining a subset A' from A and a function which maps every element from A' to a unique element of B. Then we say that A is the domain, B is the codomain (as above) and A' is the **domain of definition**.

2. If $f : A \rightharpoonup B$ is a partial function, then we denote the domain of definition of f by $dom(f)$. If $a \in A$, but $a \notin dom(f)$, we say that "$f(a)$ is undefined".

3. The careful reader may have noted that an important case of a partial function $f : A \rightharpoonup B$ occurs when $dom(f) = A$. But this means that f is a function from A to B. When we find that it is adequate to emphasize this case, we call such f as a *total function* from A to B.

A more appropriate definition of function, which is more appropriate for our "categorical" purposes in this text is the following one:

Definition 2.1.4 *A function f is a mathematical entity with the following properties:*

1. *f has a* **domain** *and a* **codomain***, each of which must be a set.*

2. *For every element x of the domain, f has a* **value** *at x, which is an element of the codomain and is denoted $f(x)$.*

3. *The domain, the codomain, and the value $f(x)$ for each x in the domain are all determined by the function.*

4. *Conversely, the data consisting of the domain, the codomain, and the value $f(x)$ for each element x of the domain completely determine the function.*

Remark 2.1.5

1. The notation $f : A \to B$ is a succint way of saying that the function f has domain A and codomain B.

2. The above "specification" for a function is closer to the way a mathematician or a computer scientist thinks of a function that the previous one as a relation with the functional property. It is wrong to think that a function is an ordered pair in the same sense that it is wrong for a programmer writing in a high level language to think of the numbers he deals with as being expressed in binary notation. The "function as a relation" definition is an "implementation" of the specification for a function, and just has with program specifications, the expectation is that one normally works with specification, not the implementation in mind.

Definition 2.1.6 *The* **image** *of a function (also called its* **range***) is its set of values, that is, the range of a function $f : S \to T$ is the set $\{f(s) \in T | s \in S\}$. If $f : S \to T$, then we use $f(S)$ to denote the image of f.*

Definition 2.1.7 *A function $f : S \to T$ is* **surjective** *(or* **onto***) if its image is T, i.e., if $f(S) = T$.*

Definition 2.1.8 *A function $f : S \to T$ is injective (or* **one-to-one***) if whenever $f(s_1) = f(s_2)$ we have $s_1 = s_2$, for $s_1, s_2 \in S$.*

Definition 2.1.9 *A function $f : S \to T$ is called* **bijective** *if it is both injective and surjective.*

Given two functions $f : A \to B$ and $g : B \to C$, with the domain of one being the codomain of the other, we can obtain a new function by the rule "apply f and then g". For $a \in A$, the output $f(a)$ is an element of B, and hence an input to g. Applying g gives the element $g(f(x))$ of C. The passage from x to $g(f(x))$ establishes a function with domain A and codomain C. It is called the *composite* of f and g, denoted $g \circ f$, and symbolically defined by the rule $g \circ f(a) = g(f(a))$.

Now suppose we have three functions $f : A \to B$, $g : B \to C$, e $h : C \to D$ whose domains and codomains are so related that we can apply the three in succession to get a function from A to D. There are actually two ways to do this, since we can first form the composites $g \circ f : A \to C$ and $h \circ g : B \to D$. Then we follow either the rule "do f then $h \circ g$" giving the composite $(h \circ g) \circ f$, or the rule "do $g \circ f$ and then h", giving the composite $h \circ (g \circ f)$.

In fact, these two functions are the same. When we examine their outputs we find that

$$[h \circ (g \circ f)](a) = h(g \circ f(a)) = h(g(f(a))),$$

while

$$[(h \circ g) \circ f](a) = h \circ g(f(a)) = h(g(f(a))).$$

Thus the two functions have the same domain and codomain, and they give the same output for the same input. They each amount to the rule "do f, and then g, and then h". In other words, they are the same function, and we have established the following

Proposition 2.1.10 Associative law for functional composition. *For all $f : A \to B$, $g : B \to C$, and $h : C \to D$,*

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

Remark 2.1.11 This law allows to drop brackets and simply write $h \circ g \circ f$ without ambiguity.

What happens when we compose a function with an identity function (i.e., $id_A = \{\langle a, a\rangle | a \in A\}$)? Given $f : A \to B$ we can follow f by id_B. Computing outputs we find, for $a \in A$, that

CATEGORY	*OBJECTS*	*MORPHISMS*
Set	sets	total functions
FinSet	finite sets	total functions
Pfn	sets	partial functions
Rel	sets	binary relations
Mon	monoids	monoid homomorphisms
Poset	posets	monotonic functions
Grp	groups	group homomorphisms
Σ-Alg	Σ-Alg	Σ-homomorphisms
Cat$(SPEC)$	$SPEC$-algebras	$SPEC$-homomorphisms
Aut	finite automata	automata homomorphisms
Graph	directed graphs	graph morphisms

Table 2.1: Examples of categories

$$id_B \circ f(a) = id_B(f(a)) = f(a).$$

Similarly, we can precede f by id_A, in which case, for $a \in A$,

$$f \circ id_A(a) = f(id_A(a)) = f(a).$$

Since $id_B \circ f$ and f have the same domain and codomain, as do $f \circ id_A$ and A, we have established the following

Proposition 2.1.12 Identity law for functional composition. *For any function $f : A \to B$,*

$$id_B \circ f = f \text{ and } f \circ id_A = f.$$

2.2 Definition of category

Table 2.1 lists some categories by specifying their objects and morphisms. Now, before we present the formal definition of the concept of a category and verify that all these examples do are categories, let us first present some features they all have in common.

In each of this examples the objects are sets with, apart from the first four cases, some additional structure. The morphisms are all set functions which in each appropriate case satisfy conditions relating to this structure. It is not in fact vital that the reader be familiar with all of these examples. What is important is that she or he understands what they all have in common–what it is that makes each of them a category. The key lies, not in the particular nature of the objects and morphisms, but in the way the morphisms behave. In each case the following things occur:

- each morphism has associated with it two special objects, ist *domain* and ist *codomain*.
- there is an operation of *composition* that can be performed on certain pairs $\langle g, f \rangle$ of morphisms in the category (when domain of g = codomain of f) to obtain a new morphism $g \circ f$, *which is also in the category* (a composition of group homomorphism is a group homomorphism, a composition of monotonic functions is a monotonic function, etc.). This operation of composition always obey the associative law from proposition 2.1.10.
- each object has associated with it a special morphism in the category, the *identity morphism* on that object (the identity function on a group is a group homomorphism, on a poset is a monotonic function, etc.). Within the category, the identity morphism satisfy the identity law from proposition 2.1.12.

There are other features common to our list of examples. But as categories it is the two properties of associative composition and existence of identities we single out for particular attention in the following

Definition 2.2.1 *A category* $\mathbf{C}$ *comprises*

1. *a collection* $Ob(\mathbf{C})$ *of* **objects***;*
2. *a collection* $Mor(\mathbf{C})$ *of* **morphisms** *(also called* **arrows** *in the literature);*
3. *two operations* $dom, cod : Mor(\mathbf{C}) \to Ob(\mathbf{C})$ *assigning to each morphism* f *two objects, called respectively* **domain** *and* **codomain** *of* f*;*
4. *a composition operator* $\circ : Mor(\mathbf{C}) \times Mor(\mathbf{C}) \to Mor(\mathbf{C})$ *assigning to each pair of morphisms* $\langle f, g \rangle$ *with* $dom(g) = cod(f)$ *a composite morphism* $g \circ f : dom(f) \to cod(g)$*, such that the following* **associative law** *holds:*

   For any morphisms f, g, h *in* $Mor(\mathbf{C})$ *such that* $cod(f) = dom(g) \wedge cod(g) = dom(h)$*:*

$$h \circ (f \circ g) = (h \circ g) \circ h$$

5. *an operator* $id : Ob(\mathbf{C}) \to Mor(\mathbf{C})$*, assigning to each object* A*, an identity morphism* $id_A : A \to A$*, such that the following* **identity law** *holds:*

   For any morphism f *such that* $dom(f) = A \wedge cod(f) = B$*:*

$$id_B \circ f = f \wedge f \circ id_A = f$$

Remarks 2.2.2

1. Category theory is based on composition as a fundamental operation, in much the same way that classical set theory is based on the "element of" or membership relation.

2. Categories will be denoted by uppercase boldface letters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$ from the beginning of the alphabet.

3. We use letters $A, B, \ldots, Y, Z$ from the alphabet (with subscripts when appropriate) to denote objects and lowercase letters $a, b, c, \ldots, f, g, h, \ldots, y, z$ (occasionally with subscripts) to denote morphisms in any category.

4. If $dom(f) = A \wedge cod(f) = B$ we write $f : A \to B$ to denote that f is a morphism from A to B.

5. The collection of all morphisms with domain A and codomain B will be written as $\mathbf{C}(A, B)$ or as $Hom_{\mathbf{C}}(A, B)$.

6. Given a category $\mathbf{C}$, we may write "$\mathbf{C}$-object A" (resp. "$\mathbf{C}$-morphism $f : A \to B$") or "$A \in Ob(\mathbf{C})$" (resp. "$f \in \mathbf{C}(A, B)$") to denote that A is a object of $\mathbf{C}$ (resp. f is a morphism from A to B in $\mathbf{C}$).

2.3 Examples

We now verify some of the examples in Table 2.1 to see that they really satisfy definition 2.2.1.

Our first example, an important source of intuition throughout this work, is the category whose objects are sets and whose morphisms are functions.

Example 2.3.1 The category **Set** as sets as objects and total functions as morphisms. We now verify, in the same format as definition 2.2.1, that **Set** is really a category.

1. The composition of a total function $f : A \to B$ with another total function $g : B \to C$ is a total function $g \circ f : A \to C$, defined by $g \circ f(a) = g(f(a))$. Besides, the composition of total functions is associative, i.e.,

   for all total functions $f : A \to B$, $g : B \to C$ and $h : C \to D$, we have:

   $[h \circ (g \circ f)](a) = [(h \circ g) \circ f](a))$ (proposition 2.1.10)

2. For each set A, the identity function $id_A : A \to A$, defined by $id_A(a) = a$ is a total function with domain and codomain A. For any function $f : A \to B$, the identity functions on A and B satisfy the identity axiom:

   $id_B \circ f = f$ and $f \circ id_A = f$ (proposition 2.1.12)

Example 2.3.2 The category **FinSet** has finites sets as objects and total functions between finite sets as morphisms. The verification is analogous to **Set**.

Definition 2.3.3 *Given partial functions* $f : A \rightharpoonup B, g : B \rightharpoonup C$, *their* **composition** $g \circ f :$ $A \rightharpoonup C$ *is defined by*

$$
\begin{array}{rcl}
dom(g \circ f) & = & \{a \in A | a \in dom(f) \wedge f(a) \in dom(g)\} \\
(g \circ f)(a) & = & g(f(a)), \; \textit{for } a \in dom(g \circ f)
\end{array}
$$

Example 2.3.4 The category **Pfn** has sets as objects and partial functions as morphisms as follows:

1. The composition of two partial functions $f : A \multimap B, g : B \multimap C$ is the partial function $g \circ f : A \multimap B$ defined above.

   Besides, for any partial functions $f : A \multimap B, g : B \multimap C, h : C \multimap D$ we have that $h \circ (g \circ f) = (h \circ g) \circ f$. To see this, we must first show that $dom(h \circ (g \circ f)) = dom((h \circ g) \circ f)$. Now we have:

$$
\begin{array}{ll}
\multicolumn{2}{l}{dom(h \circ (g \circ f)) =} \\
= & \{a \in A | a \in dom(g \circ f) \wedge g(f(a)) \in dom(h)\} \\
= & \{a \in A | a \in \{a \in A | a \in dom(f) \wedge f(a) \in dom(g)\} \wedge g(f(a)) \in dom(h)\} \\
= & \{a \in A | a \in dom(f) \wedge f(a) \in dom(g) \wedge g(f(a)) \in dom(h)\} \\
= & \{a \in A | a \in dom(f) \wedge f(a) \in \{b \in B | b \in dom(g) \wedge g(b) \in dom(h)\}\} \\
= & \{a \in A | a \in dom(f) \wedge f(a) \in dom(h \circ g)\} \\
= & dom((h \circ g) \circ f))
\end{array}
$$

   Now, for any $a \in dom(h \circ (g \circ f))$, and noting that by the previous calculation, that all the following subexpressions are defined, we have:

$$
\begin{array}{rcl}
(h \circ (g \circ f))(a) & = & h(g \circ f(a)) \\
& = & h(g(f(a))) \\
& = & (h \circ g)(f(a)) \\
& = & ((h \circ g) \circ f)(a)
\end{array}
$$

2. Now, for any $f : A \multimap B$, it is easy to see that $dom(id_B \circ f) = dom(f), id_B \circ f = f, dom(f \circ id_A) = dom(f), f \circ id_A = f$. We leave the details to the reader.

Definition 2.3.5 *A* **semigroup** $(S, \odot)$ *is a set equiped with a binary operation* $\odot$ *which satisfies the associative law, i.e., for all* $x, y, z \in S$*:*

$$x \odot (y \odot z) = (x \odot y) \odot z.$$

Definition 2.3.6 *A* **monoid** $(M, \oplus, e_M)$ *is a set equipped with a binary operation* $\oplus$ *and a nullary operation* e_M *("selects the identity element") where:*

1. $(M, \oplus)$ *is a semigroup.*

2. $e_M \oplus x = x \oplus e_M = x$, *for all* $x \in M$.

Definition 2.3.7

1. *Λ is an* **alphabet** *if and only if Λ is a non-empty finite set of symbols.*

2. *For a given alphabet Λ and a natural number n, a sequence of symbols $a_1a_2\ldots a_n$ is a* **string** *(or a* **word***) of length n over the alphabet Λ if and only if, for $a_1a_2\ldots a_n$, $a_i \in \Sigma$.*

3. *For a given alphabet Λ and a string $x = a_1a_2\ldots a_n$ over Σ, $|x|$ denotes the length of x, i.e., $|a_1a_2\ldots a_n| = n$.*

4. *Given an alphabet Λ, the empty word, denoted by λ, is defined as being the unique string consisting of zero symbols.*

5. *Given an alphabet Λ and a non-negative integer $k \in \mathbb{N}$, we define*

$$\Lambda^k = \{x | x \textit{ is a string over } \Lambda \textit{ and } |x| = k\}$$

6. *Given an alphabet Λ, we define*

$$\Lambda^* = \cup_{k=0}^{\infty}\Lambda^k = \Lambda^0 \cup \Lambda^1 \cup \Lambda^2 \ldots$$

$$\Lambda^+ = \cup_{k=1}^{\infty}\Lambda^k = \Lambda^1 \cup \Lambda^2 \cup \Lambda^3 \ldots$$

Example 2.3.8 For any set A, $(A^*, \lambda, \diamond)$ is a monoid, where the identity element is the empty string λ and the (binary) operation $\diamond$ is the operation of concatenation of strings. $(A^*, \lambda, \diamond)$ is also called the **free monoid** generated by A. The concept of freeness is a general concept applied to many kinds of structures. It is treated in detail in chapter 7, in the context of free construction.

Definition 2.3.9 *Let $(M, \oplus, e_M)$ and $(N, \odot, e_N)$ be monoids. A function $f : M \to N$ is a* **monoid homomorphism** *if for all $x, y \in M$:*

1. *$f(x \oplus y) = f(x) \odot f(y)$ (f preserves $\oplus$)*

2. *$f(e_M) = e_N$ (f preserves the identity element)*

Example 2.3.10 The category **Mon** has monoids as objects and monoid homomorphisms as morphisms. Let $(M, \oplus, e_M)$, $(N, \odot, e_N)$ and $(O, \otimes, e_O)$ be monoids. Then we have:

1. The composition of two monoid homomorphisms $f : M \to N$ and $g : N \to O$ is also a monoid homomorphism $g \circ f$ from $(M, \oplus, e_M)$ to $(O, \otimes, e_O)$, as follows:

$$\begin{array}{rcll} g \circ f(x \oplus y) & = & g(f(x \oplus y)) & \text{(composition)} \\ & = & g(f(x) \odot f(y)) & (f \text{ is a monoid homomorphism}) \\ & = & g(f(x)) \otimes g(f(y)) & (g \text{ is a monoid homomorphism}) \\ & = & g \circ f(x) \otimes g \circ f(y) & \text{(composition)} \end{array}$$

$$
\begin{aligned}
g \circ f(e_M) &= g(f(e_M)) && \text{(composition)} \\
&= g(e_N) && \text{(f is a monoid homomorphism)} \\
&= e_O && \text{(g is a monoid homomorphism)}
\end{aligned}
$$

Besides, for any monoid homomorphisms $f : M \to N$, $g : N \to O$ and $h : O \to P$, we have $h \circ (g \circ f) = (h \circ g) \circ f$ since f, g and h are functions on sets.

2. For each monoid $(M, \oplus, e_M)$, the identity homomorphism $id_M : M \to M$ is defined by $id_M(x) = x$. Now for any monoid homomorphism $f : M \to N$, the identity monoid homomorphism on M and N satisfy the identity axiom, i.e., $id_N \circ f = f$ and $f \circ id_M = f$ as follows:

$$
\begin{aligned}
id_N \circ f(m) &= id_N(f(m)) && \text{(composition)} \\
&= f(m) && \text{(definition of id_N)}
\end{aligned}
$$

Similarly, it can be shown that $f \circ id_M = f$.

Definition 2.3.11 *Let ρ_1 be a relation from A to B, and ρ_2 a relation from B to C. The composition of ρ_1 and ρ_2, denoted $\rho_1 \circ \rho_2$, is a relation from A to C given by*

$\rho_1 \circ \rho_2 = \{\langle a, c\rangle \in A \times C | (\exists b \in B) : (\langle a, b\rangle \in \rho_1 \wedge \langle b, c\rangle \in \rho_2)\}$.

Example 2.3.12 The category **Rel** has sets as objects and relations as morphisms. According to definition 2.2.1 we have:

1. The composition of a relation of two relations $\rho_1 : A \times B$ and $\rho_2 : B \times C$ is another relation $\rho_1 \circ \rho_2 : A \times C$ in the sense of definition 2.3.11.

2. The composition of relations is associative, i.e., for any relations $\rho_1 : A \times B$, $\rho_2 : B \times C$ and $\rho_3 : C \times D$ we have, according to definition 2.3.11, that for all $\langle a, d\rangle \in (\rho_1 \circ \rho_2) \circ \rho_3$:

   $\langle a, d\rangle \in ((\rho_1 \circ \rho_2) \circ \rho_3)$ *iff*
   iff $(\exists c \in C)$ $(\langle a, c\rangle \in (\rho_1 \circ \rho_2) \wedge \langle c, d\rangle \in \rho_3)$
   iff $(\exists c \in C, \exists b \in B)$ $(\langle a, b\rangle \in \rho_1 \wedge \langle b, c\rangle \in \rho_2 \wedge \langle c, d\rangle \in \rho_3)$
   iff $(\exists b \in B, \exists c \in C)$ $(\langle a, b\rangle \in \rho_1 \wedge \langle b, c\rangle \in \rho_2 \wedge \langle c, d\rangle \in \rho_3)$
   iff $(\exists b \in B)$ $(\langle a, b\rangle \in \rho_1 \wedge \langle b, d\rangle \in (\rho_2 \circ \rho_3))$
   iff $\langle a, d\rangle \in (\rho_1 \circ (\rho_2 \circ \rho_3))$

   This shows that $(\rho_1 \circ \rho_2) \circ \rho_3 = \rho_1 \circ (\rho_2 \circ \rho_3)$.

3. For any set A, we have the identity relation $id_A : A \times A = \{\langle a, a\rangle | a \in A\}$. Now, for any relation $\rho : A \times B$, the identity relations on A and B clearly satisfy the identity axiom, i.e., $id_A \circ \rho = \rho = \rho \circ id_B$.

Definition 2.3.13 *Let P be a set and $\leq$ a relation on the elements of P. $(P, \leq_P)$ is called* **partially ordered set** *(poset for short), if for all $x, y, z \in P$, the following conditions are satisfied:*

1. $x \leq_P x$ *($\leq_P$ is reflexive)*
2. $x \leq_P y \wedge y \leq_P x \Rightarrow x = y$ *($\leq_P$ is antisymmetric)*
3. $x \leq_P y \wedge y \leq_P z \Rightarrow x \leq_P z$ *($\leq_P$ is transitive)*

Definition 2.3.14 *Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be posets. A* **monotonic** *(or ordering-preserving) function from $(P, \leq_P)$ to $(Q, \leq_Q)$ is a function $f : P \to Q$ such that $x \leq_P y \Rightarrow f(x) \leq_Q f(y)$.*

Example 2.3.15 The category **Poset** has posets as objects and monotonic functions as morphisms. Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be posets. Then we have:

1. The composition of two monotonic functions $f : P \to Q$ and $g : Q \to R$ is another monotonic function $g \circ f$ from P to R as follows:

   $$\begin{aligned} & x \leq_P y \Rightarrow & \\ & \Rightarrow f(x) \leq_Q f(y)) & (f \text{ is monotonic}) \\ & \Rightarrow g(f(x)) \leq_R g(f(y)) & (g \text{ is monotonic}) \\ & = g \circ f(x) \leq_R g \circ f(y) & (\text{composition}) \end{aligned}$$

   Besides, for any monotonic functions $f : P \to Q$, $q : P \to R$ and $h : R \to S$, we have $h \circ (g \circ f) = (h \circ g) \circ f$, since f, g and h are functions on sets.

2. For each poset $(P, \leq_P)$ the identity monotonic function $id_P : P \to P$ is defined as expected, i.e., such that $x \leq_P y \Rightarrow id_P(x) \leq_P id_P(y) \Rightarrow x \leq_P y$. Now for any $f : P \to Q$, the identity monotonic functions on P and Q satisfy the identity axiom, i.e., $id_Q \circ f = f$ and $f \circ id_P = f$ since id_P, id_Q and f are functions on sets.

Definition 2.3.16 *A group $(G, \odot, {}_G^{-1}, e_G)$ is a set equipped with a binary operation $\odot$, a unary operation ${}^{-1}$ ("takes the inverse element"), and a nullary operation e_M such that, for all $x, x^{-1}, \in G$:*

1. *$(G, \odot, e_G)$ is a monoid;*
2. *$x \odot x^{-1} = x^{-1} \odot x = e_G$ (inverse law).*

Definition 2.3.17 *Let $(G, \oplus, {}_G^{-1}, e_G)$ and $(H, \odot, {}_H^{-1}, e_H)$ be groups. A function $f : G \to H$ is a* **group homomorphism** *if for all $x, y \in G$:*

1. $f(x_G^{-1}) = f(x)_H^{-1}$ *(f preserves the inverse operation)*
2. $f(x \oplus y) = f(x) \odot f(y)$ *(f preserves $\oplus$)*
3. $f(e_G) = e_H$ *(f preserves the identity element)*

Example 2.3.18 The category **Grp** has groups as objects and group homomorphism as morphisms. The detailed verification that **Grp** is indeed a category is similar to the one done in example 2.3.10. The only missing detail is that here we must show that the composition of two group homomorphisms also preserves the inverse operations. This is shown as follows:

Let $f : G \to H$ and $g : H \to I$ be two group homomorphisms. Then we have:

$$\begin{aligned}
g \circ f(x_G^{-1}) &= g(f(x_G^{-1})) && \text{(composition)}\\
&= g(f(x)_H^{-1}) && (f \text{ is a group homomorphism})\\
&= g(f(x))_I^{-1} && (g \text{ is a group homomorphism})\\
&= (g \circ f(x))_I^{-1} && \text{(composition)}
\end{aligned}$$

Definition 2.3.19 (Many-sorted sets)

1. *Let S be a set (of sorts). A S-**sorted set** is a S-indexed family of sets $X = (X_s | s \in S)$, which is empty if X_s is empty for all $s \in S$. The empty S-sorted set is written $\emptyset$.*
2. *Let $X = (X_s | s \in S)$,$Y = (Y_s | s \in S)$ be S-sorted sets. Then we have:*
   - $X \cup Y = (X_s \cup Y_s | s \in S)$.
   - $X \subseteq Y$ iff $X_s \subseteq Y_s$ *for all* $s \in S$.
   - $X = Y$ iff $X \subseteq Y$ *and* $Y \subseteq X$
3. *An S-**sorted function** $f : X \to Y$ is an S-indexed family of functions $f = (f_s : X_s \to Y_s | s \in S)$.*
4. *If $f : X \to Y$ and $g : Y \to Z$ are S-sorted functions, then their **composition** $g \circ f : X \to Z$ is the S-sorted function defined by $(g \circ f)_s(x) = g_s(f_s(x))$ for $s \in S, x \in X_s$.*

Semigroups, monoids and groups are sets equipped with a pre-defined set of operations. Technically they are called "algebras" in the mathematical literature. By asking the natural question about the existence of a notation (syntax) to speak about or to classify algebras we finally get to the point of introducing the next

Definition 2.3.20 *A **signature** $\Sigma = (S, OP)$ consists of a set S, the set of **sorts**, and a $S^* \times S$-indexed set $OP = (OP_{w,s} | w \in S^*, s \in S)$, called the set of **constants and operations symbols**, where $(OP_{w,s} | w \in S^*, s \in S)$ is the set of operations symbols with argument sorts $w \in S^*$ and range sort $s \in S$. When $w = \lambda$, we write $K = (OP_{\lambda,s} | s \in S)$ to denote the set of constant symbols of sort s for all $s \in S$.*

Example 2.3.21 A signature for strings (in the sense of 2.3.20) can be given as follows:

$\underline{\underline{\text{string}=}}$

sorts: *alphabet*
string

opns: $k_1, \ldots, k_n :\to alphabet$
$empty :\to string$
$make : alphabet \to string$
$concat : string\ string \to string$
$ladd : alphabet\ string \to string$
$radd : string\ alphabet \to string$

Definition 2.3.22 *An* **algebra** $A = (S_A, OP_A)$ *of a signature* $\Sigma = (S, OP)$, *also called* Σ-**algebra** *is given by two families* $S_A = (A_s | s \in S)$ *and* $OP_A = (N_A | N \in OP)$ *where*

1. $(A_s | s \in S)$ *is a family of S-indexed sets, called* **base sets** *of* A.

2. N_A *are elements* $N_A \in A_s$ *for all constants symbols* $N \in K_s, s \in S$, *called* **constants** *of* A.

3. $N_A : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ *are functions for all operation symbols* $N \in OP_{s_1 \ldots s_n, s}$ *(i.e.,* $N : s_1 \ldots s_n \to s$*) and* $s_1 \ldots s_n \in S^+$, $s \in S$, *called* **operations** *of* A, *where "$\times$" denotes the cartesian products of sets.*

Remark 2.3.23 If the signature $\Sigma = (S, OP)$ is given as a list $s_1, \ldots, s_n$ of sorts and a list $N_1, \ldots, N_k$ of constant and operations symbols, a Σ-algebra is represented as a list $A = (A_{s_1}, \ldots, A_{s_n}, N_{1_A}, \ldots, N_{k_A})$, where the base sets and operations are listed in corresponding order. Note that given a signature $\Omega = (s, \lambda :\to s, + : s \times s \to s)$, then a monoid $(M, e_M, \oplus)$ is just Ω-algebra, where M is a set for the sort s and $e_M, \oplus$ are the respective operations corresponding to the operation symbols λ and $+$.

Example 2.3.24 The algebra $STRING = (A, A^*, a_1, \ldots, a_n, \lambda, make, concat, ladd, radd)$ with

$$
\begin{aligned}
& make : A \to A^* \\
& \quad make(a) = a \\
& concat : A^* \times A^* \to A^* \\
& \quad concat(u, v) = uv \\
& ladd : A \times A^* \to A^* \\
& \quad ladd(a, u) = au \\
& radd : A^* \times A^* \to A \\
& \quad radd(u, a) = ua
\end{aligned}
$$

is a $\underline{\underline{string}}$-algebra in the sense of definition 2.3.22.

Definition 2.3.25 *Let* $\Sigma = (S, OP)$ *be a signature and* $X = (X_s | s \in S)$ *a S-indexed set of* **variables**. *We assume that these sets* X_s *are pairwise disjoint and also disjoint with* OP. *The set* $X = (X_s \S \in S)$ *is called* **set of variables** *w.r.t* Σ.

Definition 2.3.26 *The set* $T_{OP,s}(X)$ *of* **terms of sort** s *is defined for all* $s \in S$ *as follows:*

1. $X_s \cup K_s \subseteq T_{OP,s}(X)$ (**basic terms**), *where* K_s *is the set of constant symbols of sort* s.
2. $N(t_1, \ldots, t_n) \in T_{OP,s}(X)$ (**composite terms**), *for all operations symbols* $N \in OP$ *with* $N : s_1 \ldots s_n \to s$ *and all terms* $t_1 \in T_{OP,s_1}(X), \ldots, t_n \in T_{OP,s_n}(X)$.
3. *There are no further terms of sort* s *in* $T_{OP,s}(X)$.
4. *The set* $T_{OP,s}$ *of* **terms without variables of sort s**, *also called set of* **ground terms** *of sort* s, *is defined for the empty set* $X = \emptyset$ *of variables by* $T_{OP,s} = T_{OP,s}(\emptyset)$.
5. *The set of terms* $T_{OP}(X)$ *and the set of terms without variables* T_{OP} *are defined by* $T_{OP}(X) = (T_{OP,s}(X) | s \in S)$ *and* $T_{OP} = (T_{OP,s} | s \in S)$.

Definition 2.3.27

1. *Let* T_{OP} *be set of terms of a signature* $\Sigma = (S, OP)$ *and* A *a* Σ*-algebra. The* **evaluation** $eval = (eval_s : T_{OP_s} \to A_s | s \in S)$ *is recursively defined for all* $s \in S$ *by*

   a. $eval_s(N) = N_A$ *for all constant symbols* $N \in K_s$;

   b. $eval_s(N(t_1, \ldots, t_n)) = N_A(eval_{s1}(t_1), \ldots, eval_{sn}(t_n))$ *for all* $N(t_1, \ldots, t_n) \in T_{OP,s}, t_1 \in T_{OP,s1}, \ldots, t_n \in T_{OP,sn}$.
2. *Given a set of variables* X *for* $\Sigma = (S, OP)$ *and an* **assignment** $ass = (ass_s : X_s \to A_s | s \in S)$, *the* **extended assignment**, *or simply* **extension** $ass^\sharp : T_{OP}(X) \to A$ *is recursively defined by*

   a. $ass^\sharp_s(x) = ass_s(x)$ *for all variables* $x \in X_s$; $ass^\sharp_s(N) = N_A$ *for all constants symbols* $N \in K_s$;

   b. $ass^\sharp_s(N(t_1, \ldots, t_n)) = N_A(ass^\sharp_{s1}(t_1), \ldots, ass^\sharp_{sn}(t_n))$ *for all* $N(t_1, \ldots, t_n) \in T_{OP,s}(X)$, $t_1 \in T_{OP,s1}, \ldots, t_n \in T_{OP,sn}$.

Remarks 2.3.28

For $X = \emptyset$ there is exactly one assignment ass which is the "empty assignment" $\emptyset$, and we have $ass^\sharp = eval$.

Definition 2.3.29 *Given a signature* $\Sigma = (S, OP)$ *and variables* X *with respect to* Σ.

1. *A triple* $e = (X, L, R)$ *with* $L, R \in T_{OP,s}(X)$ *for some* $s \in S$ *is called an* **equation** *of sort* s *w.r.t* Σ.
2. *The equation* $e = (X, L, R)$ *is called* **valid** *in a* Σ*-algebra* A *if for all assignments* $ass : X \to A$ *we have:*

   $$ass^\sharp_s(L) = ass^\sharp_s(R)$$

   where $ass^\sharp$ *is the extended assignment of* ass. *If* e *is valid in* A, *we also say that* A **satisfies** e.

3. **Ground equations** *are equations* $e = (X, L, R)$ *with* $X = \emptyset$*. In this case,* L *and* R *are ground terms.*

Definition 2.3.30

- *A* **specification** $SPEC = (S, OP, E)$ *consists of a signature* $\Sigma = (S, OP)$ *and a set* E *of equations* e *w.r.t* Σ.
- *An* **algebra** A *of the specification* $SPEC$*, short* $SPEC$*-algebra, is an algebra* A *of the signature* Σ *which satisfies all equations in* E.

Example 2.3.31 In this example, we give a specification *string* for the abstract data type string such that the algebra $STRING$ becomes a *string*-algebra in the sense of 2.3.30.2.

string=

sorts:	$alphabet$
	$string$
opns:	$k_1, \ldots, k_n :\to alphabet$
	$empty :\to string$
	$make : alphabet \to string$
	$concat : string\ string \to string$
	$ladd : alphabet\ string \to string$
	$radd : string\ alphabet \to string$
eqns:	$a \in alphabet; s, s_1, s_2, s_3 \in string$
	$concat(s, empty) = s$
	$concat(empty, s) = s$
	$concat(concat(s_1, s_2), s_2) =$
	$concat(s_1, concat(s_2, s_3))$
	$ladd(a, s) = concat(make(a), s)$
	$radd(s, a) = concat(s, make(a))$

Definition 2.3.32 *Let a* A *and* B *be algebras of the same signature* $\Sigma = (S, OP)$ *or specification* $SPEC = (S, OP, E)$.

- *A* **homomorphism** $f : A \to B$*, also called* Σ*- or* $SPEC$*- homomorphism, is a family of functions* $f_s : A_s \to B_s$*, for* $s \in S$*, such that for each constant symbol* $N :\to s$ *in* OP *and* $s \in S$*, we have that*

$$f_s(N_A) = N_B,$$

and for each operation symbol $N : s_1 \ldots s_n \to s$ *in* OP *and all* $a_i \in A_{s_i}$*, for* $i = 1, \ldots, n$

$$f_s(N_A(a_1, \ldots, a_n)) = N_B(f_{s_1}(a_1), \ldots, f_{s_n}(a_n)).$$

- *A homomorphism* $f : A \to B$ *is called an* **isomorphism** *if all functions* $f_s : A_s \to B_s$*, for* $s \in S$*, are bijective.*

Example 2.3.33 For each signature $\Sigma = (S, OP)$ (resp. each specification $SPEC = (S, OP, E)$ the category Σ-**Alg** (resp. **Cat**$(SPEC)$ has Σ- algebras (resp. $SPEC$-algebras) as objects, and Σ-homomorphisms (resp. $SPEC$-homomorphisms) as morphisms as follows:

1. The composition of the Σ-homomorphisms (resp. $SPEC$-homomorphisms) $f : A \to B$ and $g : B \to C$, denoted by $g \circ f : A \to C$ and defined by $g_s \circ f_s(a) = g_s(f_s(a))$ for $a \in A_s$, $s \in S$, it is also a Σ-homomorphism (resp. $SPEC$-homomorphism) as follows:

   For constants symbols $N \in OP_s$ and $s \in S$ we have

   $$\begin{aligned} & g_s \circ f_s(N_A) = \\ & = g_s(f_s(N_A)) && \text{(composition)} \\ & = g_s(N_B) && \text{(f is a Σ- (or $SPEC$-) homomorphism)} \\ & = N_C && \text{(g is a Σ- (or $SPEC$-) homomorphism)} \end{aligned}$$

   For $N : s_1 \dots s_N \to s$ and $a_i \in A_{s_i}$, $i = 1, \dots, n$ we have:

   $$\begin{aligned} & g_s \circ f_s(N_A(a_1, \dots, a_n)) = \\ & = g_s(f_s(N_A(a_1, \dots, a_n))) && \text{(composition)} \\ & = g_s(N_B(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) && \text{(f is a Σ- (or $SPEC$-) homomorphism)} \\ & = N_C(g_{s_1}(f_{s_1}(a_1)), \dots, g_{s_n}(f_{s_n}(a_n))) && \text{(g is a Σ- (or $SPEC$-) homomorphism)} \\ & = N_C(g_{s_1} \circ f_{s_1}(a_1), \dots, g_{s_n} \circ f_{s_n}(a_n)) && \text{(composition)} \end{aligned}$$

   Moreover, for any Σ-homomorphisms (or $SPEC$-homomorphisms) $f : A \to B$, $g : B \to C$ and $h : C \to D$ we have that $h \circ (g \circ f) = (h \circ g) \circ f$, since f, g and h are functions on sets.

2. For each Σ-algebra A (resp. $SPEC$-algebra A), the identity Σ- (or $SPEC$) homomorphism id_A, defined by $id_{A,s}(a) = a$, for $s \in S$ and $a \in A_s$, is clearly an identity Σ or $SPEC$-homomorphism as follows:

   For each constant symbol $N :\to s$ in OP and $s \in S$ we have:

   $$id_{A,s}(N_A) = N_A \qquad \text{(by definition)}$$

   For each operation symbol $N : s_1 \dots s_n \to s$ in OP and all $a_i \in A_{s_i}$, for $i = 1, \dots, n$.

   $$\begin{aligned} & id_{A,s}(N_A(a_1, \dots, a_n)) = \\ & = N_A(a_1, \dots, a_n) && \text{(definition of $id_{A,s}$)} \\ & = N_A(id_{A,s}(a_1), \dots, id_{A,s}(a_n)) && \text{(definition of $id_{A,s}$)} \end{aligned}$$

   Moreover, for any identity $\Sigma-$ or $SPEC$- homomorphism $f : A \to B$, the identity Σ- or $SPEC$- homomorphisms id_A and id_B satisfy the identity axiom since the families $(f_s | s \in S)$, $(id_{A_s} | s \in S)$, $(id_{B_s} | s \in S)$ are functions on sets.

Definition 2.3.34 *A* **directed graph** $G = (E, V, s, t)$ *consists of a set E of edges, a set V of vertices, and two maps, $s : E \to V$ (called* **source map***) and $t : E \to V$ (called* **target map***).*

Remarks 2.3.35

1. Note that a graph is just an algebra for a signature declaring two sort symbols, say $arc, node$ and two operation symbols,say $source, target : a \to n$.

2. If $e \in E$, $u, v \in V$, $s(e) = u$, $t(e) = v$, then we write $e : u \to v$ to denote that e is an edge of G.

Definition 2.3.36 *A direct graph is called* **discrete** *if it has no edges, i.e., a discrete graph is essentially a set.*

Definition 2.3.37 *A* **graph morphism** *(or* **graph homomorphism***) from a graph $G = (E_G, V_G, s_G, t_G)$ to a graph $H = (E_H, V_H, s_H, t_H)$, denoted $f = (f_E, f_V) : G \to H$, is a pair of functions $f_V : V_G \to V_H$ and $f_E : E_G \to E_H$, with the property that if $u : m \to n$ is an edge of G, then $f_E(u) : f_V(m) \to f_V(n)$ is an edge of H.*

Definition 2.3.38 *For any graph morphism $f : G \to H$ we have:*

1. *f is* **injective** *if both f_E and f_V are injective.*

2. *f is* **surjective** *if both f_E and f_V are surjective.*

3. *f is* **bijective** *if both f_E and f_V are bijective.*

Example 2.3.39 The category **Graph** has graphs as objects and graph morphisms as morphisms as follows:

1. The composition of two graph morphisms $f : G \to H$ and $g : H \to I$, defined by $g \circ f = (g_E \circ f_E, g_V \circ f_V)$ from G to I, is also a graph morphism as follows:

   Let $u : m \to n$ be an edge of G. Then we have:

   $$\begin{array}{lr} u : m \to n \text{ in } G \Rightarrow & \\ \Rightarrow f_E(u) : f_V(m) \to f_V(n) \text{ in } H & (f \text{ is a graph morphisms}) \\ \Rightarrow g_E(f_E(u)) : g_V(f_V(m)) \to g_V(f_V(n)) \text{ in } I & (g \text{ is a graph morphism}) \\ \Rightarrow g_E \circ f_E(u) : g_V \circ f_V(m) \to g_V \circ f_V(n) \text{ in } I & (\text{composition}) \end{array}$$

   Moreover, for any graph morphisms $f : G \to H$, $g : H \to I$ and $h : I \to J$, we have $h \circ (g \circ f) = (h \circ g) \circ f$, since f, g and h are functions on sets.

2. For each graph $G = (E_G, V_G, s_G, t_G)$, the identity graph morphism $id_G = (id_E, id_V) : G \to H$ is defined by: if $u : m \to n$ is an edge in G then $id_E(u) : id_V(m) \to id_V(n)$ is (clearly) an edge in G. Now, we trivially have that $id_H \circ f = f$ and $f \circ id_G = f$.

Definition 2.3.40 *A* **finite automata** *is a 6-tuple $A = (I, O, S, s_0, \delta, \lambda)$ where:*

- *I is a finite set of input symbols;*

- *O is a finite set of output symbols;*

- *S is a finite set of states;*
- *s_0 is the initial state;*
- *$\delta : I \times S \to S$ is the transition function.*
- *$\lambda : I \times S \to O$ is the output function.*

Definition 2.3.41 *Let A and A' be finite automata. An* **automata homomorphism** *f from A to A', written $f = (f_I, f_O, f_S) : A \to A'$, is a triple of functions $f_I : I \to I'$, $f_O : O \to O'$, $f_S : S \to S'$ such that the following conditions are satisfied.*

- $f_S(s_0) = s'_0$ *(preservation of the initial state)*
- $f_S(\delta(i,s)) = \delta'(f_I(i), f_S(s))$ *(preservation of transitions)*
- $f_O(\lambda(i,s)) = \lambda'(f_I(i), f_S(s))$ *(preservation of outputs)*

Proposition 2.3.42 *Let $f : A_1 \to A_2$ and $g : A_2 \to A_2$ be automata homomorphisms. Then, the composition $g \circ f = (g_I \circ f_I, g_O \circ f_O, g_S \circ f_S) : A_1 \to A_2$ is (again) an automata homomorphism.*

Proof: For all $(i,s) \in I_1 \times S_1$, we have:

1. Preservation of the initial state

$g_S \circ f_S(s_{0_1}) =$
$= g_S(f_S(s_{0_1}))$ (composition)
$= g_S(s_{0_2})$ (f is an automata homomorphism)
$= s_{O_3}$ (g is an automata homomorphism)

2. Preservation of transitions

$\delta_3(g_I \circ f_1(i), g_S \circ f_S(s)) =$
$= \delta_3(g_I(f_I(I)), g_S(f_S(s)))$ (composition)
$= g_S(\delta_2(f_I(i), f_S(s)))$ (g is an automata homomorphism)
$= g_S(f_S(\delta_1(i,s)))$ (f is an automata homomorphism)
$= g_S \circ f_S(\delta(i,s))$ (composition)

3. Preservation of outputs

$\lambda_3((g_I \circ f_1(i), g_S \circ f_S(s)) =$
$= \lambda_3(g_I(f_I(i)), g_S(f_S(s)))$ (composition)
$= g_O(\lambda_2(f_I(i), f_S(s)))$ (g is an automata homomorphism)
$= g_O(f_O(\lambda_1(i,s)))$ (f is an automata homomorphism)
$= g_O \circ f_O(\lambda_1(i,s))$ (composition)

$\square$

Example 2.3.43 The category **Aut** has finite automata as objects and automata homomorphisms as morphisms as follows:

1. The composition of two automata homomorphisms $f : A_1 \to A_2$ and $g : A_2 \to A_3$ is also an automata homomorphism $g \circ f : A_1 \to A_3$ (proposition 2.3.42)

   Moreover, for any automata homomorphisms $f : A_1 \to A_2$, $g : A_2 \to A_3$ and $h : A_3 \to A_4$ we have that $h \circ (g \circ f) = (h \circ g) \circ f$. This follows immediately by associativity of function composition, since the components of each automata homomorphism are functions on the sets I, O and S.

2. For any automata $A = (I, O, S, s_0, \delta, \lambda)$ we have the identity automata homomorphism $id_A = (id_I, id_O, id_S)$, where id_I, id_O, id_S are the respective identities on the sets I, O and S. It is immediate that this indeed defines an automata homomorphism. For example, we have:

   $\lambda(id_I(i), id_S(s)) =$
   $= \lambda(i, s)$ (definition of id_I and id_S)
   $= id_O(\lambda(i, s))$ (definition of id_O)

   Now for any automata homomorphism $f : A_1 \to A_2$ we have that for all $s \in S_1$

   $f \circ id_{S_1}(s) =$
   $= f(id_{S_1}(s))$ (composition)
   $= f(s)$

   The other equalities (for the alphabets I and O) as well $id_{A_2} \circ f = f = f \circ id_{A_1}$ for each $f : A_1 \to A_2$ can be trivially verified.

The intuitive notion of "category" suggested by the examples we have presented so far is to consider objects as a class of sets with "structure", and the morphisms as functions "associated" or "acceptable" in relation to that structure. However, this consideration is too restrictive, since nothing was stated in the (formal) definition of a category that can restrict the morphisms to be structure-preserving functions. The examples in the sequel illustrate this observation.

Example 2.3.44 A monoid $(M, \oplus, e_M)$ can be seen as a one-object category $\mathbf{C}(M)$ in the following way:

1. The composition of two morphisms $x, y \in \mathbf{C}(M)(M, M)$ is a morphism $y \circ x \in \mathbf{C}(M)(M, M)$ defined by $y \circ x = y \oplus x$. Besides, the composition of morphisms in this category is clearly associative, since for all x, y, $z \in \mathbf{C}(M)(M, M)$ we have:

$$\begin{aligned} z \circ (y \circ x) &= z \circ (y \oplus x) && \text{(definition of composition)} \\ &= z \oplus (y \oplus x) && \text{(definition of composition)} \\ &= (z \oplus y) \oplus x && \text{(associativity of } \oplus\text{)} \\ &= (z \circ y) \oplus x && \text{(definition of composition)} \\ &= (z \circ y) \circ x && \text{(definition of composition)} \end{aligned}$$

2. For each $x \in \mathbf{C}(M)(M, M)$ there exists a (unique) identity morphism $e_M : M \to M$ (the identity element of the monoid) which trivially satisfies the identity axiom (see definition 2.3.6).

Example 2.3.45 A poset $(P, \leq)$ can be seen as a category $\mathbf{C}(P)$ in the following way:

For all $x, y, z \in P$

1. An object in $\mathbf{C}(P)$ is a element of P.
2. The morphisms of $\mathbf{C}(P)$ are defined in the following way:

$$\mathbf{C}(P)(x, y) = \begin{cases} \{x \to y\} & \text{if } x \leq_P y \\ \emptyset & otherwise \end{cases}$$

3. The composition of two morphisms $x \to y$ and $y \to z$ is defined by:

   $y \to z \circ x \to y = x \leq_P y \wedge y \leq_P z$.

   The composition of morphisms in $(P, \leq)$ is associative, which can be seen as follows:

   For any $x \to y$, $y \to z$ and $z \to w$ we have:

$$\begin{aligned}
&z \to w \circ (y \to z \circ x \to y) = \\
&= z \to w \circ (x \leq_P y \wedge y \leq_P z) && \text{(definition of composition)} \\
&= (x \leq_P y \wedge y \leq_P z) \wedge z \leq_P w && \text{(definition of composition)} \\
&= x \leq_P z \wedge z \leq_P w && \text{(transitivity of } \leq_P) \\
&= x \leq_P w && \text{(transitivity of } \leq_P) \\
&= x \to w && \text{(definition of } x \to w)
\end{aligned}$$

   and

$$\begin{aligned}
&(z \to w \circ y \to z) \circ x \to y = \\
&= (y \leq_P z \wedge z \leq_P w) \circ x \to y && \text{(definition of composition)} \\
&= x \leq_P y \wedge (y \leq_P z \wedge z \leq_P w) && \text{(definition of composition)} \\
&= x \leq_P y \wedge y \leq_P w && \text{(transitivity of } \leq_P) \\
&= x \leq_P w && \text{(transitivity of } \leq_P) \\
&= x \to w && \text{(definition of } x \to w)
\end{aligned}$$

4. For each element $x \in P$, there exists a (unique) morphism $x \to x$ (since $\leq_P$ is reflexive), such that for any morphism $x \to y$ the identity axiom is satisfied, i.e.,

$$\begin{aligned}
x \to y \circ x \to x &= x \leq_P x \wedge x \leq_P y && \text{(definition of composition)} \\
&= x \leq_P y && \text{(transitivity of } \leq_P) \\
&= x \to y && \text{(definition of } x \to y)
\end{aligned}$$

   Similarly, it can be shown that $y \to y \circ x \to y = x \to y$.

Example 2.3.46 Category **0**: This category has no objects and no morphisms. The associativity and identity axioms are vacuously satisfied. To see this suppose that, for example, the identity axiom is not satisfied. Then we must able to find at least one morphism $f : A \to B$ such that $f \circ id_A \neq f$ and $id_B \circ f \neq f$. However, by definition, there is no object and no morphism in **0** and therefore we get a contradiction. Hence the axiom should hold.

Example 2.3.47 Category **1**: This category has one object and one morphism. Having said that, we find that its structure is completely determined. To see this, suppose we call the object A and the morphism f. Then we must put $dom(f) = A$ and $cod(f) = A$, as A is the only available object. The associativity and identity axioms are verified as follows:

- Since $f : A \to A$ is the only available morphism we put $f \circ f = f \in \mathbf{1}(A, A)$. Hence, the associativity axiom trivially holds, as we undoubtedly have $f \circ (f \circ f) = (f \circ f) \circ f$.
- Since f is the only available morphism, we put $id_A = f$. Hence, we have:

$$id_A \circ f = f \circ f = f = f \circ f = f \circ id_A$$

Example 2.3.48 Category **2**: This category has two objects, two identity morphisms and a morphism from one object to the other. Again, it does not matter what the objects and morphisms represent, but to make easier to talk about them, we might call the objects A and B and the non-identity morphism f. There is only one way to define composition in this category, i.e.,

$$\begin{aligned}
&id_A \circ id_A = id_A \\
&id_B \circ id_B = id_B \\
&id_B \circ f = f \\
&f \circ id_A = f
\end{aligned}$$

The last two equations show that this category satisfies the identity axiom, while for the associativity axiom we have trivially

$$id_B \circ (f \circ id_A) = id_B \circ f = f = f \circ id_B = (id_B \circ f) \circ id_A$$

Example 2.3.49 Category **3**: This category has three objects (call them A, B and C), three identity morphisms, and three other morphisms, say $f : A \to B$, $g : B \to C$ and $h : A \to C$. Again, the composition can only defined in only way, i.e., besides the compositions presented in the above example, we have also the following ones:

$$\begin{aligned}
&id_C \circ id_C = id_C \\
&id_C \circ g = g \\
&g \circ id_B = g \\
&id_C \circ h = h \\
&h \circ id_A = h \\
&g \circ f = h
\end{aligned}$$

The last equality holds because both $g \circ f$ and h are morphisms from A to C. However, by definition, there is only one morphism from A to C, namely h, and so we must have $g \circ f = h$.

The associativity axiom can be verified similarly as was done in the category **1**.

Definition 2.3.50 A category **C** is said to be **small** provided that the collection of all **C**-objects is a set and not a class. Otherwise it is called **large**.

Example 2.3.51 All the categories from Table 2.1 are large. The categories $\mathbf{C}(P)$ and $\mathbf{C}(M)$, where P is a poset and M is a monoid, are small.

Remark 2.3.52 If **C** is a small category, then for all $A, B \in Ob(\mathbf{C}), Hom_{\mathbf{C}}(A, B)$ is a set, usually called **hom-set** of A and B. We say that a category is **locally small** when for all $A, B \in Ob(\mathbf{C}), Hom_{\mathbf{C}}(A, B)$ is a set and not a class. If **C** is a locally small category, it is possible to define some functors from **C** to **Set**, called **hom-functors**, that play a central role in the development of categoy theory, as we shall se later on, when we introduce the concept of functors.

2.4 Bibliographic notes

[Gol86], [EP72], [AM85], [BW90] and [Wal91] are very nice introductions to category theory. Especially, [Gol86] is an excellent beginner's book, since it makes liberal use of simple, set-theoretic examples and motivating intuitions. [BW90] and [Wal91] present in some detail a extensive number of examples both from discrete mathematical structures and computer science. The presentation of basic concepts of algebraic specification is adapted from [EM85]. [EP72] is especially recommended for computer scientists interested in a categorical treatment of the main concepts related to automata theory. [Poi92] is a very modern and detailed introduction to an quite extensive part of category theory. It is especially useful for computer scientists since it uses many concepts from functional programming languages both in the motivation as well as in the illustration of the constructions. For a short and highly readable discussion on foundational aspects of category theory (which we don't treat in this text), we especially recommend [AHS90], chapter 2 and [MaC71] pp. 21-26.

Chapter 3

Basic concepts

Commutative diagrams are the categorist's way of expressing equations. The first section introduces the concepts of diagrams and commutative diagrams. In this work we make heavy use of commutative diagrams, and the reader will see that they are quite valuable in visualizing complex situations.

We also examine a number of standard set-theoretic constructions (e.g., injective, surjective and bijective functions) and reformulate them in the language of morphisms. The general theme, as mentioned in the introduction, is that concepts defined by reference to "internal" membership structure of a set are to be characterized "externally" by reference to connections with other sets, these connections being established by functions. Once these concepts are defined entirely in terms of morphisms, we generalize them to any category.

The concept of category is well balanced in the sense that it allows an economical and useful *duality*. Thus in category theory the "two for the price of one" principle holds: every concept is two concepts, and every result is two results. Section 3.4.1 introduces this very useful concept which culminates in the presentation of a most valuable result: *The Duality Principle*. Moreover, 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 componentwise. Also, a structure may have substructures, which are closed under the operations. Another construction that is possible in many cases is the formation of a "free" structure of a given type for a given set. In sections 3.4.2 and 3.4.3 we generalize these constructions in the categorical setting.

3.1 Diagrams

We now give a somewhat informal definition for the concept of a diagram. Later, in section 6.1.5, we present a precise, categorical definition.

Definition 3.1.1 *A* **diagram** *in a category* $\mathbf{C}$ *is a collection of vertices and directed edges, consistently labeled with objects and morphisms of* $\mathbf{C}$*, where "consistently" means that if an edge in the diagram is labeled with a morphism* f*, and* f *has domain* A *and codomain* B*, then the endpoints of this edge must be labeled with* A *and* B*.*

Definition 3.1.2 *A diagram in a category* **C** *is said to* **commute** *(or is said to be* **commutative***) if, for each pair of vertices v_1 and v_2 labeled by X and Y, all paths in the diagram from X to Y are equal, in the sense that each path in the diagram determines a composite morphism and these composites are equal in* **C***.*

As an example, saying that the diagram below commutes is exactly the same as saying that $f \circ g' = g \circ f'$.

$$\begin{tikzcd} X \arrow[r, "f'"] \arrow[d, "g'"'] & Z \arrow[d, "g"] \\ W \arrow[r, "f"'] & Y \end{tikzcd}$$

A useful convention normally used to denote that a certain diagram commutes is to put the symbol "=" inside of the diagram. In this case the above diagram could be represented as follows:

$$\begin{tikzcd} X \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "="] & Z \arrow[d, "g"] \\ W \arrow[r, "f"'] & Y \end{tikzcd}$$

Example 3.1.3 The associativity and identity axioms, in the definition of a category, are represented in the following (commutative) diagrams:

$$\begin{tikzcd} A \arrow[r, "h \circ (g \circ f) = (h \circ g) \circ f"] \arrow[d, "f"'] \arrow[dr, "g \circ f"] & D \\ B \arrow[r, "g"'] \arrow[ur, "h \circ g"'] & C \arrow[u, "h"'] \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "id_A"] \arrow[dr, "f"'] & A \arrow[d, "f"] \arrow[dr, "f"] & \\ & B \arrow[r, "id_B"'] & B \end{tikzcd}$$

When a property is stated in terms of commutative diagrams, proofs involving that property can often be given "visually". The following proposition demonstrates this technique.

Proposition 3.1.4 *If both inner squares of the following diagram commute, then so does the outer rectangle.*

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "a"'] \arrow[dr, phantom, "="] & B \arrow[r, "f'"] \arrow[d, "b"] \arrow[dr, phantom, "="] & C \arrow[d, "c"] \\
A' \arrow[r, "g"] & B' \arrow[r, "g'"] & C'
\end{tikzcd}
$$

Proof:

$$
\begin{aligned}
(g' \circ g) \circ a &= g' \circ (g \circ a) && \text{(associativity)} \\
&= g' \circ (b \circ f) && \text{(commutativity of first square)} \\
&= (g' \circ b) \circ f && \text{(associativity)} \\
&= (c \circ f') \circ f && \text{(commutativity of second square)} \\
&= c \circ (f' \circ f) && \text{(associativity)}
\end{aligned}
$$

$\square$

Definition 3.1.5 *Let $f : A \to B$ and $g : C \to D$ be functions on sets. Then the* **product function** $f \times g : A \times C \to B \times D$ *is defined by:* $f \times g(\langle a, c \rangle) = \langle f(a), g(c) \rangle$.

Example 3.1.6 Now, the definition of automata homomorphism can be made more simple and more understandable with the help of commutative diagrams. Thus, if A and A' are finite automata, then $f : A \to A'$ is an automata homomorphism if and only if the following diagram commutes:

$$
\begin{tikzcd}
O \arrow[d, "f_O"'] \arrow[dr, phantom, "="] & I \times S \arrow[l, "\lambda"'] \arrow[r, "\delta"] \arrow[d, "f_I \times f_S"] \arrow[dr, phantom, "="] & S \arrow[d, "f_S"] \\
O' & I' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] & S'
\end{tikzcd}
$$

In fact, the only missing information in this diagram it is that it does not represent the fact that a automata homomorphism must also preserve the initial state. Later, in section 4.2, when we study terminal objects, we will see how to represent this fact.

We now restate proposition 2.3.42 and prove it with the help of commutative diagrams.

Proposition 3.1.7 *If $f : A \to A'$ and $g : A' \to A''$ are automata homomorphisms then $g \circ f : A \to A''$ is also an automata homomorphism.*

Proof: Consider the next diagram:

$$
\begin{tikzcd}
O & I \times S \arrow[l, "\lambda"'] \arrow[r, "\delta"] \arrow[d, "f_I \times f_S"] & S \arrow[d, "f_S"] \\
O' \arrow[from=u, "f_O"'] & I' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] \arrow[d, "g_I \times g_S"] & S' \arrow[d, "g_S"] \\
O'' \arrow[from=u, "g_O"'] & I'' \times S'' \arrow[l, "\lambda''"'] \arrow[r, "\delta''"] & S''
\end{tikzcd}
$$

By assumption, we have that (1) and (2) commute since f is an automata homomorphism, and that (3) and (4) commute since g is also an automata homomorphism. To see that the whole diagram commutes, note that

$$
\begin{aligned}
&\delta'' \circ ((g_I \times g_S) \circ (f_I \times f_S)) = \\
&= (\delta'' \circ (g_I \times g_S)) \circ (f_I \times f_S) && \text{(associativity)} \\
&= (g_S \circ \delta') \circ (f_I \times f_S) && \text{(by (4))} \\
&= g_S \circ (\delta' \circ (f_I \times f_S)) && \text{(associativity)} \\
&= g_S \circ (f_S \circ \delta) && \text{(by (2))} \\
&= (g_S \circ f_S) \circ \delta && \text{(associativity)}
\end{aligned}
$$

However, it can easily be verified (using definition 3.1.5) that $(g_I \times g_S) \circ (f_I \times f_S) = (g_I \circ f_I) \times (g_S \circ f_S)$, and hence that

$$\delta'' \circ (g_I \circ f_I) \times (g_S \circ f_S) = (g_S \circ f_S) \circ \delta$$

Similarly, it can be shown that

$(g_O \circ f_O) \circ \lambda = \lambda'' \circ ((g_I \circ f_I) \times (g_S \circ f_S))$.

Thus, we have shown that $g \circ f : A \to A''$ is an automata homomorphism.

Remark 3.1.8 The fact that $(g_I \times g_S) \circ (f_I \times f_S) = (g_I \circ f_I) \times (g_S \circ f_S)$ can be deduced immediately by applying proposition 4.3.17 which we will see later. That it is the reason we have here avoided giving the (somewhat) more tedious (and useless for the rest of this text) set-theoretic proof.

Example 3.1.9 We restate now definition 2.3.34 using the maps $s, t : E \to V$ by saying that if G and G' are graphs then $f = (f_E, f_V)$ is a graph morphism if and only if for all $e \in E_G$ we have that $s' \circ f_E = f_V \circ s$ and $t' \circ f_E = f_V \circ t$, or simply saying that the next diagram (for source and target maps, separately) commutes:

$$\begin{tikzcd}
E \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f_E"'] & V \arrow[d, "f_V"] \\
E' \arrow[r, shift left, "s'"] \arrow[r, shift right, "t'"'] & V'
\end{tikzcd}$$

Now using the above definition for graph morphism, we can state the next

Proposition 3.1.10 *The composition of two graph morphisms $f : G \to G'$ and $f' : G' \to G''$, defined by $f' \circ f = (f'_E \circ f_E, f'_V \circ f_V)$ from G to G'', is again a graph morphism.*

Proof: Consider the next diagram:

$$\begin{tikzcd}
E \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f_E"'] \arrow[dr, phantom, "(1)"] & V \arrow[d, "f_V"] \\
E' \arrow[r, shift left, "s'"] \arrow[r, shift right, "t'"'] \arrow[d, "f'_E"'] \arrow[dr, phantom, "(2)"] & V' \arrow[d, "f'_V"] \\
E'' \arrow[r, shift left, "s''"] \arrow[r, shift right, "t''"'] & V''
\end{tikzcd}$$

By assumption we have that (1) and (2) commutes, since f and g are graph morphisms. To see that the whole diagram commutes note that

$(f'_V \circ f_V) \circ s =$
$= f'_V \circ (f_V \circ s)$ (associativity)
$= f'_V \circ (s' \circ f_E)$ (by (1))
$= (f'_V \circ s') \circ f_E$ (associativity)
$= (s'' \circ f_{E'}) \circ f_E$ (by (2))
$= s'' \circ (f_{E'} \circ f_E)$ (associativity)

Similarly, it can be shown that $(f'_V \circ f_V) \circ t = t'' \circ (f_{E'} \circ f_E)$. Thus, we have shown that

$$\begin{tikzcd}
E \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f'_E \circ f_E"'] & V \arrow[d, "f'_V \circ f_V"] \\
E'' \arrow[r, shift left, "s''"] \arrow[r, shift right, "t''"'] & V''
\end{tikzcd}$$

and hence that $f' \circ f = (f'_E \circ f_E, f'_V \circ f_V)$ from G to G'' is indeed a graph morphism.

□

3.2 Basic types of morphisms

Definition 2.1.8 presents a set-theoretic definition of injective functions (which are morphisms in the category **Set**). Let us try to characterize the injective functions without mentioning elements.

Then suppose that $f : B \to C$ is injective and also that we have two functions $h : B \to B$ and $g : B \to B$ such that $f(g(b)) = f(h(b))$ for every $b \in B$. By assumption, f is injective and hence $g(b) = h(b)$ for all $b \in B$, i.e., $g = h$.

We now verify the converse (i.e., if $f \circ g = f \circ h \Rightarrow g = h$ then f is injective) by showing that if $f : B \to C$ is not injective, we can construct $g, h : B \to B$ for which $f \circ g = f \circ h$ and $g \neq h$ (using here the logical equivalence $P \Rightarrow Q$ *iff* $\neg Q \Rightarrow \neg P$). Please, note that

$$\begin{array}{rll} \neg(f \circ g = f \circ h \Rightarrow g = h) & \textit{iff} & \neg(\neg(f \circ g = f \circ h) \vee g = h) \\ & \textit{iff} & f \circ g = f \circ h \wedge \neg(g = h) \end{array}$$

Then, suppose that $f : B \to C$ is not injective. In this way, we can find distinct elements $b_1, b_2 \in B$ such that $f(b_1) = f(b_2)$. Now we define $g = id_B$ while $h : B \to B$ is defined as:

$$h(b) = \begin{cases} b & \text{se } b \neq b_1 \\ b_2 & \text{if } b = b_1 \end{cases}$$

Clearly, $f \circ g(b) = f \circ id_B(b) = f(b)$ for all $b \in B$, while $f \circ h(b) = f(b)$ for $b \neq b_1$, and $f \circ h(b_1) = f(b_2) = f(b_1)$ by our choice of b_1 and b_2. Therefore, we have $f \circ g = f \circ h$ but $g \neq h$, since $h(b_1) \neq g(b_1)$.

The observations just made motivate the following

Definition 3.2.1 *Let* **C** *be a category. An morphism $f \in$* **C**(B, C) *is a* **monomorphism** *(or is* **monic***) if for any A and any pair of morphisms $g \in$* **C**(A, B) *and $h \in$* **C**(A, B)*, the equality $f \circ g = f \circ h$ implies $g = h$ (i.e., f is left-cancelable with respect to composition).*

In this case, the equality $f \circ g = f \circ h$ is represented by the following diagram:

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

Corollary 3.2.2 *In* **Set***, a function is a monomorphism if and only if it is injective.*

Proof: See the discussion before 3.2.1. □

Proposition 3.2.3

1. *In any category, if two morphisms $f : A \to B$ and $g : B \to C$ are monic, then their composition $g \circ f$ is also monic.*
2. *Also, if $g \circ f$ is monic, then f is also monic.*

Proof: For any $h, k : D \to A$, such that we have:

1. $(g \circ f) \circ h = (g \circ f) \circ k \Rightarrow$
$\Rightarrow g \circ (f \circ h) = g \circ (f \circ k)$ (associativity)
$\Rightarrow f \circ h = f \circ k$ (g is monic)
$\Rightarrow h = k$ (f is monic)

2. $f \circ h = f \circ k \Rightarrow$
$\Rightarrow g \circ (f \circ h) = g \circ (f \circ k)$ (composition)
$\Rightarrow (g \circ f) \circ h = (g \circ f) \circ k$ (associativity)
$\Rightarrow h = k$ ($g \circ f$ is monic)

□

We now try also to characterize surjective functions without mentioning elements.

To do so, suppose that $f : A \to B$ is surjective, and suppose that we have two functions $g, h : B \to C$ with the special property that $g(f(a)) = h(f(a))$ for all $a \in A$. As f is surjective, every $b \in B$ is a $f(a)$ for at least one $a \in A$, and this implies that $g(b) = h(b)$ for every $b \in B$, i.e., $g = h$.

Conversely (using here the logical equivalence $P \Rightarrow Q$ *iff* $\neg Q \Rightarrow \neg P$), suppose that f is not surjective. Then there exists some $b_1 \in B$ that it is not an $f(a)$ for any $a \in A$. Take any $b_2 \neq b_1$, and then define the two functions g and h in the following way:

$$g = id_B : B \to B$$

$$h(b) = \begin{cases} b & \text{if } b \neq b_1 \\ b_2 & \text{if } b = b_1 \end{cases}$$

Thus, $g \circ f(a) = id_B(f(a)) = f(a)$ for all $a \in A$, and since $f(a) \neq b_1$ for every $a \in A$, we also have $h \circ f(a) = h(f(a)) = f(a)$. Therefore, we have that $g \circ f = h \circ f$, although it is clear that $g \neq h$, since that $g(b_1) \neq h(b_1)$.

Generalizing the observations just made we get the following

Definition 3.2.4 *Let* $\mathbf{C}$ *be a category. A morphism $f \in \mathbf{C}(A, B)$ is an epimorphism (or* **comonomorphism** *or only* **epic***) if for any C and pair of morphisms $g \in \mathbf{C}(B, C)$ and $h \in \mathbf{C}(B, C)$, the equality $g \circ f = h \circ f$ implies $g = h$ (i.e., f is right-cancelable with respect to composition).*

In this case, the equality $g \circ f = h \circ f$ is represented by the following diagram:

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

Proposition 3.2.5 *In* **Set** *a function is* **epic** *if and only if it is surjective.*

Proof: See discussion before 3.2.4. $\square$

Proposition 3.2.6

1. *In any category, if two morphisms $f : B \to A$ and $g : C \to B$ are epic, then their composition $f \circ g$ is also epic.*

2. *Also, if $f \circ g$ is epic, then so is f.*

Proof: For any $h, k : A \to D$ we have:

1. $h \circ (f \circ g) = k \circ (f \circ g) \Rightarrow$

$\Rightarrow (h \circ f) \circ g = (k \circ f) \circ g$	(associativity)
$\Rightarrow h \circ f = k \circ f$	(g is epic)
$\Rightarrow h = k$	(f is epic)

2. $h \circ f = k \circ f \Rightarrow$

$\Rightarrow (h \circ f) \circ g = (k \circ f) \circ g$	(composition)
$\Rightarrow h \circ (f \circ g) = k \circ (f \circ g)$	(associativity)
$\Rightarrow h = k$	($f \circ g$ is epic)

$\square$

Example 3.2.7 Both $(\mathbb{Z}, +, 0)$, the monoid of integers under addition, and $(\mathbb{N}, +, 0)$, the monoid of nonnegative integers under addition, are objects of the category **Mon**. The inclusion function $i : (\mathbb{N}, +, O) \to (\mathbb{Z}, +, 0)$ that maps each nonnegative integer z to the integer z is a monomorphism, as we would expect by analogy with **Set**. But i is also an epimorphism, although it is clearly *not* surjective. To see this, assume that $f \circ i = g \circ i$ for two homomorphisms f and g from $(\mathbb{Z}, +, 0)$ to some monoid $(M, *, e)$. Take any $z \in \mathbb{Z}$. If $z \geq 0$, then it is the image under i of the same z considered as an element of $\mathbb{N}$. Now, $f(z) = f(i(z)) = g(i(z)) = g(z)$, and so $f \circ i = g \circ i \Rightarrow f = g$. If $z < 0$, then $-z \geq 0$ and $-z \in \mathbb{N}$. We reason as follows:

$f(z) =$

$= f(z) * e$	(e is the identity of $(M, *e)$)
$= f(z) * g(0)$	(g is a monoid homomorphism)
$= f(z) * g(-z + z)$	(elementary arithmetic)
$= f(z) * (g(-z) * g(z))$	(g is a monoid homomorphism)

$= ((f(z) * g(-z)) * g(z)$ (associativity of $*$)
$= ((f(z) * g(i(-z))) * g(z)$ (i is an inclusion)
$= ((f(z) * f(i(-z))) * g(z)$ (assumption)
$= (f(z) * f(-z)) * g(z)$ (i is an inclusion)
$= f(z + -z) * g(z)$ (f is a monoid homomorphism)
$= f(0) * g(z)$ (elementary arithmetic)
$= e * g(z)$ (f is a monoid homomorphism)
$= g(z)$ (e is the identity of $(M, *, e)$)

Since $f(z) = g(z)$ for all z, we have $f = g$, so i is (also) an epimorphism.

We have already seen that a bijective function is both injective and surjective. If $f : A \to B$ is bijective, then the passage from A to B under f can be reversed or "inverted". We can think of f as being simply a "relabeling" of A. Any $b \in B$ is the image $f(a)$ of some $a \in A$ (surjective property) and in fact is the image of only one such a (injective property). Thus the rule which assigns to b this unique a has

$$g(b) = a \text{ iff } f(a) = b$$

and establishes a function $g : B \to A$ which has

$$g(f(a)) = g(b) = a, \text{ for all } a \in A \text{ and}$$

$$f(g(b)) = f(a) = b, \text{ for all } b \in B.$$

Hence $g \circ f = id_A$ and $f \circ g = id_B$.

A function that is related to f in this way is said to be an inverse of f.

Generalizing this idea to any category we have the following

Definition 3.2.8 *Let* $\mathbf{C}$ *be a category. A morphism* $f \in \mathbf{C}(A, B)$ *is an* **isomorphism** *(or is* **iso***) if there exists a morphism* $f^{-1} \in \mathbf{C}(B, A)$*, called* inverse *of* f*, such that* $f \circ f^{-1} = id_B$ *and* $f^{-1} \circ f = id_A$*. In this case, the objects* A *and* B*, are said to be isomorphic, which we represent as* $A \cong B$*. The following diagram illustrates this definition.*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "id_A"'] & B \arrow[r, "id_B"] \arrow[d, "f^{-1}"] \arrow[dl, phantom, "="] \arrow[dr, phantom, "="] & B \\ & A \arrow[ur, dotted, "f"'] & \end{tikzcd}$$

Proposition 3.2.9 *In* **Set***, a morphism is an isomorphism if and only if it is bijective.*

Proof:"$\Rightarrow$ ": If $f : A \to B$ is an isomorphism, then there exists $f^{-1} : B \to A$ such that $f \circ f^{-1} = id_B$ and $f^{-1} \circ f = id_A$. To see that f is an epimorphism (and therefore surjective in **Set**) note that

$$
\begin{aligned}
& g \circ f = h \circ f \Rightarrow \\
& \Rightarrow (g \circ f) \circ f^{-1} = (h \circ f) \circ f^{-1} && \text{(composition)} \\
& \Rightarrow g \circ (f \circ f^{-1}) = h \circ (f \circ f^{-1}) && \text{(associativity)} \\
& \Rightarrow g \circ id_B = h \circ id_B && (f \text{ is iso}) \\
& \Rightarrow g = h && \text{(identity)}
\end{aligned}
$$

To see that f is monic (and therefore injective in **Set**) note that

$$
\begin{aligned}
& f \circ g = f \circ h \Rightarrow \\
& \Rightarrow f^{-1} \circ (f \circ g) = f^{-1} \circ (f \circ h) && \text{(composition)} \\
& \Rightarrow (f^{-1} \circ f) \circ g = (f^{-1} \circ f) \circ h && \text{(associativity)} \\
& \Rightarrow id_A \circ g = id_A \circ h && (f \text{ is iso}) \\
& \Rightarrow g = h && \text{(identity)}
\end{aligned}
$$

According to the implication just proved, we have that if f is an isomorphism in **Set**, then f is injective (monic) and surjective (epic). But then f is also bijective.

"$\Leftarrow$": See remarks before definition 3.2.8.

□

Corollary 3.2.10 *Every iso is also monic and epic.*

Proof: □

Remark 3.2.11 The reverse implication of the above corollary does not hold in every category, as for instance the category **PAlg**, of partial algebras and homomorphisms between them. A partial algebra for a given signature is just like a total algebra with the crucial difference that the operation symbols are interpreted as partial and not total functions (see [Bur86] for the details).

Proposition 3.2.12 *Every identity morphism is iso.*

Proof: Only define $id_A^{-1} = id_A$ and then we have that $id_A \circ id_A^{-1} = id_A$ and $id_A^{-1} \circ id_A = id_A$.

□

Proposition 3.2.13 *If f is iso then so is f^{-1}.*

Proof: Straightforward from definition 3.2.8, considering f as the inverse of f^{-1}.

□

Proposition 3.2.14 *If $f : A \to B$ and $g : B \to C$ are iso, then $g \circ f$ is also iso, with $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.*

Proof:

$$
\begin{aligned}
&(g \circ f) \circ (f^{-1} \circ g^{-1}) = \\
&= g \circ (f \circ f^{-1}) \circ g^{-1} && \text{(associativity)} \\
&= g \circ id_B \circ g^{-1} && \text{(f is iso)} \\
&= g \circ g^{-1} && \text{(identity)} \\
&= id_C && \text{(g is iso)}
\end{aligned}
$$

$$
\begin{aligned}
&(f^{-1} \circ g^{-1}) \circ (g \circ f) = \\
&= f^{-1} \circ (g^{-1} \circ g) \circ f && \text{(associativity)} \\
&= f^{-1} \circ id_B \circ f && \text{(g is iso)} \\
&= f^{-1} \circ f && \text{(identity)} \\
&= id_A && \text{(f is iso)}
\end{aligned}
$$

□

Definition 3.2.15 *Let P be a class and $\approx$ a relation on the elements of P. $\approx$ is called an* **equivalence relation**, *if for all x, y, $z \in P$ the following conditions are satisfied:*

1. $x \approx x$ *($\approx$ is reflexive)*
2. $x \approx y \Rightarrow y \approx x$ *($\approx$ is symmetric)*
3. $x \approx y \wedge y \approx z \Rightarrow x \approx z$ *($\approx$ is transitive)*

Proposition 3.2.16 *Let* **C** *be a category. The isomorphisms in* **C** *provide an equivalence relation on $Ob(\mathbf{C})$.*

Proof: According to proposition 3.2.12, $A \cong A$, and therefore, $\cong$ is reflexive. According to proposition 3.2.13, if $A \cong B$ then $B \cong A$, and so $\cong$ is symmetric. According to proposition 3.2.14, if $A \cong B$ and $B \cong C$ then $A \cong C$, and hence $\cong$ is also transitive.

□

Example 3.2.17 In the category **Grp**, two groups are isomorphic if there exists a group homomorphism from one to the other, whose set-theoretic inverse is also a group homomorphism.

Example 3.2.18 In the category **Poset**, two posets are isomorphic if there exists a monotonic function from one to the other, whose set-theoretic inverse is also a monotonic function.

CATEGORY	*MONOMORPHISMS*	*EPIMORPHISMS*	*ISOMORPHISMS*
Set	injective functions	surjective functions	bijective functions
FinSet	injective functions	surjective functions	bijective functions
Rel	injective relations	surjective relations	relational isomorphisms
Poset	injective monot. functions	surjective monot. functions	bijective monot. functions
Grp	injective group homomorphisms	surjective group homomorphisms	bijective group homomorphisms
Σ-**Alg**	injective Σ-homomorphisms	surjective Σ-homomorphisms	bijective Σ-homomorphisms
Aut	injective autom. homomorphisms	surjective autom. homomorphisms	bijective autom. homomorphisms
Graph	injective graph morphisms	surjective graph morphisms	bijective graph morphisms

Table 3.1: Characterization of mono-, epi-, and isomorphisms in examples of categories

Example 3.2.19 In a poset $(P, \leq_P)$ considered as a category, if $f : p \to q$ has a inverse $f^{-1} : q \to p$, then $p \leq q$ and $q \leq p$. However, due to the fact that $\leq_P$ is antisymmetric we must have $p = q$. Hence, f must be the morphism $id_p : p \to p$, i.e., the only isomorphisms are the identities.

Example 3.2.20 A group, considered as a category, is the same thing as a category with only one object, where each morphism is an isomorphism (see definition 2.3.16 and example 2.3.44).

In the above examples, as well as in proposition 3.2.9, isomorphic objects "look the same". One can pass freely from one to the other by an isomorphism and its inverse. Moreover, these morphisms, which establish a "one-to-one correspondence" or "matching" between the elements of the two objects, preserve any relevant structure.

An object will be said to be *unique up to isomorphism* in possession of a particular attribute, if the only other objects possessing that attribute are isomorphic to it. A concept will be defined up to isomorphism if its description specifies a particular entity, not uniquely, but only uniquely up to isomorphism.

Category theory is then the subject that provides an abstract formulation of the idea of mathematical isomorphism and studies notions that are invariant under all forms of isomorphism. In category theory, *is isomorphic to* is virtually synonymous with *is*. Indeed, most of the basic definitions and constructions that one can perform in a category do not specify things uniquely at all, but only, as we shall see, *uniquely up to isomorphism.*

In Table 3.1 we show some (examples of) categories presented so far with the characterization of their mono-, epi- and isomorphisms.

3.3 Other types of morphisms

We have already seen that a morphism $f : A \to B$ in a category is an isomorphism if it has an inverse $g : B \to A$ which must satisfy both the equations $g \circ f = id_A$ and $f \circ g = id_B$. If it only satisfies the second equation, $f \circ g = id_B$, then f is an **left inverse** of g and (naturally) g is a **right inverse** of f.

Definition 3.3.1 *Let* $\mathbf{C}$ *be a category. If* $f \in \mathbf{C}(A, B)$ *has a right inverse* $g \in \mathbf{C}(B, A)$*, i.e.,* $f \circ g = id_B$*, then it is called a* **retraction** *or* **split epimorphism**.

Example 3.3.2 A morphism in **Set** is a retraction if and only if it is a surjective function.

Proposition 3.3.3

1. *If* $f : A \to B$ *and* $g : B \to C$ *are retractions then* $g \circ f : A \to C$ *is also a retraction.*
2. *Every retraction is a epimorphism.*

Proof:
1. By assumption we have $h : B \to A$ such that $f \circ h = id_B$ and $k : C \to B$ such that $g \circ k = id_C$. Thus

$$
\begin{aligned}
&(g \circ f) \circ (h \circ k) = \\
&= g \circ (f \circ h) \circ k && \text{(associativity)} \\
&= g \circ id_B \circ k && \text{(assumption)} \\
&= g \circ k && \text{(identity)} \\
&= id_C && \text{(assumption)}
\end{aligned}
$$

2. By assumption, if $f : A \to B$ is a retraction, then there is a $g : B \to A$ such that $f \circ g = id_B$. Thus for any $h : B \to C$ and $k : B \to C$ such that $h \circ f = k \circ f$ we have:

$$
\begin{aligned}
&h \circ f = k \circ f \Rightarrow \\
&\Rightarrow (h \circ f) \circ g = (k \circ f) \circ g && \text{(composition)} \\
&\Rightarrow h \circ (f \circ g) = k \circ (f \circ g) && \text{(associativity)} \\
&\Rightarrow h \circ id_B = k \circ id_B && (f \text{ is a retraction}) \\
&\Rightarrow h = k && \text{(identity)}
\end{aligned}
$$

□

Definition 3.3.4 *Let* $\mathbf{C}$ *be a category. If* $f \in \mathbf{C}(A, B)$ *has a left inverse* $g \in \mathbf{C}(B, A)$ *, i.e.,* $g \circ f = id_A$*, then it is called a* **section** *or* **split monomorphism**.

Example 3.3.5 A morphism in **Set** is a section if and only if it is an injective function and is not the empty function from the empty set to a nonempty set.

Proposition 3.3.6

1. *If $f : A \to B$ and $g : B \to C$ are sections then $g \circ f : A \to C$ is also a section.*

2. *Every section is a monomorphism.*

Proof:

1. By assumption we have $h : B \to A$ such that $h \circ f = id_A$ and $k : C \to B$ such that $k \circ g = id_B$. Thus

$$
\begin{aligned}
&(h \circ k) \circ (g \circ f) = \\
&= h \circ (k \circ g) \circ f && \text{(associativity)} \\
&= h \circ id_B \circ f && \text{(assumption)} \\
&= h \circ f && \text{(identity)} \\
&= id_A && \text{(assumption)}
\end{aligned}
$$

2. By assumption, if $f : A \to B$ is a section, then there is a $g : B \to A$ such that $g \circ f = id_A$. Thus for any $h : C \to A$ and $k : C \to A$ such that $f \circ h = f \circ k$ we have:

$$
\begin{aligned}
&f \circ h = f \circ k \Rightarrow \\
&\Rightarrow g \circ (f \circ h) = g \circ (f \circ k) && \text{(composition)} \\
&\Rightarrow (g \circ f) \circ h = (g \circ f) \circ k && \text{(associativity)} \\
&\Rightarrow id_A \circ h = id_A \circ k && (f \text{ is a retraction}) \\
&\Rightarrow h = k && \text{(identity)}
\end{aligned}
$$

□

Remark 3.3.7 Many textbooks on category theory define other types of morphisms, e.g., constant morphisms, zero morphisms, bimorphisms, quotient object, and more. See [AHS90] for a detailed treatment of these concepts.

3.4 New categories from old ones

3.4.1 Duality

Categorical duality is the process "Reverse all morphisms". To each category $\mathbf{C}$ we associate the opposite category $\mathbf{C}^{op}$ by reversing the direction of all morphisms in $\mathbf{C}$. More formally, we have the following

Definition and proposition 3.4.1.1 *For any category $\mathbf{C}$, the* **dual** *(or* **opposite***) category of $\mathbf{C}$, is the category $\mathbf{C}^{op}$, where $Ob(\mathbf{C}) = Ob(\mathbf{C}^{\mathbf{op}})$, and for each morphism $f : A \to B$ in $\mathbf{C}$ we have a corresponding morphism $f^{op} : B \to A$ in $\mathbf{C}^{\mathbf{op}}$ such that $cod(f^{op}) = dom(f) \wedge dom(f^{op}) = cod(f)$. These are the only morphisms in $\mathbf{C}^{\mathbf{op}}$. Moreover, composition is defined as follows: for any f^{op}, g^{op} in $\mathbf{C}^{\mathbf{op}}$ such that $dom(g^{op}) = cod(f^{op})$ one has:*

$$g^{op} \circ f^{op} = (f \circ g)^{op}$$

Proof: For any category $\mathbf{C}$, $\mathbf{C}^{op}$ satisfies the associativity and identity axioms as follows:

For any morphisms $f^{op} \in \mathbf{C}^{op}(A, B)$, $g^{op} \in \mathbf{C}^{op}(B, C)$, $h^{op} \in \mathbf{C}^{op}(C, D)$ we have:

$$
\begin{aligned}
h^{op} \circ (g^{op} \circ f^{op}) &= h^{op} \circ (f \circ g)^{op} && \text{(by 3.4.1.1)} \\
&= ((f \circ g) \circ h)^{op} && \text{(by 3.4.1.1)} \\
&= (f \circ (g \circ h))^{op} && \text{(}\mathbf{C}\text{ is a category)} \\
&= (g \circ h)^{op} \circ f^{op} && \text{(by 3.4.1.1)} \\
&= (h^{op} \circ g^{op}) \circ f^{op} && \text{(by 3.4.1.1)}
\end{aligned}
$$

while for any $f^{op} \in \mathbf{C}^{op}(A, B), id_A^{op} \in \mathbf{C}^{op}(A, A)$ we have:

$$
\begin{aligned}
f^{op} \circ id_A^{op} &= (id_A \circ f)^{op} && \text{(by 3.4.1.1)} \\
&= f^{op} && \text{(}\mathbf{C}\text{ is a category)}
\end{aligned}
$$

The equation $id_B^{op} \circ f^{op} = f^{op}$ holds by an analogous argument.

Corollary 3.4.1.2 For any category $\mathbf{C}$ we have $(\mathbf{C}^{op})^{op} = \mathbf{C}$.

Example 3.4.1.3 As expected, for any category $\mathbf{C}$ and for any morphisms $f \in \mathbf{C}(B, A)$, $g \in \mathbf{C}(C, B)$, $h \in \mathbf{C}(D, C)$, $id_A \in \mathbf{C}(A, A)$, $id_B \in \mathbf{C}(B, B)$, the associativity and identity axioms in $\mathbf{C}^{op}$ can be illustrated, respectively, by the following commutative diagrams:

$$
\begin{tikzcd}
A \arrow[r, "h^{op} \circ (g^{op} \circ f^{op}) = (h^{op} \circ g^{op}) \circ f^{op}"] \arrow[d, "f^{op}"'] \arrow[dr, "g^{op} \circ f^{op}" near start] & D \\
B \arrow[r, "g^{op}"'] \arrow[ur, "h^{op} \circ g^{op}" near end] & C \arrow[u, "h^{op}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "id_A^{op}"] \arrow[dr, "f^{op}"] & A \arrow[d, "f^{op}"] \arrow[dr, "f^{op}"] & \\
& B \arrow[r, "id_B^{op}"'] & B
\end{tikzcd}
$$

Remark 3.4.1 We think that it is worth pointing out that the duality construction is just an abstract, formal one. The usefulness of this concept will be made clear below as a way to provide a clear association between several categorical constructions and also as a way to provide for free a lot of results. On the other hand, in several categories already presented, as well as in other ones which will be introduced, the inversion of the direction of a morphism gives rise to meaningful constructions within set theory. For instance, if the morphism is a total function in **Set**, then the inversion of its direction (thus a morphism in $\mathbf{Set}^{op}$) "can be" interpreted as the inverse of a function and therefore as a relation (see 3.4.2.5). This has often been done in the literature, but it is much beyond of what the definition of a dual category states. It is useful as a guiding intuition for the reader, but it is at the same time, *misleading and incorrect*. Once this is clear, it is fruitful to present "concrete" interpretation for the respective dual construction, whenever this makes sense.

Example 3.4.1.4 If $\mathbf{C} = (P, \leq_P)$ is a poset considered as a category, then $\mathbf{C}^{op} = (P, \geq_P)$.

Example 3.4.1.5 If $\mathbf{C} = (M, \oplus, e_M)$ is a monoid considered as a category, then $\mathbf{C}^{op} = (M, \oplus, e_M)$ where $x \oplus y = y \oplus x$. Note that if the binary operation is commutative, i.e., if for all x, $y \in M$, $x \oplus y = y \oplus x$, then $\mathbf{C} = \mathbf{C}^{op}$.

Let the "atomic statements" of the theory of categories be statements built up with the usual undefined terms of category theory. Thus atomic statements are the ones like "$A = dom(f)$, $B = cod(f)$, $h = f \circ g$ and so on. So statements P about objects and morphisms in a given category $\mathbf{C}$ is defined to be any well-formed formula built up from types of atomic statements listed above in the usual fashion by means of the ordinary propositional connectives ("and", "or", "not", "implies", "if and only if") and the usual quantifiers ("for all", "there exists", "there is a unique"). Now, a "dual" statement P^{op} is formed by making the following replacements throughout in P :"dom" by "cod", $h = g \circ f$ by $h^{op} = f^{op} \circ g^{op}$, that is to say, morphisms and composites are reversed. Logic remains unchanged. A more formal presentation is given in the next

Definition 3.4.1.6

1. *A statement P is called a* **theorem of category theory** *if P holds for every category $\mathbf{C}$.*
2. *The statement $P^{op}(\mathbf{C})$ is called* **dual** *to $P(\mathbf{C})$ when*

   $P^{op}(\mathbf{C})$ has the same meaning as $P(\mathbf{C}^{op})$.

3. *The statement $P(\mathbf{C})$ is called* **self-dual** *when $P(\mathbf{C})$ has the same meaning as $P^{op}(\mathbf{C})$.*

Remarks 3.4.1.7

1. The intution behind 3.4.1.6.2 is that any property formulated in $\mathbf{C^{op}}$ has an associated equivalent formulation in $\mathbf{C}$ because of the way $\mathbf{C^{op}}$ is defined from $\mathbf{C}$.
2. In general one can obtain several dual statements from a specific one. A "canonical" dualization from a statement $P(\mathbf{C})$ can be obtained (in two steps) when one formulates the statement $P(\mathbf{C})$ as the statement $P(\mathbf{C}^{op})$ in the category $\mathbf{C}^{op}$, and then interpretes $P(\mathbf{C}^{op})$ as a statement in the category $\mathbf{C}$, i.e., by inversion of the direction of all morphisms in $P(\mathbf{C}^{op})$ one gets the statement $P^{op}(\mathbf{C})$ (see examples below).
3. Since $(\mathbf{C}^{op})^{op} = \mathbf{C}$, we have:

$$\begin{aligned}(P^{op})^{op}(\mathbf{C}) &= P^{op}(\mathbf{C}^{op}) \\ &= P((\mathbf{C}^{op})^{op}) \\ &= P(\mathbf{C})\end{aligned}$$

As an exercise in constructing dual statements, consider the next

Example 3.4.1.8 Consider the following statement of an object X in the category $\mathbf{C}$.

$P(\mathbf{C})(X) =_{def}$ For any $\mathbf{C}$-object A, there exists exactly one $\mathbf{C}$-morphism $f : A \to X$.

Step 1: In $P(\mathbf{C})(X)$ replace all occurrences of $\mathbf{C}$ by $\mathbf{C}^{op}$ and (rename) all occurrences of morphisms f for f^{op}, thus obtaining the property

$P(\mathbf{C}^{op})(X) =_{def}$ For any $\mathbf{C}^{op}$-object A, there exists exactly one $\mathbf{C}^{op}$-morphism $f^{op} : A \to X$.

Step 2: Translate $P(\mathbf{C}^{op})(X)$ into the logically equivalent statement:

$P^{op}(\mathbf{C})(X) =_{def}$ For any $\mathbf{C}$-object A, there exists exactly one $\mathbf{C}$-morphism $f : X \to A$.

In general, $P^{op}(\mathbf{C})(X)$ is not equivalent to $P(\mathbf{C})(X)$. For example, if we take $\mathbf{C}$ as $\mathbf{Set}$, we will see later that $P(\mathbf{Set})(X)$ holds if and only if X is a singleton set, and $P^{op}(\mathbf{Set})(X)$ holds if and only if X is the empty set.

Similarly, any statement about morphisms gives rise to a dual statement concerning morphisms in categories, as demonstrated by the next

Example 3.4.1.9 Consider the following statement of morphisms in the category $\mathbf{C}$.

$Q(\mathbf{C})(f : A \to B) =_{def}$ There exists a $\mathbf{C}$-morphism $g : B \to A$ such that $g \circ f = id_A$ in $\mathbf{C}$.

Step 1: In $Q(\mathbf{C})(f : A \to B)$ replace all occurrences of $\mathbf{C}$ by $\mathbf{C}^{op}$ and (rename) all occurrences of morphisms f for f^{op}, thus obtaining the property

$Q(\mathbf{C}^{op})(f^{op} : A \to B) =_{def}$ There exists a $\mathbf{C}^{op}$-morphism $g^{op} : B \to A$ such that $g^{op} \circ f^{op} = id_A^{op}$ in $\mathbf{C}^{op}$.

Step 2: Translate $Q(\mathbf{C}^{op})(f^{op} : A \to B)$ into the logically equivalent statement:

$Q^{op}(\mathbf{C})(f : B \to A) =_{def}$ There exists a $\mathbf{C}$-morphism $g : A \to B$ such that $f \circ g = id_A$ in $\mathbf{C}$.

The attent reader may have noted that $Q(\mathbf{C})(f)$ corresponds exactly to the definition of a **section** (see 3.3.4) and that $Q^{op}(\mathbf{C})(f)$ to the definition of a **rectraction** (see 3.3.1). But this says exactly that the concept of rectraction is dual to the concept of section.

Example 3.4.1.10 By a quick examination at definitions 3.2.1 and 3.2.4 we see that each one is simmetric to the other, and thus that the concept of epimorphism is dual to the concept of monomorphism.

Example 3.4.1.11 The dual construction of an isomorphism, i.e., a coisomorphism, is represented by the following diagram:

$$
\begin{tikzcd}
A & B \arrow[l, "f^{op}"'] & B \arrow[l, "id_B"'] \arrow[dl, dotted, "f^{op}"] \\
& A \arrow[ul, "id_A"] \arrow[u, "(f^{-1})^{op}"'] &
\end{tikzcd}
$$

However, the commutativity of this diagram says exactly that a coisomorphism is the same thing as an isomorphism, i.e., the concept of isomorphism is self-dual!

More complex statements $P(\mathbf{C})(A, B, \ldots, f, g, \ldots)$ that involve objetcs $A, B, \ldots$ and morphisms $f, g, \ldots$ in a category $\mathbf{C}$ can be dualized in a similar way.

If $P = P(\mathbf{C})(A, B, \ldots, f, g, \ldots)$ holds for all $\mathbf{C}$-objects $A, B, \ldots$ and all $\mathbf{C}$-morphisms $f, g, \ldots$, then we say that $\mathbf{C}$ has the property P or that P holds.

Now we are able to introduce the most important concept of this section which is presented in the powerful and extremely useful

Theorem 3.4.1.12 *The* **Duality Principle for Categories** *states: Whenever a statement P holds for all categories then the statement P^{op} also holds for all categories.*

Proof: Suppose that P holds for every category $\mathbf{C}$. This means that P also hold in $\mathbf{C}^{op}$, since by 3.4.1.1, $\mathbf{C}^{op}$ is also a category. By 3.4.1.6 we have that $P(\mathbf{C}^{op})$ has the same meaning as $P^{op}(\mathbf{C})$, which is the desired result.

$\square$

Remark 3.4.1.13 Because of this principle, each result in category theory has two equivalent formulations (which at first glance might seem to be quite different). However, only one of them need to be proved, since the other follows by virtue of the Duality Principle.

3.4.2 Subcategories

Definition 3.4.2.1 *A category* $\mathbf{D}$ *is a* **subcategory** *of a category* $\mathbf{C}$ *if:*

1. $Ob(\mathbf{D}) \subseteq Ob(\mathbf{C})$;
2. *for all* $A,\ B \in Ob(\mathbf{D})$, $\mathbf{D}(A, B) \subseteq \mathbf{C}(A, B)$;
3. *compositions and identities coincide with those of* $\mathbf{C}$.

Definition 3.4.2.2 *A subcategory* $\mathbf{D}$ *of* $\mathbf{C}$ *is said to be* **full** *if for all* $A,\ B \in Ob(\mathbf{D})$, $\mathbf{D}(A, B) = \mathbf{C}(A, B)$.

Example 3.4.2.3 For any category $\mathbf{C}$, the empty category $\mathbf{0}$ and $\mathbf{C}$ itself are full subcategories of $\mathbf{C}$.

Example 3.4.2.4 The subcategories of a monoid $(M, \oplus, e_M)$ considered as a category are exactly the submonoids of $(M, \oplus, e_M)$.

Example 3.4.2.5 Consider now the opposite category of **Set**, i.e., $\mathbf{Set}^{op}$, where the objects are sets and the morphisms are functions in the opposite direction. "Assume" a particulat interpretation of $f^{op} : B \to A$ as $f^{-1} : B \to A$, i.e., the inverse of a function $f : A \to B$. Then the construction acquires a familiar meaning and $\mathbf{Set}^{op}$ can be "considered" as a subcategory of **Rel**. However, it would not be a full subcategory. As an a example, for $A = \{a, b\}$ and $B = \{c\}$, we have that the relation $\rho = \{\langle a, c\rangle, \langle b, c\rangle\} \in \mathbf{Rel}(A, B)$, but certainly $\rho \notin \mathbf{Set}^{op}(A, B)$, since $\rho^{op} : B \to A$ would not be a function as we would have $\rho^{op}(c) = a \neq b = \rho^{op}(c)$.

Example 3.4.2.6 The category **FinSet** is a full subcategory of **Set**

Example 3.4.2.7 The category **Set** is a subcategory of **Pfn**. However, it is not a full subcategory, since between any two sets there are *much* more partial functions than total functions. Besides, **Pfn** is a subcategory of **Rel**. It is, however, not a full subcategory.

3.4.3 Other Constructions

Example 3.4.3.1 For any pair of categories **C** and **D**, we can construct the **product category $\mathbf{C} \times \mathbf{D}$** as follows:

1. An object in $\mathbf{C} \times \mathbf{D}$ is a pair $\langle A, B\rangle$, where A is an **C**-object and B is an **D**-object.

2. An $\mathbf{C} \times \mathbf{D}$-morphism $\langle A_1, B_1\rangle \to \langle A_2, B_2\rangle$ is a pair $\langle f, g\rangle$ where $f : A_1 \to A_2$ is an **C**-morphism, and $g : B_1 \to B_2$ is an **D**-morphism.

3. The composition of two morphisms $\langle f_1, g_1\rangle : \langle A_1, B_1\rangle \to \langle A_2, B_2\rangle$ and $\langle f_2, g_2\rangle : \langle A_2, B_2\rangle \to \langle A_3, B_3\rangle$ is a morphism $\langle f_2, g_2\rangle \circ \langle f_1, g_1\rangle = \langle f_2 \circ f_1, g_2 \circ g_1\rangle : \langle A_1, B_1\rangle \to \langle A_3, B_3\rangle$, i.e., the composition is defined "componentweise" according to the composition in **C** and **D**. Besides, the composition of morphisms in $\mathbf{C} \times \mathbf{D}$ is associative, i.e., for any morphisms $\langle f_1, g_1\rangle : \langle A_1, B_1\rangle \to \langle A_2, B_2\rangle$, $\langle f_2, g_2\rangle : \langle A_2, B_2\rangle \to \langle A_3, B_3\rangle$, $\langle f_3, g_3\rangle : \langle A_3, B_3\rangle \to \langle A_4, B_4\rangle$ we have that:

   $$\begin{aligned}
   &\langle f_3, g_3\rangle \circ \langle\langle f_2, g_2\rangle \circ \langle f_1, g_1\rangle\rangle = \\
   &= \langle f_3, g_3\rangle \circ \langle f_2 \circ f_1, g_2 \circ g_1\rangle && \text{(composition in } \mathbf{C} \times \mathbf{D}) \\
   &= \langle f_3 \circ \langle f_2 \circ f_1\rangle, g_3 \circ \langle g_2 \circ g_1\rangle\rangle && \text{(composition in } \mathbf{C} \times \mathbf{D}) \\
   &= \langle\langle f_3 \circ f_2\rangle \circ f_1, \langle g_3 \circ g_2\rangle \circ g_1\rangle && (\mathbf{C} \text{ and } \mathbf{D} \text{ are cat.}) \\
   &= \langle f_3 \circ f_2, g_3 \circ g_2\rangle \circ \langle f_1, g_1\rangle && \text{(composition in } \mathbf{C} \times \mathbf{D}) \\
   &= \langle\langle f_3, g_3\rangle \circ \langle f_2, g_2\rangle\rangle \circ \langle f_1, g_1\rangle && \text{(composition in } \mathbf{C} \times \mathbf{D})
   \end{aligned}$$

4. For any object $\langle A_1, B_1\rangle$, the identity morphism is defined as $\langle id_{A_1}, id_{B_1}\rangle$. So, for any $\langle f_1, g_1\rangle : \langle A_1, B_1\rangle \to \langle A_2, B_2\rangle$ we have that

   $$\langle f_1, g_1\rangle \circ \langle id_{A_1}, id_{B_1}\rangle = \langle f \circ id_{A_1}, g \circ id_{B_1}\rangle = \langle f_1, g_1\rangle$$

   In a similar way, it can be shown that $\langle id_{A_2}, id_{B_2}\rangle \circ \langle f_1, g_1\rangle = \langle f_1, g_1\rangle$.

Definition 3.4.3.2 *In a graph G, a* **path** *from a vertex v_1 to a vertex v_2 of length k is a sequence $(f_1, f_2, \ldots, f_k)$ of (not necessarily distinct) edges for which*

1. $s(f_k) = v_1$
2. $s(f_i) = t(f_{i-1})$, *for* $i = 2, \ldots, k$, *and*
3. $t(f_1) = v_2$.

where $s, t : E \to V$ are the source and target maps in the sense of 2.3.34.

Example 3.4.3.3 (The free category generated by a graph). For any given graph G, there is a category $\mathbf{F}(G)$ whose objects are the vertices of G and whose morphisms are the paths in 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, id_A is the empty path from A to A. The category $\mathbf{F}(G)$ is called the **free category generated by the graph** G. It is also called the **path category** of G.

The free category generated by the graph with one vertex and no edges is the category with one object and only the identity morphism (see example 2.3.47), 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).

It is useful to regard the free category generated by *any* graph as analogous to Kleene closure (free monoid) generated by a set (see 2.3.7 and 2.3.8). 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 edges can be strung together only head to tail, thus taking into account the typing. In chapter 7 we give a precise technical meaning to the word 'free' in the context of free construction.

Example 3.4.3.4 For any category $\mathbf{C}$, we can construct the morphism category $\mathbf{C}^{\to}$ as follows:

1. An object in $\mathbf{C}^{\to}$ is a morphism of $\mathbf{C}$.
2. An morphism in $\mathbf{C}^{\to}$ from the $\mathbf{C}^{\to}$-object $f_1 : A_1 \to B_1$ to the $\mathbf{C}^{\to}$-object $f_2 : A_2 \to B_2$ is a pair of $\mathbf{C}$-morphisms $(a, b) : (f_1 : A_1 \to B_1) \to (f_2 : A_2 \to B_2)$ such that the following diagram commutes:

$$\begin{tikzcd} A_1 \arrow[r, "a"] \arrow[d, "f_1"'] \arrow[dr, phantom, "="] & A_2 \arrow[d, "f_2"] \\ B_1 \arrow[r, "b"'] & B_2 \end{tikzcd}$$

3. The composition of two $\mathbf{C}^{\rightarrow}$-morphisms $(a_1, b_1) : (f_1 : A_1 \to B_1) \to (f_2 : A_2 \to B_2)$ and $(a_2, b_2) : (f_2 : A_2 \to B_2) \to (f_3 : A_3 \to B_3)$ is defined componentwise, i.e., $(a_2, b_2) \circ (a_1, b_1) = (a_2 \circ a_1, b_2 \circ b_1)$. Moreover, the outer rectangle of the next diagram should commute.

$$\begin{tikzcd}
A_1 \arrow[r, "a_1"] \arrow[d, "f_1"'] \arrow[dr, phantom, "="] & A_2 \arrow[r, "a_2"] \arrow[d, "f_2"] \arrow[dr, phantom, "="] & A_3 \arrow[d, "f_3"] \\
B_1 \arrow[r, "b_1"'] & B_2 \arrow[r, "b_2"'] & B_3
\end{tikzcd}$$

The axioms for associativity and identity are inherited from $\mathbf{C}$ (see previous example).

There is another very important construction which is called **comma-category**. However, to present it in all its generality we have to wait until chapter 6, since we still have not met the concept of a functor.

3.5 Bibliographic notes

The concept of diagram is elementary in category theory and it is introduced virtually in any introduction to the subject. We have covered the most common types of morphisms that may exists in a given category. However, the reader may wish to look at [AHS90], chapter 7, where several other types of morphisms are presented. The concept of duality is given a detailed presentation with a careful motivation in [AM75].

Chapter 4

Basic constructions

In this chapter we consider some fundamental categorical constructions, i.e., particular objects (and morphisms) that satisfy a given set of axioms described in the language of Category Theory. Since in this language there is no way to look at the internal membership structure of objects, all the concepts must be defined by their relations with other objects, and these relations are established by the existence and equality of particular morphisms. This property of the categorical language, if compared to the traditional set-theoretic jargon, may be well understood by an analogy with computer science; namely, the categorical description corresponds to an abstract data type specification, while the traditional set-theoretic approach is more similar to a concrete implementation.

4.1 Initial objects

What morphism properties distinguish $\emptyset$, the null or empty set, in **Set** ? Given a set A, can we find any function $\emptyset \to A$? Yes, there exists only one such function from $\emptyset$ to A, i.e., the *empty function.* To see this, observe that considering a function as a subset of the cartesian product, we have that $\emptyset$ is the only subset of $\emptyset \times A$, and hence, $\emptyset$ is the only function from $\emptyset$ to A. This observation leads to the following

Definition 4.1.1 *Let* $\mathbf{C}$ *be a category. An object* A *is a* **initial object** *in* $\mathbf{C}$ *if for any* $\mathbf{C}$*-object* A *there exists a unique morphism* $f \in \mathbf{C}(A, B)$

Proposition 4.1.2 *If* A *and* B *are initial objects in a category* $\mathbf{C}$*, then they are isomorphic.*

Proof: First note that, because A is initial, we know that there exists a unique morphism $f : A \to B$ and also that the cardinality of $\mathbf{C}(A, A) = 1$. Because B is initial, then there exists a unique morphism $g : B \to A$. But then both $g \circ f$ and id_A are morphisms from A to A. However A is initial, and so we must have $g \circ f = id_A$. Interchanging the roles of A and B in the above argument leads to $f \circ g = id_B$, and so that $A \cong B$ (see next diagram for an illustration of this demonstration).

□

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "id_A"'] & B \arrow[d, "g"] \arrow[dr, "id_B"] & \\
 & A \arrow[r, "f"] & B
\end{tikzcd}$$

Remark 4.1.3 Note that, by the above proposition, in a generic category **C**, we are not able to speak about "the" initial object, but only about "a" initial object, this one being unique up to isomorphism.

Proposition 4.1.4 *If A is a initial object and B is isomorphic to A the B is also initial.*

Proof: Let $f : A \to B$ be an isomorphism with an inverse $g : B \to A$ such that $g \circ f = id_A$ and $f \circ g = id_B$. For an arbitrary object C, we have to show that there exists a unique morphism $h : B \to C$. Since A is initial, we have a unique $k : A \to C$. Therefore, $k \circ g : A \to C$ is another morphism from B to C. For any other $h : B \to C$ we have $h \circ f : A \to C$ which implies $k = h \circ f$ since A is initial. But this implies that

$k \circ g =$
$= (h \circ f) \circ g$ (definition of k)
$= h \circ (f \circ g)$ (associativity)
$= h \circ id_B$ (f is an isomorphism)
$= h$ (identity)

and hence that there exists a unique morphism from B to C.

□

Example 4.1.5 As discussed above, in **Set** the initial object is the empty set.

Example 4.1.6 In a poset $(P, \leq)$ considered as a category, the initial object is an element $0 \in P$ with $0 \leq p$ for all $p \in P$ (i.e., the least element of $(P, \leq)$). Note that in a poset, "isomorphic" means "equal" due to the a(see example 3.2.19), and so we are also able to speak about "the" initial object.

Example 4.1.7 In the product category **Set** $\times$ **Set**, the initial object is the pair $\langle \emptyset, \emptyset \rangle$, while in the category **Set**$^{\rightarrow}$, the initial object is the empty function $\emptyset : \emptyset \to \emptyset$ (see examples 3.4.3.1 and 3.4.3.4).

Definition 4.1.8 *Let $\Sigma = (S, OP)$ be an arbitrary signature and $X = (X_s | s \in S)$ a S-indexed set of variables. The algebra (S_T, OP_T) with*

- $S_T = (T_{OP,s}(X))_{s \in S}$ *as the family of base sets;*

- *$N_T = N$ as the constant for each $N : \rightarrow s$;*
- *$N_T : T_{OP,s}(X) \times \ldots \times T_{OP,s_n}(X) \rightarrow T_{OP,s}(X)$ defined by*

$$N_T(t_1, \ldots, t_n) := N(t_1, \ldots, t_N)$$

for $N : s_1 \ldots s_n \rightarrow s$ and $t_i \in T_{OP,s_i}(X), i = 1, \ldots, n$, as the operations, is called **algebra of terms w.r.t Σ and X**, *or simply* **term algebra**. *We denote this algebra as $T_{OP}(X)$, just as the set of terms, and also T_{OP} for $T_{OP}(\emptyset)$ (the reader may wish to give a quick look at definition 2.3.26).*

Theorem 4.1.9

1. *If $ass : X \rightarrow A$ is an assignment then there is one and only one Σ-homomorphism $ass^\sharp$ which extends ass, i.e., which makes the following diagram (in* $\mathbf{Set}^S$, *which is the category of S-sorted sets and S-sorted functions) commutative ($inc_X = (inc_{X_s} | s \in S)$ is the inclusion of X in $T_{OP}(X)$):*

$$\begin{tikzcd} X \arrow[r, "ass"] \arrow[d, "inc_X"'] & A \\ T_{OP}(X) \arrow[ur, dashed, "ass^\sharp"'] & \end{tikzcd}$$

2. *Let $\Sigma = (S, OP)$ be an arbitrary signature and X a set of variables w.r.t Σ. The algebra $T_{OP}(\emptyset) = T_{OP}$ is initial in the category Σ-***Alg***.*

Proof of 4.1.9.1: We show that the extension $ass^\sharp$ defined in 2.3.27 satisfies the desired properties:

$ass^\sharp$ satisfies, by its definition and the definitions of operations in 4.1.8 the properties of a homomorphism. Moreover, $ass^\sharp$ extends ass by definition, i.e., $ass^\sharp \circ inc_X = ass$. It remains to show that $ass^\sharp$ is the only Σ-homomorphism which extends ass.

Assume there is another Σ-homomorphism $h : T_{OP}(X) \rightarrow A$ which satisfies $h \circ inc_X = ass$. We show by structural induction that $h = ass^\sharp$:

For all constant symbols we have:

$ass^\sharp(N) =$
$= N_A$ ($ass^\sharp$ is a Σ-homomorphism)
$= h(N)$ (h is a Σ-homomorphism)

For all variables we have:

$ass^\sharp(x) =$
$= ass^\sharp(inc_X(x))$ (inc_X is an inclusion)
$= ass(x)$ (by the diagram)

$= h(inc_X(x))$ (assumption)
$= h(x)$ (inc_X is an inclusion)

Assume now that $N(t_1, \ldots, t_N) \in T_{OP}(X)$ and for $i = 1, \ldots, n$ $ass^\sharp(t_i) = h(t_i)$. Then we have:

$ass^\sharp(N(t_1, \ldots, t_n)) =$
$= N_A(ass^\sharp(t_1, \ldots, ass^\sharp(t_n))$ (by definition of $ass^\sharp$)
$= N_A(h(t_1), \ldots, h(t_n))$ (inductive hypothesis)
$= h(t_1, \ldots, t_n)$ (h is a Σ-homomorphism)

Proof of 4.1.9.2: Since $T_{OP} = T_{OP}(\emptyset)$ and $\emptyset$ is the only assignment from $\emptyset$ into A, it follows from 4.1.9.1 that there is exactly one homomorphism $\emptyset^\sharp : T_{OP} \to A$. Moreover this (unique) homomorphism coincides with the evaluation *eval* defined in 2.3.27, since *eval* is a Σ-homomorphism.

□

Definition 4.1.10 *Given a specification $SPEC = (S, OP, E)$, the relation $\equiv$ on ground terms defined for all $t_1, t_2 \in T_{OP}$ by*

$t_1 \equiv t_2$ if and only if $eval_A(t_1) = eval_A(t_2)$ for all $SPEC$ algebras A

is called **congruence on ground terms**.

Definition 4.1.11 *Given a specification $SPEC = (S, OP, E)$ the* **quotient term algebra** *$T_{SPEC} = ((Q_s)_{s \in S}, (N_Q)_{N \in OP})$ is defined by:*

1. *For each $s \in S$ we have a base set $Q_s = \{[t] | t \in T_{OP,s}\}$, where the congruence class $[t]$ is defined by $[t] = \{t' | t \equiv t'\}$.*
2. *For each constant symbol $N :\to s$ in OP the constant N_Q is the congruence class generated by N, i.e., $N_Q = [N]$.*
3. *For each operation symbol $N : s_1 \ldots s_n \to s$ in OP, the operation $N_Q : Q_{s_1} \times \ldots \times Q_{s_n} \to Q_s$ is defined by*
$$N_Q([t_1], \ldots, [t_n]) = [N(t_1, \ldots, t_n)]$$
for all terms t_i of sort s_i and all $i = 1, \ldots, n$.

Theorem 4.1.12

1. *The quotient term algebra T_{SPEC} is a initial object in* **Cat**$(SPEC)$.
2. *The quotient term algebra T_{SPEC} is a $SPEC$-algebra.*

Proof: See [EM85], chapter 2.

Example 4.1.13 (The quotient term algebra $T_{\underline{\underline{string}}}$). First, we construct the sets of terms $T_{alph} = T_{OP,alphabet}$ and $T_{string} = T_{OP,string}$ of the signature $\Sigma = (S, OP)$ from the specification $\underline{\underline{string}}$ in 2.3.31. T_{alph} and T_{string} are inductively defined by (see 2.3.26).

1. $K_1, \ldots, K_n \in T_{alph}$
   $empty \in T_{string}$

2. For all $t_a \in T_{alph}$ and $t_{s_1}, t_{s_2} \in T_{string}$ we have

   $make(t_a), concat(t_{s1}, t_{s2}), ladd(t_a, t_{s1}), radd(t_{s1}, t_a) \in T_{string}$.

These are the base sets from the following $\underline{\underline{string}}$-algebra $T_{\underline{\underline{string}}}$, which is a quotient term algebra for the specification $\underline{\underline{string}}$.

$$\begin{aligned} T_{\underline{\underline{string}}} = (&Q_{alph}, Q_{string}, K_{1_Q}, \ldots, K_{n_Q},\\ &empty_Q, make_Q, concat_Q, ladd_Q, radd_Q) \end{aligned}$$

with

$$\begin{aligned}
&K_{i_Q} = [K_i] \text{ for } i = 1, \ldots, n\\
&empty_Q = [empty]\\
&make_Q : Q_{alph} \to Q_{string}\\
&\qquad make_Q([t_a]) = [make(t_a)]\\
&concat_Q : Q_{string} \times Q_{string} \to Q_{string}\\
&\qquad concat_Q([t_{s1}], [t_{s2}]) = [concat(t_{s1}, t_{s2})]\\
&ladd_Q : Q_{alph} \times Q_{string} \to Q_{string}\\
&\qquad ladd_Q([t_a], [t_{s1}]) = [ladd(t_a, t_{s1})]\\
&radd_Q : Q_{string} \times Q_{alph} \to Q_{string}\\
&\qquad ladd_Q([t_{s1}], [t_a]) = [ladd(t_{s1}, t_a)]
\end{aligned}$$

Note that, in order to verify that $T_{\underline{\underline{string}}}$ is a $\underline{\underline{string}}$-algebra, we have to show that the operations are well defined and that they satisfy the equations in $\underline{\underline{string}}$ or by using directly 4.1.12.2.

Proposition 4.1.14 *$A_\emptyset = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ is the (unique) initial object in* **Aut**.

Proof: Let A' be any finite automata and consider the following diagram:

$$\begin{tikzcd}
\emptyset \arrow[d, "\emptyset"'] \arrow[dr, phantom, "="] & \emptyset \times \emptyset \arrow[l, "\emptyset"'] \arrow[r, "\emptyset"] \arrow[d, "\emptyset \times \emptyset = \emptyset"] \arrow[dr, phantom, "="] & \emptyset \arrow[d, "\emptyset"] \\
O' & I' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] & S'
\end{tikzcd}$$

To see that it commutes, note that $\emptyset \circ \emptyset$ and $\delta' \circ \emptyset$ are both morphisms from $\emptyset$ to S'. However $\emptyset$ is the (unique) initial object in **Set** and hence we must have $\delta' \circ \emptyset = \emptyset \circ \emptyset$. With a similar argument it can be shown that $\emptyset \circ \emptyset = \lambda' \circ \emptyset$ and thus that $f = (\emptyset, \emptyset, \emptyset) : A_\emptyset \to A'$ is unique for every A' in **Aut**.

□

4.2 Terminal objects

By reversing the direction of the morphisms in the definition of initial object, we have the following (dual) notion:

Definition 4.2.1 *Let* $\mathbf{C}$ *be a category. An object* 1 *is a* **terminal** *object in* $\mathbf{C}$*, if for all* $\mathbf{C}$*-objects* A*, there is only one morphism* $f \in \mathbf{C}(A, 1)$*.*

Proposition 4.2.2 *If* A *and* B *are terminal objects in a category* $\mathbf{C}$*, then they are isomorphic.*

Proof 1: Since A is terminal, then there exists a unique morphism $f : B \to A$. Because B is terminal, then there exists a unique morphism $g : A \to B$. But then, both $g \circ f$ and id_B are morphisms from B to B, and since B is terminal, we must have $g \circ f = id_B$. Similarly, it can be shown that $f \circ g = id_A$, and so that $A \cong B$ (see next diagram).

$$
\begin{tikzcd}
A & B \arrow[l, "f"'] & \\
 & A \arrow[u, "g"] \arrow[ul, "id_A"] \arrow[ul, phantom, "=", near end, shift left=3] & B \arrow[l, "f"] \arrow[ul, "id_B"'] \arrow[l, phantom, "=", shift right=3]
\end{tikzcd}
$$

Proof 2: Note that the concept of a terminal object is dual to the concept of initial object. Now let $P_{\mathbf{C}}(A, B)$ be the statement "If A and B are initial objects in a category $\mathbf{C}$ then they are isomorphic" and $P^{op}(\mathbf{C})(A, B)$ (the dual statement) be the statement "If A and B are terminal objects in a category $\mathbf{C}$, then they are isomorphic". However, by the duality principle, we have that if $P(\mathbf{C})(A, B)$ holds then $P^{op}(\mathbf{C})(A, B)$ also holds. Thus, Proof 1 was actually not necessary.

□

From now on, we make heavy use of this most valuable principle. The symbol $\boxed{\text{D}}$ will be used to indicate that no proof is needed, since the dual result has been stated and proved at an earlier point

Example 4.2.3 In **Set**, the terminal objects are the singletons, i.e., the one-element sets $\{x\}$, since for all sets A, there exists only one function from A to $\{x\}$, i.e., the one defined by

$f(a) = x$ for all $a \in A$. Therefore, **Set** has infinitely many terminal objects, and they are all isomorphic.

Example 4.2.4 In a poset $(P, \leq)$ considered as a category, a terminal object is an element $1 \in P$ with $1 \geq p$ for all $p \in P$ (i.e., the greatest element of $(P, \leq)$). Again, in the case of posets, we can speak about "the" terminal object.

Example 4.2.5 In the product category $\mathbf{Set} \times \mathbf{Set}$, a terminal object is a pair of singletons sets, say $\langle \{x\}, \{y\} \rangle$, while in the category $\mathbf{Set}^{\rightarrow}$, a terminal object is a function $f : \{x\} \rightarrow \{y\}$ between two singletons sets (see examples 3.4.3.1 and 3.4.3.4).

Example 4.2.6 In the categories **Grp** and **Mon**, a terminal object is the one-element monoid $(\{e\}, \oplus, e)$, and so each of these categories has infinitely many terminal objects.

Proposition 4.2.7 *In the category* **Aut**, *the finite automata with singleton base sets I, O and S are terminal objects.*

Proof: Let A' be any finite automata and consider the following diagram:

$$\begin{tikzcd}
O' \arrow[d, "f_O"'] & I' \times S' \arrow[l, "\delta'"'] \arrow[r, "\delta'"] \arrow[d, "f_I \times f_S"] \arrow[dl, phantom, "="] \arrow[dr, phantom, "="] & S' \arrow[d, "f_S"] \\
\{y\} & \{x\} \times \{s\} \arrow[l, "\lambda"'] \arrow[r, "\delta"] & \{s\}
\end{tikzcd}$$

To see that it commutes note that $f_S \circ \delta'$ e $\delta \circ (f_I \times f_S)$ are both morphisms from $I' \times S'$ to $\{s\}$. However, $\{s\}$ is terminal in **Set** and so we must have $\delta \circ (f_I \times f_S) = f_S \circ \delta'$. A similar argument shows that $\lambda \circ (f_I \times f_S) = f_O \circ \lambda'$. Thus for every A' in **Aut** there is only one morphism $f : A' \rightarrow (\{i\}, \{o\}, \{s\}, \delta, \lambda)$.

Example 4.2.8 The terminal object of Σ-Alg is the trivial **unit** algebra $U(\Sigma)$, where every carrier set consists of one element and the operations are trivially defined.

Example 4.2.9 We can recover the general concept of an "element" of an object A in a category $\mathbf{C}$ with a terminal object by using morphisms $p : 1 \rightarrow A$, where 1 is the terminal object of the category, i.e, the establishment of a bijection between A and $\mathbf{C}(1, A)$, where the later is the set of all morphisms from the terminal object to A. Such morphisms are called **points** (also called **variables** or **global elements**) of A. In a category as **Set** there exists an exact correspondence between the elements of A (in the usual sense), and the points of A in the categorical sense. In **Set**, the unique object for which there exists no morphism $f : 1 \rightarrow A$ is the empty set. More generally, we say that an object A of a category with a terminal object is **non-empty** if there exists a morphism $p : 1 \rightarrow A$.

Example 4.2.10 Using the concept of **global element** in the previous example and knowing that **Aut** has a terminal object, we can now express that an automata homomorphism must preserve the initial state by the commutativity of the following diagram:

$$
\begin{tikzcd}
O \arrow[dd, "f_O"'] \arrow[ddr, phantom, "="] & I \times S \arrow[l, "\lambda"'] \arrow[r, "\delta"] \arrow[dd, "f_I \times f_S"] \arrow[ddr, phantom, "="] & S \arrow[dd, "f_S"] & \\
 & & & 1 \arrow[ul, "s_0"'] \arrow[dl, dotted, "s'_0"] \\
O' & I' \times S' \arrow[l, "\lambda'"'] \arrow[r, "\delta'"] & S' &
\end{tikzcd}
$$

Now consider the category **Pfn**. There exists only one partial function from $\emptyset$ to A where A is any set, i.e., the empty function $\emptyset : \emptyset \multimap A$. Moreover, there exists also only one partial function from A to $\emptyset$ for every set A, i.e., the partial function totally undefined $\emptyset : A \multimap \emptyset$. This means that $\emptyset$ is both initial and terminal in **Pfn**. Generalizing this for any category we have the following definition:

Definition 4.2.11 *An object A in a category* **C** *is called a* **zero object** *provided that it is both an initial and a terminal object.*

Remark 4.2.12 Note that, since "terminal object" is dual to "initial object", the notion of zero object is self-dual, i.e., A is a zero object in **C** if and only if A is a zero object in $\mathbf{C}^{op}$.

Example 4.2.13 **Set**, **Poset**, Σ-Alg and **Cat**$(SPEC)$ don't have zero objects.

Example 4.2.14 **Grp** and **Mon** both have zero objects, i.e., the one-element monoid $(\{e\}, \oplus, e)$. As discussed above $\emptyset$ is a zero object in **Pfn**.

Table 4.1 presents examples of initial, terminal and zero objects in some categories presented so far.

4.3 Products

We come now to the problem of giving a characterization, using morphisms, of the product set

$$A \times B = \{\langle x, y\rangle : x \in A \wedge y \in B\}$$

of two sets A and B.

Associated with A and B are two special functions, the *projections*

$$p_A : A \times B \to A,$$

CATEGORY	INITIAL	TERMINAL	ZERO
Set	$\emptyset$	$\{x\}$	—
Pfn	$\emptyset$	$\emptyset$	$\emptyset$
$\mathbf{C}(P, \leq)$	minimum element	maximum element	—
Grp	$(\{e\}, \oplus, e)$	$(\{e\}, \oplus, e)$	$(\{e\}, \oplus, e)$
Mon	$(\{e\}, \oplus, e)$	$(\{e\}, \oplus, e)$	$(\{e\}, \oplus, e)$
Σ-**Alg**	T_{OP}	$U(\Sigma)$	—
Aut	$(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$	$(\{i\}, \{o\}, \{s\}, \delta, \lambda)$	—
Graph	$E = \emptyset$, $V = \emptyset$, $s, t : \emptyset \to \emptyset$	$E = \{x\}$, $V = \{x\}$, $s, t : \{x\} \to \{x\}$	—

Table 4.1: Initial, terminal and zero objects in examples of categories

$$p_B : A \times B \to B$$

represented by the rules

$$p_A(\langle x, y \rangle) = x$$

$$p_B(\langle x, y \rangle) = y$$

Now suppose we are given some other set C with a pair of maps $f : C \to A$ and $g : C \to B$. Then we define $p : C \to A \times B$ by the rule $p(c) = \langle f(c), g(c) \rangle$. So we have $p_A(p(c)) = f(c)$ and $p_B(p(c)) = g(c)$ for all $c \in C$, such that the following diagram commutes:

$$\begin{tikzcd} & C \arrow[dl, dotted, "f"'] \arrow[d, "p"] \arrow[dr, "g"] & \\ A & A \times B \arrow[l, "p_A"'] \arrow[r, "p_B"] & B \end{tikzcd}$$

Moreover, p as defined is the only morphism that can make the diagram commute. For if $p(c) = \langle y, z \rangle$, then simply knowing that $p_A \circ p = f$, we have that $p_A(p) = f$ i.e., $y = f$. Similarly if $p_B \circ p = g$, then we must have $z = g$.

The observations just made motivate the following

Definition 4.3.1 *A* **product** *of two objects A and B in a category $\mathbf{C}$, is a $\mathbf{C}$-object $A \times B$, together with two projection morphisms $\pi_A \in \mathbf{C}(A \times B, A)$ and $\pi_B \in \mathbf{C}(A \times B, B)$ such that, for any object C and pair of morphisms $f \in \mathbf{C}(C, A)$ and $g \in \mathbf{C}(C, B)$, there exists exactly one morphism $h \in \mathbf{C}(C, A \times B)$ such that the following diagram commutes, i.e., such that $\pi_A \circ h = f$ and $\pi_B \circ h = g$.*

$$\begin{tikzcd}
 & C \arrow[dl, dotted, "f"'] \arrow[d, "h"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_A"'] \arrow[r, "\pi_B"] & B
\end{tikzcd}$$

Remarks 4.3.2

1. The morphism h is the **product morphism** of f and g with respect to the projections π_A and π_B, and to express its dependence with respect to f and g, it is normally represented as $\langle f, g \rangle$.

2. Note that, although we usually write $A \times B$ to denote the product of A and B, it is important to remember that the projection morphisms are also part of the definition. Strictly speaking, we should define the product as the tuple $(A \times B, \pi_A, \pi_B)$.

3. Note that, in the above definition, we are talking about products *within* a category and not about products *between* categories, as in example 3.4.3.1.

Proposition 4.3.3 *If (D, p, q) and $(A \times B, \pi_A, \pi_B)$ are products of A and B in a category* **C***, then they are isomorphic, i.e., products are unique up to isomorphism.*

Proof: Observe the next diagram.

$$\begin{tikzcd}
 & D \arrow[dl, dotted, "p"'] \arrow[d, "h_1"] \arrow[dr, "q"] & \\
A & A \times B \arrow[l, "\pi_A"'] \arrow[r, "\pi_B"] \arrow[d, "h_2"] & B \\
 & D \arrow[ul, "p"] \arrow[ur, dotted, "q"'] &
\end{tikzcd}$$

Because $A \times B$ is a product of A and B, the upper triangle commutes. Because D is a product of A and B, the bottom triangle commutes. Now, since D is a product of A and B, there can be only one morphism $i : D \to D$ such that the following diagram commutes.

$$\begin{tikzcd}
 & D \arrow[dl, dotted, "p"'] \arrow[d, "i"] \arrow[dr, "q"] & \\
A & D \arrow[l, "p"'] \arrow[r, "q"] & B
\end{tikzcd}$$

Putting $i = h_2 \circ h_1 : D \to D$ we have that

$$\begin{aligned}
& p \circ (h_2 \circ h_1) = \\
& = (p \circ h_2) \circ h_1 && \text{(associativity)} \\
& = \pi_A \circ h_1 && \text{(D is a product)} \\
& = p && \text{($A \times B$ is a product)}
\end{aligned}$$

$$\begin{aligned}
& q \circ (h_2 \circ h_1) = \\
& = (q \circ h_2) \circ h_1 && \text{(associativity)} \\
& = \pi_B \circ h_1 && \text{(D is a product)} \\
& = q && \text{($A \times B$ is a product)}
\end{aligned}$$

which shows that $i = h_2 \circ h_1 : D \to D$ makes the diagram commute. However, $i = id_D : D \to D$ also fits since $p \circ id_D = p$ and $q \circ id_D = q$ (by the identity axiom). But as D is a product of A and B, i is unique and so we must have $h_2 \circ h_1 = id_D$. Interchanging the roles of $A \times B$ and D in this argument leads to $h_1 \circ h_2 = id_{A \times B}$ and hence that $D \cong A \times B$.

□

Proposition 4.3.4 *As a consequence of the definition of product we have that* $\langle \pi_A \circ h, \pi_B \circ h \rangle = h$.

Proof: Consider the following diagram:

$$\begin{tikzcd}
 & C \arrow[dl, dashed, "\pi_A \circ h"'] \arrow[d, "s"] \arrow[dr, "\pi_B \circ h"] & \\
A & A \times B \arrow[l, "\pi_A"'] \arrow[r, "\pi_B"] & B
\end{tikzcd}$$

According to the definition of a product, there exists a unique $s : C \to A \times B$ such that the above diagram commutes. Putting $s = h$ we have, trivially, $\pi_A \circ h = \pi_A \circ h$ and $\pi_B \circ h = \pi_B \circ h$. Now, $s = \langle \pi_A \circ h, \pi_B \circ h \rangle$ also does the job since$\pi_A \circ \langle \pi_A \circ h, \pi_B \circ h \rangle = \pi_A \circ h$ and $\pi_B \circ \langle \pi_A \circ h, \pi_B \circ h \rangle = \pi_B \circ h$. Since s must be unique, we must have $\langle \pi_A \circ h, \pi_B \circ h \rangle = s = h$.

□

Example 4.3.5 As discussed previously, in **Set** the categorical product corresponds to the cartesian product.

Example 4.3.6 In a poset $(P, \leq_P)$ considered as a category, for $p, q \in P$, the product of p and q when it exists, is defined by the properties:

- $p \times q \leq p$, $p \times q \leq q$, i.e., $p \times q$ is a lower bound of p and q.

- If $c \leq p$ and $c \leq q$ then $c \leq p \times q$, i.e., $p \times q$ is the greatest lower bound (g.l.b) of p and q.

Here, the fact that $c \leq_P$ is unique is not important, since in a poset there exists at most one morphism between two objects. For an illustration of this example, see the following diagram:

$$\begin{tikzcd}
 & c \arrow[dl, dashed, "\leq_P"'] \arrow[d, "\leq_P"] \arrow[dr, "\leq_P"] & \\
p & g.l.b(p,q) \arrow[l, "\geq_P"] \arrow[r, "\leq_P"'] & q
\end{tikzcd}$$

Example 4.3.7 In the category **Pfn**, the categorical product is a little bit more elaborated than in **Set**, since we have to consider the case of undefinedness for the functions involved in the construction. So we have

$A \times B = ((A \times B) + A + B, \pi_1, \pi_2)$

$\pi_1 : (A \times B) + A + B \to A$

$$\pi_1(z) = \begin{cases} a & \text{if } z = \langle a, b \rangle \in A \times B \\ a & \text{if } z = a \in A \\ \text{n. def} & \text{otherwise} \end{cases}$$

$\pi_2 : (A \times B) + A + B \to B$

$$\pi_2(z) = \begin{cases} b & \text{if } z = \langle a, b \rangle \in A \times B \\ b & \text{if } z = b \in B \\ \text{n. def} & \text{otherwise} \end{cases}$$

$h : C \to (A \times B) + A + B$

$$h = \begin{cases} \langle f, g \rangle(c) & \text{if } c \in dom(f) \wedge c \in dom(g) \\ f(c) & \text{if } c \in dom(f) \wedge c \notin dom(g) \\ g(c) & \text{if } c \notin dom(f) \wedge c \in dom(g) \\ \text{n. def} & \text{otherwise} \end{cases}$$

To see that this definition really works, note that if $f(c)$ and $g(c)$ are defined, then $z = \langle f(c), g(c) \rangle \in A \times B$ and so this case coincides with the cartesian product in **Set**. If only $f(c)$ is defined, we have that $z = f(c) \in A$, and so that $\pi_1(f(c)) = f(c)$. In the same way, if only $g(c)$ is defined, then we have that $z = g(c) \in B$ and $\pi_2(g(c)) = g(c)$. It is easy to see that h is really the unique function with this property.

Example 4.3.8 In the category **Grp**, the product of two objects it is the standard direct product of two groups, with the binary operation defined componentwise on the product set of the two groups. Therefore, if $(G, \oplus, {}^{-1}_{G}, e_G)$ and $(H, \otimes, {}^{-1}_{H}, e_H)$ are two groups, then the respective group product is the algebraic structure $(G \times H, \odot, {}^{-1}_{G\times H}, e_{G\times H})$, where the binary operation $\odot$ is defined by

$$\langle g_1, h_1\rangle \odot \langle g_2, h_2\rangle = \langle g_1 \oplus g_2, h_1 \otimes h_2\rangle$$

for all $\langle g_1, h_1\rangle$, $\langle g_2, h_2\rangle \in G \times H$ (the operations ${}^{-1}_{G\times H}$ and $e_{G\times H}$ are defined similarly). The projections are defined as in **Set** and it is easy to see that $p_G : G\times H \to G$ and $p_H : G\times H \to H$ are group homomorphisms. For example, we have that

$$
\begin{aligned}
& p_G(\langle g_1, h_1\rangle \odot \langle g_2, h_2\rangle) = \\
&= p_G(\langle g_1 \oplus g_2, h_1 \otimes h_2\rangle) && \text{(definition of } \odot \text{ in } G \times H) \\
&= g_1 \oplus g_2 && \text{(definition of } p_G) \\
&= p_G(\langle g_1, h_1\rangle) \oplus p_G(\langle g_2, h_2\rangle) && \text{(definition of } p_G)
\end{aligned}
$$

Example 4.3.9 In the category **Poset**, the product of the posets $(P, \leq_P)$ and $(Q, \leq_Q)$ is the cartesian product $P \times Q$ of the base sets P and Q, with the order $\leq_{P\times Q}$ given by $(x, y) \leq_{P\times Q} (x', y')$ if and only if $x \leq_P x' \wedge y \leq_Q y'$. The projections are defined as in **Set**, and it is easy to see that they are monotonic functions.

The idea of a product between two objects can be generalized for a family of objects. More precisely:

Definition 4.3.10 Let **C** be a category and $(A_j | j \in J)$ a family of objects of **C**. A **product** of $(A_j | j \in J)$ is a tuple $(\times A_j, (\pi_{A_j} | j \in J))$, where $\times A_j$ is an object of **C** and for each $j \in J$, $\pi_{A_j} : \times A_j \to A_j$ is a **C**-morphism, all subject to the following property: given $(C, (f_j | j \in J))$ with C a **C**-object and $f_j : C \to A_j$ **C**-morphisms, there exists only one morphism $h : C \to \times A_j$, such that $\pi_{A_j} \circ h = f_j$ for all $j \in J$ as shown by the following diagram:

$$
\begin{tikzcd}
C \arrow[d, "h"'] \arrow[dr, "f_j"] \arrow[dr, phantom, "=", shift right=3] & \\
\times A_j \arrow[r, "\pi_{A_j}"] & A_j
\end{tikzcd}
$$

Example 4.3.11 If $J = \{j\}$, i.e., J is a singleton set, then (A_J, id_{A_j}) is a product (see next diagram).

$$
\begin{tikzcd}
C \arrow[d, "f_j"'] \arrow[dr, "f_j"] \arrow[dr, phantom, "=" near start, shift right=3] & \\
A_j \arrow[r, "id_{A_j}"] & A_j
\end{tikzcd}
$$

In case $J = \emptyset$, then the product corresponds to a terminal object (see chapter 5 on general limit constructions for a detailed discussion).

Definition 4.3.12 *Let* **C** *be a category. If the product objects $A \times C$ and $B \times D$ exist, then for each pair of morphisms $f : A \to B$, and $g : C \to D$, the* **product morphism** *$f \times g : A \times C \to B \times D$ is the morphism $\langle f \circ \pi_A, g \circ \pi_C \rangle : A \times C \to B \times D$ (see next diagram).*

$$
\begin{tikzcd}
 & C \arrow[r, "g"] & D \\
A \times C \arrow[ur, dashed, "\pi_C"] \arrow[rr, "\langle f \circ \pi_A, g \circ \pi_C \rangle"] \arrow[dr, "\pi_A"'] & & B \times D \arrow[u, "\pi_D"'] \arrow[d, "\pi_B"] \\
 & A \arrow[r, "f"] & B
\end{tikzcd}
$$

Proposition 4.3.13 $id_A \times id_B = id_{A \times B}$.

Proof: Consider the next diagram:

$$
\begin{tikzcd}
 & B \arrow[r, "id_B"] & B \\
A \times B \arrow[ur, dashed, "\pi_B"] \arrow[rr, "s"] \arrow[dr, "\pi_A"'] & & A \times B \arrow[u, "\pi_B"'] \arrow[d, "\pi_A"] \\
 & A \arrow[r, "id_A"] & A
\end{tikzcd}
$$

putting $s = id_A \times id_B$ we have that

$\pi_A \circ (id_A \times id_B) =$

$= \pi_A \circ \langle id_A \circ \pi_A, id_B \circ \pi_B \rangle$ (definition of $id_A \times id_B$)

$= id_A \circ \pi_A$ (definition of π_A)

$= \pi_A$ (identity)

Similarly, it can be shown that $\pi_B \circ (id_A \times id_B) = \pi_B$, and hence the commutativity of the diagram. On the other hand, $s = id_{A\times B}$ also does the job, since that $\pi_A \circ id_{A\times B} = \pi_A = id_A \circ \pi_A$ and $\pi_B \circ id_{A\times B} = \pi_B = id_B \circ \pi_B$. However, since $A \times B$ is a product, s is the unique (product) morphism, and so we must have $id_A \times id_B = id_{A\times B}$.

□

Remark 4.3.14 In fact, the above property is just states the preservation of identities, if we consider $\times$ as a functor from $\mathbf{C} \times \mathbf{C}$ to $\mathbf{C}$ (where $\mathbf{C}$ must be a category with all binary products), a fact that is proved in 6.1.1.19.

Proposition 4.3.15

1. *$(f \times h) \circ \langle g, k\rangle = \langle f \circ g, h \circ k\rangle$.*
2. *$\langle g, k\rangle = \langle g \times id\rangle \circ \langle id, k\rangle$.*

1.**Proof:** Consider the next diagram:

$$\begin{tikzcd}
 & C \arrow[r, "h"] & D \\
E \arrow[ur, dashed, "k"] \arrow[r, "\langle g,k\rangle"] \arrow[dr, "g"'] & A \times C \arrow[u, "\pi_c"] \arrow[r, "f\times h"] \arrow[d, "\pi_A"] & B \times D \arrow[u, "\pi_D"] \arrow[d, "\pi_B"] \\
 & A \arrow[r, "f"] & B
\end{tikzcd}$$

Since $A \times C$ is a product, $\langle g, k\rangle$ is the unique morphism that makes the left triangle commute. Because $B \times D$ is a product, $f \times h = \langle f \circ \pi_A, h \circ \pi_C\rangle$ is the unique product morphism which makes the right rectangle commute. To see that $\langle f \circ \pi_A, g \circ \pi_C\rangle \circ \langle g, k\rangle$ makes the whole diagram commute note that

$\pi_B \circ (\langle f \circ \pi_A, h \circ \pi_C\rangle \circ \langle g, k\rangle) =$

$= (\pi_B \circ \langle f \circ \pi_A, h \circ \pi_C\rangle) \circ \langle g, k\rangle$ (associativity)

$= (f \circ \pi_A) \circ \langle g, k\rangle$ (definition of π_B)

$= f \circ (\pi_A \circ \langle g, k \rangle)$ (associativity)
$= f \circ g$ (definition of π_A)

Similarly, it can be shown that $\pi_A \circ (\langle f \circ \pi_A, g \circ \pi_C \rangle \circ \langle g, k \rangle) = h \circ k$. However, $\langle f \circ g, h \circ k \rangle$ also does the job, since $\pi_B \circ \langle f \circ g, h \circ k \rangle = f \circ g$ and $\pi_D \circ \langle f \circ g, h \circ k \rangle = h \circ k$. But by the product property of $B \times D$, this morphism has to be unique, and so we must have $(f \times h) \circ \langle g, k \rangle = \langle f \circ g, h \circ k \rangle$.

2. Proof: According to the previous diagram, we have that $\langle g \times id \rangle \circ \langle id, k \rangle = \langle g \times id_C \rangle \circ \langle id_E, k \rangle$. Now

$\langle g \times id_C \rangle \circ \langle id_E, k \rangle =$
$= \langle g \circ id_E, id_C \circ k \rangle$ (previous demonstration)
$= \langle g, k \rangle$ (identity)

□

Proposition 4.3.16 $\langle f \circ h, g \circ h \rangle = \langle f, g \rangle \circ h.$

Proof: Consider the two diagrams below:

Putting $s = \langle f \circ h, g \circ h \rangle$ we have that $\pi_A \circ \langle f \circ h, g \circ h \rangle = f \circ h$ and $\pi_B \circ \langle f \circ h, g \circ h \rangle = g \circ h$, and hence the right diagram commutes. Now $s = \langle f, g \rangle \circ h$ also does the job since $\pi_A \circ \langle f, g \rangle \circ h = f \circ h$ and $\pi_B \circ \langle f, g \rangle \circ h = g \circ h$. However, by definition of a product, s must be unique and thus we have that $\langle f \circ h, g \circ h \rangle = s = \langle f, g \rangle \circ h$.

$$\begin{tikzcd}
 & D \arrow[d, "h"] & \\
 & C \arrow[dl, dashed, "f"'] \arrow[d, "\langle f, g \rangle"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B
\end{tikzcd}
\qquad
\begin{tikzcd}
 & C \arrow[dl, dashed, "f \circ h"'] \arrow[d, "s"] \arrow[dr, "g \circ h"] & \\
A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B
\end{tikzcd}$$

□

Proposition 4.3.17 $(f \times h) \circ (g \times k) = (f \circ g) \times (h \circ k)$

Proof: Consider the following diagram.

$$
\begin{tikzcd}
E' \arrow[r, "k"] & C \arrow[r, "h"] & D \\
E \times E' \arrow[u, "\pi_{E'}"] \arrow[d, "\pi_E"'] \arrow[r, "g \times k"] & A \times C \arrow[u, "\pi_C"'] \arrow[d, "\pi_A"] \arrow[r, "f \times h"] & B \times D \arrow[u, "\pi_D"'] \arrow[d, "\pi_B"] \\
E \arrow[r, "g"] & A \arrow[r, "f"] & B
\end{tikzcd}
$$

Because $A \times C$ is a product, $g \times k$ is the unique (product) morphism that makes the left rectangle commute. Because $B \times D$ is a product, $f \times h$ is the unique (product) morphism that makes the right rectangle commute. To see that $(f \times h) \circ (g \times k)$ makes the whole diagram commute, note that

$$
\begin{aligned}
&\pi_B \circ ((f \times h)(g \times k)) = \\
&= (\pi_B \circ (f \times h)) \circ (g \times k) && \text{(associativity)} \\
&= (\pi_B \circ \langle f \circ \pi_A, h \circ \pi_C \rangle) \circ (g \times k) && \text{(definition of } f \times h) \\
&= (f \circ \pi_A) \circ (g \times k) && \text{(definition of } \pi_B) \\
&= f \circ \pi_A) \circ (g \times k)) && \text{(associativity)} \\
&= f \circ (\pi_A \circ \langle g \circ \pi_E, k \circ \pi_{E'} \rangle) && \text{(definition of } g \times k) \\
&= f \circ g \circ \pi_E && \text{(definition of } \pi_A)
\end{aligned}
$$

Similarly, it can be shown that $\pi_D \circ ((f \times h) \circ (g \times k)) = h \circ k \circ \pi_{E'}$. However, $(f \circ g) \times (h \circ k)$ also does the job, since we have that

$$
\begin{aligned}
&\pi_B \circ ((f \circ g) \times (h \circ k)) = \\
&= \pi_B \circ \langle f \circ g \circ \pi_E, h \circ k \circ \pi_{E'} \rangle && \text{(definition of } (f \circ g) \times (h \circ k)) \\
&= f \circ g \circ \pi_E && \text{(definition of } \pi_B)
\end{aligned}
$$

In the same way, we have that $\pi_D \circ ((f \circ g) \times (h \circ k)) = h \circ k \circ \pi_{E'}$. However, since $B \times D$ is a product, we have, by uniqueness, $(f \times h) \circ (g \times k) = (f \circ g) \times (h \circ k)$.

□

Proposition 4.3.18 *If* **C** *be a category with a terminal object then* $A \cong 1 \times A \cong A \times 1$.

Proof: Consider the following diagram.

$$
\begin{tikzcd}
 & A \arrow[dl, dashed, "id_A"'] \arrow[d, "h_1"] \arrow[dr, "t_A"] & \\
A & A \times 1 \arrow[l, "\pi_A"'] \arrow[r, "t_{A\times 1}"] \arrow[d, "h_2"] & 1 \\
 & A \arrow[ul, "id_A"] \arrow[ur, dashed, "t_A"'] &
\end{tikzcd}
$$

We put $h_1 = \langle id_A, t_A \rangle$, where t_A is the unique morphism from A to 1. Now, $\pi_A \circ \langle id_A, t_A \rangle = id_A$, and the top left triangle commutes. Also, $t_{A\times 1} \circ \langle id_A, t_A \rangle = t_A$, since both t_A and $t_{A\times 1} \circ \langle id_A, t_A \rangle$ are morphisms from A to 1 and according to the definition of terminal object, there can be one such morphism, and so the top right triangle commutes and it is clear that $\langle id_A, t_A \rangle$ is unique, since it is the only possible choice. On the other hand, putting $h_2 = \pi_A$ we have that $id_A \circ \pi_A = \pi_A$ (identity axiom) and so the bottom left triangle commutes. Because 1 is a terminal object, $t_A \circ \pi_A = t_{A\times 1}$. Since all the triangles commute, we have that the whole diagram commutes. Now consider the following diagram:

$$
\begin{tikzcd}
 & A \arrow[dl, dashed, "id_A"'] \arrow[d, "s"] \arrow[dr, "t_A"] & \\
A & A \arrow[l, "id_A"'] \arrow[r, "t_A"] & 1
\end{tikzcd}
$$

Now, considering A as product of one object, we have that there exists only one s such that the diagram commutes. Clearly $s = id_A$ does the job. But, according to what was shown above, $s = \pi_A \circ \langle id_A, t_A \rangle : A \to A$ also does the job. As s must be unique, we have that $\pi_A \circ \langle id_A, t_A \rangle = s = id_A$. Interchanging the roles of A and $A \times 1$ in this argument, we have that $\langle id_A, t_A \rangle \circ \pi_A = id_{A\times 1}$. By 4.3.18 we have that $A \times 1 \cong 1 \times A$, and noting that $\cong$ is an equivalence relation on $Ob(\mathbf{C})$ (proposition 3.2.16), we have the desired result.

□

4.4 Coproducts

The dual notion for a product is a coproduct or sum of objects, which we present in the following

Definition 4.4.1 A **coproduct** of two objects A and B in a category $\mathbf{C}$ is a $\mathbf{C}$-object $A + B$ together with two **injection morphisms** $in_A \in \mathbf{C}(A, A+B)$ and $in_B \in \mathbf{C}(B, A+B)$, such that for any $\mathbf{C}$-object C and pair of morphisms $f \in \mathbf{C}(A, C)$ and $g \in \mathbf{C}(B, C)$, there exists exactly one morphism $h \in \mathbf{C}(A+B, C)$, such that $h \circ in_A = f$ and $h \circ in_B = g$ (see next diagram).

$$\begin{tikzcd}
 & C & \\
A \arrow[ur, dotted, "f"] \arrow[r, "in_A"] & A+B \arrow[u, "h"] & B \arrow[l, "in_B"'] \arrow[ul, "g"']
\end{tikzcd}$$

Remarks 4.4.2

1. The morphism h is the **coproduct morphism** of f and g in relation to the injections in_A and in_B, and to express its dependence on f and g, it is usually represented as $[f, g]$.
2. Note that, although we usually write only $A+B$ to denote the coproduct of A and B, it is important to remember that the injection morphisms does make part of the definition. More rigorously, we should represent the coproduct as the tuple $(A+B, in_A, in_B)$.

Proposition 4.4.3 *If (D, p, q) and $(A+B, in_A)$ are coproducts of A and B in a category $\mathbf{C}$, then they are isomorphic, i.e., coproducts are unique up to isomorphism.*

Proof: $\boxed{\text{D}}$

Example 4.4.4 In **Set**, the categorical coproduct of A_1 and A_2 is the disjoint union of $A_1 + A_2$. First, we form the disjoint union of A_1 and A_2 in the following way:

$$A_1 + A_2 = \{\langle a, A_i\rangle : i = 1, 2, a \in A_i\}$$

The injections $in_{A_i} : A_i \to C$ are defined as

$$in_{A_i}(a) = (a, i)$$

for all $a \in A_i$, $i = 1, 2$.

Now, given the functions $f_i : A_i \to C$, $i = 1, 2$, consider the following diagram:

$$\begin{tikzcd}
 & C & \\
A_1 \arrow[ur, dotted, "f_1"] \arrow[r, "in_{A_1}"] & A_1 + A_2 \arrow[u, "h"] & A_2 \arrow[l, "in_{A_2}"'] \arrow[ul, "f_2"']
\end{tikzcd}$$

The morphism $h : A_1 + A_2 \to C$ is defined as $h(a, i) = f_i(a)$. To see that this diagram really commutes, note that, for $i = 1, 2$, $a \in A$, $h \circ in_{A_i}(a) = h(a, i) = f_i(a)$. Moreover, such h is unique, since if $h_1 \circ in_{A_i} = f_i$ for $i = 1, 2$, then for all $(a, i) \in A_1 + A_2$ we have that

$$
\begin{aligned}
h_1(a, i) &= h_1 \circ in_{A_i}(a) && \text{(definition of } in_{A_i}) \\
&= f_i(a) && \text{(assumption)} \\
&= h(a, i) && \text{(definition of } f_i(a))
\end{aligned}
$$

□

Example 4.4.5 In a poset $(P, \leq_P)$ considered as a category, for $p, q \in P$, a coproduct of p and q, when it exists, is defined by the following properties:

- $p \leq p + q$, $q \leq p + q$, i.e., $p + q$ is a *upper bound* for p and q.
- If $p \leq c$ and $q \leq c$, then $p + q \leq c$, i.e., $p + q$ is less than any other upper bound of p and q.

In other words, $p+q$ is the **least upper bound** $(l.u.b)$ of p and q. The following diagram illustrates this example.

$$
\begin{tikzcd}
& c & \\
p \arrow[ur, dotted, "\leq_P"] \arrow[r, "\leq_P"'] & l.u.b(p,q) \arrow[u, "\leq_P"'] & q \arrow[l, "\geq_P"] \arrow[ul, "\leq_P"']
\end{tikzcd}
$$

Example 4.4.6 In the category **Poset**, the coproduct of two posets $(P_1 \leq_{P_1})$ and $(P_2, \leq_{P_2})$ is the disjoint union $P_1 + P_2$ of the base sets P_1 and P_2, with the order $\leq_{P_1+P_2}$ given by $(x, i) \leq_{P_1+P_2} (x', i)$ if and only if $x \leq_{P_i} x'$. The injections are defined as in **Set**.

The notion of a coproduct can be generalized for a family of objects. More precisely:

Definition 4.4.7 *Let* $\mathbf{C}$ *be a category and* $(A_j|j \in J)$ *a family of objects of* $\mathbf{C}$. *A* ***coproduct*** *of* $(A_j|j \in J)$ *is a tuple* $(+A_j, (in_{A_j}|j \in J))$, *where* $+A_j$ *is an object of* $\mathbf{C}$ *and for each* $j \in J$, $in_{A_j} : +A_j \to C$ *is a* $\mathbf{C}$*-morphism, all subject to the following property: given* $(C, (f_j|j \in J))$, *with* C *a* $\mathbf{C}$*-object and* $f_j : A_j \to C$ $\mathbf{C}$*-morphisms, there exists a unique* $h : +A_j \to C$, *such that* $h \circ in_{A_j} = f_j$ *for all* $j \in J$, *as shown by the following diagram:*

$$
\begin{tikzcd}
C & \\
\times A_j \arrow[u, "h"] & A_j \arrow[l, "in_{A_j}"'] \arrow[ul, "f_j"'] \arrow[ul, phantom, "=", shift right=3ex]
\end{tikzcd}
$$

Definition 4.4.8 *Let* **C** *be a category. If the coproduct objects $A + C$ and $B + D$ exist, then for each pair of morphisms $f : B \to A$ and $g : D \to C$, the* **coproduct morphism** *$f + g : B + D \to A + C$ is the morphism $[in_A \circ f, in_C \circ g]$ (see next diagram).*

$$
\begin{tikzcd}
& C \arrow[dl, dashed, "in_C"'] & D \arrow[l, "g"'] \arrow[d, "in_D"] \\
A + C & & B + D \arrow[ll, "{[in_A \circ f, in_C \circ g]}"'] \\
& A \arrow[ul, "in_A"] & B \arrow[l, "f"'] \arrow[u, "in_B"']
\end{tikzcd}
$$

Proposition 4.4.9 $[g, k] \circ (f + h) = [g \circ f, k \circ h]$.

Proof: $\boxed{\text{D}}$

Proposition 4.4.10 $(g + k) \circ (f + h) = (g \circ f) + (k \circ h)$.

Proof: $\boxed{\text{D}}$

4.5 Equalizers

In **Set**, equalizers can be constructed in the following way:

Given two functions $f, g : A \to B$, we can construct the subset E of A in which f and g are equal, i.e.:

$$E = \{a : a \in A \wedge f(a) = g(a)\}$$

Then, the inclusion function $in_E : E \to A$ equalizes f and g, i.e., $f \circ in_E(a) = f(in_E(a)) = f(a) = g(a) = g(in_E(a)) = g \circ in_E(a)$ for all $a \in E$. Moreover, in_E is a "canonical" equalizer of f and g, since for any other $h : C \to A$, such that $f \circ h = g \circ h$, there exists a unique k such that $in_E \circ k = h$. We define $k(c) = h(c)$ for every $c \in C$. This definition really does the job, since $in_E \circ k(c) = in_E(k(c)) = in_E(h(c)) = h(c)$, and $h(c) \in E$ since $f(h(c)) = g(h(c))$ by assumption (see definition of E above). Now, in case that there exists another $k' : C \to A$ such that $in_E \circ k' = h$, we have $in_E(k'(c)) = h(c) = in_E(k(c))$, and as in_E is injective, we have $k(c) = k'(c)$ for all $c \in C$, i.e., $k = k'$ (see next diagram).

$$
\begin{tikzcd}
E \arrow[r, "in_E"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
C \arrow[u, "k", "="'] \arrow[ur, dashed, "h"'] & &
\end{tikzcd}
$$

This discussion motivates to present the following

Definition 4.5.1 *A morphism $e \in \mathbf{C}(X, A)$ is an equalizer of a pair of morphisms $f \in \mathbf{C}(A, B)$ and $g \in \mathbf{C}(A, B)$ if*

1. $f \circ e = g \circ e$;
2. *whenever there is a morphism $e' \in \mathbf{C}(X', A)$ which satisfies $f \circ e' = g \circ e'$, then there exists a unique morphism $k \in \mathbf{C}(X', X)$ such that $e \circ k = e'$, as shown by the following diagram:*

$$
\begin{tikzcd}
X \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
X' \arrow[u, "k", "="'] \arrow[ur, dashed, "e'"'] & &
\end{tikzcd}
$$

Proposition 4.5.2 *Equalizers are unique up to isomorphism*

Proof: Consider the next diagram:

$$
\begin{tikzcd}
E' \arrow[dr, "e'"] & & \\
E \arrow[u, "k_2", "="'] \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
E' \arrow[u, "k_1", "="'] \arrow[ur, dashed, "e'"'] & &
\end{tikzcd}
$$

Suppose that e and e' are equalizers of f and g. Since e is an equalizer of f and g, there exists a unique $k_1 : E' \to E$ such that $k_1 \circ e = e'$. Because e is an equalizer of f and g, there exists a unique $k_2 : E \to E'$ such that $k_2 \circ e' = e$. Now $k_2 \circ k_1 : E' \to E'$ makes the bigger triangle commute, since we have

$$
\begin{aligned}
& e' \circ (k_2 \circ k_1) = \\
& = (e' \circ k_2) \circ k_1 && \text{(associativity)} \\
& = e \circ k_1 && \text{(e' is an equalizer)} \\
& = e' && \text{(e is an equalizer)}
\end{aligned}
$$

Now $id_{E'} : E' \to E'$ also does the job, since $e' \circ id_{E'} = e'$. However, because e' is an equalizer, we have, by uniqueness, $k_2 \circ k_1 = id_{E'}$. Interchanging the roles of e and e' in the above argument, we have that $k_1 \circ k_2 = id_E$.

$\square$

Proposition 4.5.3 *Every equalizer is a monomorphism.*

Proof: Consider the following diagram:

$$
\begin{tikzcd}
E \arrow[r, "i"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
C \arrow[u, "k"] \arrow[ur, dotted, "h = i \circ j"'] \arrow[u, phantom, shift left=4ex, "="] & &
\end{tikzcd}
$$

Suppose that i equalizes f and g. We must show that, given $l, j : C \to E$, $i \circ l = i \circ j \Rightarrow l = j$. Putting $h = i \circ j$, we have that

$$
\begin{aligned}
& f \circ h = \\
& = f \circ (i \circ j) && \text{(definition of h)} \\
& = (f \circ i) \circ j && \text{(associativity)} \\
& = (g \circ i) \circ j && \text{(i is an equalizer)} \\
& = g \circ (i \circ j) && \text{(associativity)} \\
& = g \circ h && \text{(definition of h)}
\end{aligned}
$$

But then, as i is an equalizer of f and g, there exists a unique $k : C \to E$ with $i \circ k = h$. However, by definition, $i \circ j = h$, and so $k = j$. On the other hand, we have, by assumption, that $i \circ j = i \circ l = h$, and k must be l. So $j = k = l$.

$\square$

Proposition 4.5.4 *Every section is a equalizer.*

Proof: Let $f : A \to B$ be a section such that we have the existence of $g : B \to A$ with $g \circ f = id_A$. Moreover, consider the following diagram:

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, phantom, "="] & B \arrow[r, shift left, "f \circ g"] \arrow[r, shift right, "id_B"'] & B \\
C \arrow[u, "k"] \arrow[ur, dashed, "h"'] & &
\end{tikzcd}$$

First note that $(f \circ g) \circ f = f \circ (g \circ f) = f \circ id_A = f = id_B \circ f$ such that f satisfies the first requirement for being an equalizer. Now we define $h = f \circ g$ such that we have $(f \circ g) \circ (f \circ g) = f \circ (g \circ f) \circ g = f \circ id_A \circ g = f \circ g = id_B \circ (f \circ g)$. This means that h is another candidate for an equalizer of $f \circ g$ and id_B. Therefore we must show the existence of a unique $k : B \to A$ such that $f \circ k = f \circ g$. We define $k = g$ such that the commutativity is trivially defined. To see uniqueness, assume that there is another $k' : B \to A$ such that $k' \circ f = h = k \circ f$. But by 3.3.6, every section is a monomorphism, which implies $k = k'$ as desired. $\square$

4.6 Coequalizers

The dual notion for equalizers is represented in the following

Definition 4.6.1 *A morphism $c \in \mathbf{C}(A, X)$ is a* **coequalizer** *of a pair of morphisms $f \in \mathbf{C}(B, A)$ and $g \in \mathbf{C}(B, A)$ if*

1. *$c \circ f = c \circ g$;*
2. *whenever $c' \in \mathbf{C}(A, X')$ satisfies $c' \circ f = c' \circ g$, then there exists a unique morphism $k \in \mathbf{C}(X, X')$ such that $k \circ c = c'$, as shown in the next diagram.*

$$\begin{tikzcd}
X \arrow[d, "k"'] \arrow[dr, phantom, "="] & A \arrow[l, "c"'] \arrow[dl, dashed, "c'"] & B \arrow[l, shift right, "f"'] \arrow[l, shift left, "g"] \\
X' & &
\end{tikzcd}$$

Definition 4.6.2 *Given a function $f : A \to B$, we define the* **kernel relation** *of f, R_f, as*

$$R_f = \{\langle a_1, a_2 \rangle : a_1, a_2 \in A \wedge f(a_1) = f(a_2)\}$$

Example 4.6.3 In **Set**, coequalizers can be constructed by using equivalence relations. Given two functions $f, g : X \to A$, the intuitive idea is to identify $f(x)$ with $g(x)$ for all $x \in X$. First we construct the relation $R = \{\langle f(x), g(x)\rangle\}$ for all $x \in X$. Next we define R' as being the least equivalence relation which contains R' (i.e., the reflexive, symmetric and transitive closure of R). Then we define the coequalizer $nat : A \to A/R'$ by $nat(a) = [a]$ (also called "natural" mapping). Now, see that $\langle f(x), g(x)\rangle \in R \subseteq R'$ by construction. Therefore, we have that $nat \circ f(x) = nat(f(x)) = [f(x)]_{R'} = [g(x)]_{R'} = nat(g(x)) = nat \circ g(x)$, since $f(x)$ and $g(x)$ are clearly in the same equivalence class. On the other hand, if there exists another morphism $c : A \to B$ such that $c \circ f = c \circ g$, we must have a unique $k : A/R' \to B$ such that $k \circ nat = c$. We define $k([a]) = c(a)$ for this job, since we must have $k \circ nat(a) = k(nat(a)) = k([a]) = c(a)$. However, we have a problem here if k is or not well-defined, and so we must show that $[a_1] = [a_2] \Rightarrow k([a_1]) = k([a_2])$. But according to the definition of k, it is only necessary to show that $[a_1] = [a_2] \Rightarrow c(a_1) = c(a_2)$. Now $[a_1] = [a_2] \Rightarrow \langle a_1, a_2\rangle \in R'$ and $c(a_1) = c(a_2) \Rightarrow \langle a_1, a_2\rangle \in R_c$. Therefore, we need to show that $(a_1, a_2) \in R' \Rightarrow \langle a_1, a_2\rangle \in R_c$, i.e., $R' \subseteq R_c$. Now, by assumption, we have that $c \circ f = c \circ g$, i.e., $c(f(x)) = c(g(x))$ for all $x \in X$. This tell us that $\langle f(x), g(x)\rangle \in R_c$ and hence that $R \subseteq R_c$. Now we have both $R \subseteq R'$ and $R \subseteq R_c$. However, we know by construction that R' is the least equivalence relation that contains R and so we have that $R \subseteq R' \subseteq R_c$ (see next diagram).

$$
\begin{tikzcd}
A/R' \arrow[d, "k"'] \arrow[dr, phantom, "="] & A \arrow[l, "nat"'] \arrow[dl, dashed, "c"] & X \arrow[l, shift right, "f"'] \arrow[l, shift left, "g"] \\
B & &
\end{tikzcd}
$$

Proposition 4.6.4 *Coequalizers are unique up to isomorphism.*

Proof: $\boxed{\text{D}}$

Proposition 4.6.5 *Every coequalizer is an epimorphism.*

Proof: $\boxed{\text{D}}$

Proposition 4.6.6 *Every retraction is a coequalizer.*

Proof: $\boxed{\text{D}}$

4.7 Pullbacks

Definition 4.7.1 *A* **pullback** *of a pair of morphisms $f \in \mathbf{C}(A, C)$ and $g \in \mathbf{C}(B, C)$ is a $\mathbf{C}$-object P and a pair of morphisms $g' \in \mathbf{C}(P, A)$ and $f' \in \mathbf{C}(P, B)$, such that*

1. *$f \circ g' = g \circ f'$;*
2. *for any other $\mathbf{C}$-object X and morphisms $i \in \mathbf{C}(X, A)$, $j \in \mathbf{C}(X, B)$ with $f \circ i = g \circ j$, there exists only one morphism $k : X \to P$ such that $i = g' \circ k$ and $j = f' \circ k$ as shown by the following diagram:*

$$\begin{tikzcd}
 & B \arrow[dr, "g"] & \\
X \arrow[ur, dotted, "j"] \arrow[r, "k"] \arrow[dr, "i"'] \arrow[ur, phantom, "=", pos=0.6, shift right=3ex] \arrow[dr, phantom, "="', pos=0.6, shift left=3ex] & P \arrow[u, "f'"'] \arrow[d, "g'"] \arrow[r, phantom, "="] & C \\
 & A \arrow[ur, dotted, "f"'] &
\end{tikzcd}$$

Proposition 4.7.2 *Pullbacks are unique up to isomorphism.*

Proof: Consider the following diagram:

$$\begin{tikzcd}
 & B \arrow[dr, "g"] & \\
P_2 \arrow[ur, dotted, "j"] \arrow[r, shift left, "k_1"] \arrow[dr, "i"'] & P_1 \arrow[l, shift left, "k_2"] \arrow[u, "f'"'] \arrow[d, "g'"] \arrow[r, phantom, "="] & C \\
 & A \arrow[ur, dotted, "f"'] &
\end{tikzcd}$$

Since P_1 is a pullback of f and g, then there exists a unique $k_1 : P_2 \to P_1$ such that $f' \circ k_1 = j$ and $g' \circ k_1 = i$. Since P_2 is a pullback of f and g then there exists a unique $k_2 : P_1 \to P_2$ such that $j \circ k_2 = f'$ and $i \circ k_2 = g'$. Now consider the following diagram:

$$\begin{tikzcd}
 & B \arrow[dr, "g"] & \\
P_2 \arrow[ur, dotted, "j"] \arrow[r, "s"] \arrow[dr, "i"'] \arrow[ur, phantom, "(1)", pos=0.6, shift right=3ex] \arrow[dr, phantom, "(2)"', pos=0.6, shift left=3ex] & P_2 \arrow[u, "j"'] \arrow[d, "i"] \arrow[r, phantom, "="] & C \\
 & A \arrow[ur, dotted, "f"'] &
\end{tikzcd}$$

Since P_2 is a pullback of f and g, there must be a unique $s : P_2 \to P_2$ such that $s \circ j = j$ and $s \circ i = i$. Putting $s = k_2 \circ k_1$ we have that

$$
\begin{aligned}
& j \circ (k_2 \circ k_1) = && \\
& = (j \circ k_2) \circ k_1 && \text{(associativity)} \\
& = f' \circ k_1 && (P_2 \text{ is a pullback}) \\
& = j && (P_1 \text{ is a pullback})
\end{aligned}
$$

which shows that (1) commutes. In the same way, we can show that (2) commutes. Now, $s = id_{P_2}$ also does the job, since we have $j \circ id_{P_2} = j$ and $i \circ id_{P_2} = i$ and hence, by uniqueness of s we have that $k_2 \circ k_1 = id_{P_2}$. Interchanging the roles of P_1 and P_2 in the argument above we have that $k_1 \circ k_2 = id_{P_1}$.

□

Theorem 4.7.3 *(adapted from [EP72]) Pullbacks can be constructed from products and equalizers, i.e., in any category where each pair of objects has a product and each pair of morphisms (with same domain and codomain) have an equalizer, then it is also the case that each pair of morphisms (with same codomain) have a pullback.*

Proof: Consider the following diagram:

$$
\begin{tikzcd}
 & A \arrow[dr, "f"] & \\
K \arrow[ur, dotted, "q_A"] \arrow[r, "q"] \arrow[dr, "q_B"'] & A \times B \arrow[u, "\pi_A"'] \arrow[d, "\pi_B"] \arrow[r, shift left, "f \circ \pi_A"] \arrow[r, shift right, "g \circ \pi_B"'] & C \\
 & B \arrow[ur, dotted, "g"'] &
\end{tikzcd}
$$

Let $A \times B$, together with the projections π_A and π_B be a product of $A \times B$ and $q : K \to A \times B$ an equalizer of $f \circ \pi_A : A \times B \to C$ and $g \circ \pi_B : A \times B \to C$. We also define $q_A = \pi_A \circ q$ and $q_B = \pi_B \circ q$. We have to show that the following diagram is a pullback.

$$
\begin{tikzcd}
& A \arrow[dr, "f"] & \\
K \arrow[ur, dashed, "q_A"] \arrow[dr, "q_B"'] & & C \\
& B \arrow[ur, dashed, "g"'] &
\end{tikzcd}
$$

1. Since $q : K \to A \times B$ is an equalizer of $f \circ \pi_A$ and $g \circ \pi_B$ we have that

$f \circ q_A =$	
$= f \circ (\pi_A \circ q)$	(definition of q_A)
$= (f \circ \pi_A) \circ q$	(associativity)
$= (g \circ \pi_B) \circ q$	(q is a equalizer)
$= g \circ (\pi_B \circ q)$	(associativity)
$= g \circ q_B$	(definition of q_B)

which shows that the previous diagram commutes.

2. Now let D be an object with morphisms $h : D \to A$ and $k : D \to B$, such that $f \circ h = g \circ k$ (i.e., another candidate for a pullback of f and g). Since $(A \times B, \pi_A, \pi_B)$ is a product, it holds that there exists only one morphism $\langle h, k\rangle : D \to A \times B$ such that $\pi_A \circ \langle h, k\rangle = h$ and $\pi_B \circ \langle h, k\rangle = k$ (see the next diagram).

$$
\begin{tikzcd}
& A \arrow[dr, "f"] & \\
D \arrow[ur, dashed, "h"] \arrow[r, "\langle h, k\rangle"] \arrow[dr, "k"'] & A \times B \arrow[u, "\pi_A"'] \arrow[d, "\pi_B"] & C \\
& B \arrow[ur, dashed, "g"'] &
\end{tikzcd}
$$

Therefore we have:

$(f \circ \pi_A) \circ \langle h, k\rangle =$	
$= f \circ (\pi_A \circ \langle h, k\rangle)$	(associativity)

$$
\begin{aligned}
&= f \circ h && \text{(definition of } \pi_A\text{)}\\
&= g \circ k && \text{(assumption)}\\
&= g \circ (\pi_B \circ \langle h, k\rangle) && \text{(definition of } k\text{)}\\
&= (g \circ \pi_B) \circ \langle h, k\rangle && \text{(associativity)}
\end{aligned}
$$

which shows that $\langle h, k\rangle$ is another candidate for an equalizer of $f \circ \pi_A$ and $g \circ \pi_B$. But then we now that q is an equalizer of $f \circ \pi_A$ and $g \circ \pi_B$, and then it follows that there exists only one $l : D \to K$ such that $q \circ l = \langle h, k\rangle$ (see the next diagram).

$$
\begin{tikzcd}
& B \arrow[dr, "f"] & \\
D \arrow[ur, dashed, "h"] \arrow[r, "l"] \arrow[dr, "k"'] & K \arrow[u, "q_A"'] \arrow[d, "q_B"] \arrow[r, phantom, "="] & C \\
& B \arrow[ur, dashed, "g"'] &
\end{tikzcd}
$$

Therefore, it holds that

$$
\begin{aligned}
&q_A \circ l = \\
&= (\pi_A \circ q) \circ l && \text{(definition of } q_A\text{)}\\
&= \pi_A \circ (q \circ l) && \text{(associativity)}\\
&= \pi_A \circ \langle h, k\rangle && (q \text{ is an equalizer)}\\
&= h && \text{(definition of } \pi_A\text{)}
\end{aligned}
$$

Similarly, it can be shown that $q_B \circ l = k$, which then shows the commutativity of (1) and (2) in the above diagram.

3. It is still necessary to show that l is the unique morphism which makes (1) and (2) commute. Therefore, let $l' : D \to K$ be such that $q_A \circ l' = h$ and $q_B \circ l' = k$. But then

$$
\begin{aligned}
&\pi_A \circ (q \circ l') = \\
&= (\pi_A \circ q) \circ l' && \text{(associativity)}\\
&= q_A \circ l' && \text{(definition of } q_A\text{)}\\
&= h && \text{(assumption)}
\end{aligned}
$$

Similarly, we can show that $\pi_B \circ (q \circ l') = k$. However, we know that $A \times B$ is a product and hence it follows by uniqueness that $q \circ l = \langle h, k\rangle = q \circ l'$. Now $q \circ l = q \circ l' \Rightarrow l = l'$, since every equalizer is a monomorphism (proposition 4.5.3).

Proposition 4.7.4 The Pullback Lemma. *Consider the following diagram:*

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "c"'] \arrow[dr, phantom, "(1)"] & B \arrow[r, "b"] \arrow[d, "d"] \arrow[dr, phantom, "(2)"] & C \arrow[d, "e"] \\ D \arrow[r, "f"] & E \arrow[r, "g"] & F \end{tikzcd}$$

1. *If both squares are pullbacks, then the outer rectangle is also a pullback.*
2. *If the outer rectangle is a pullback, (1) commutes and (2) is also a pullback, then (1) is also a pullback.*

1. Proof: First note that (1)+(2) (i.e., the outer rectangle) commute since

$g \circ f \circ c =$
$= g \circ d \circ a$ (commutativity of (1))
$= e \circ b \circ a$ (commutativity of (2))

In order to show the universal property, let $i : P \to C$ and $h : P \to D$ be morphisms such that $e \circ i = g \circ (f \circ h)$. Since (2) is a pullback, we obtain a unique $k_2 : P \to B$ such that $b \circ k_2 = i$ and $d \circ k_2 = f \circ h$. The last equation implies, since (1) is a pullback, that there exists a unique $k_1 : P \to A$ such that $c \circ k_1 = h$ and $a \circ k_1 = k_2$ (see next diagram).

$$\begin{tikzcd} P \arrow[dr, "k_1"] \arrow[drr, "k_2"] \arrow[drrr, "i"] \arrow[ddr, "h"'] & & & \\ & A \arrow[r, "a"] \arrow[d, "c"'] & B \arrow[r, "b"'] \arrow[d, "d"'] & C \arrow[d, "e"] \\ & D \arrow[r, "f"'] & E \arrow[r, "g"'] & F \end{tikzcd}$$

Now we have:

$b \circ a \circ k_1 =$
$= b \circ k_2$ ((1) is a pullback)
$= i$ ((2) is a pullback)

and $c \circ k_1 = h$ (assumption).

It remains to show that $k_1 : P \to A$ is the unique morphism such that $c \circ k_1 = h$ and $b \circ a \circ k_1 = i$. Therefore, suppose that there exists another $k_1' : P \to A$ such that $c \circ k_1' = h$ and $b \circ a \circ k_1' = i$. Then we have, using first the universal property of the pullback (2):

$f \circ h =$
$= f \circ c \circ k_1'$ (assumption)
$= d \circ a \circ k_1'$ (commutativity of (1))

Now, we have that

$b \circ (a \circ k_1') = i$ (assumption) and

$d \circ (a \circ k_1') = f \circ h$ (by the above calculation).

However, since (2) is a pullback, k_2 is the unique morphism with this property, and hence we must have

$a \circ k_1' = k_2$

Now, we have that $c \circ k_1' = h$ (assumption) and $a \circ k_1' = k_2$ (see above). But since (1) is a pullback, k_1 is the unique morphism with this property, and hence $k_1' = k_1$, as desired.

2. Proof: We have to show for $h : P \to D$ and $k_2 : P \to B$ such that $f \circ h = d \circ k_2$, that there is a unique $k_1 : P \to A$ such that $c \circ k_1 = h$ and $a \circ k_1 = k_2$. First, note that we have

$$
\begin{aligned}
& g \circ f \circ h = \\
& = g \circ d \circ k_2 && \text{(assumption)} \\
& = e \circ b \circ k_2 && \text{((2) commutes)}
\end{aligned}
$$

Now, since $g \circ f \circ h = e \circ b \circ k_2$ and $(1) + (2)$ is a pullback, we have a unique $k_1 : P \to A$ such that $c \circ k_1 = h$ and $b \circ a \circ k_1 = b \circ k_2$. Since $c \circ k_1 = h$, we still have to show that $a \circ k_1 = k_2$ (see next diagram).

$$
\begin{tikzcd}
P \arrow[dr, "k_1"] \arrow[drr, "k_2"] \arrow[drrr, "b \circ k_2"] \arrow[ddr, "h"'] & & & \\
& A \arrow[r, "a"] \arrow[d, "c"] & B \arrow[r, "b"'] \arrow[d, "d"] & C \arrow[d, "e"] \\
& D \arrow[r, "f"'] & E \arrow[r, "g"'] & F
\end{tikzcd}
$$

Now observe that

$$
\begin{aligned}
& d \circ a \circ k_1 = \\
& = f \circ c \circ k_1 && \text{(commutativity of (1))} \\
& = f \circ h && \text{((1)+(2) is a pullback)}
\end{aligned}
$$

Now $d \circ (a \circ k_1) = f \circ h$ (by the above calculation) and $b \circ (a \circ k_1) = b \circ k_2$ (assumption).

However, since (2) is a pullback, there is a unique $s : P \to B$ such that $b \circ s = b \circ k_2$ and $d \circ s = f \circ h$. Since $s = k_2$ and $s = a \circ k_1$ do the job, we have that $a \circ k_1 = s = k_2$. It remains to show that $k_1 : P \to A$ is the unique morphism such that $c \circ k_1 = h$ and $a \circ k_1 = k_2$. Therefore, let k_1' be such that $c \circ k_1' = h$ and $a \circ k_1' = k_2$. Now, note that

$$
\begin{aligned}
& a \circ k_1' = k_2 \Rightarrow \\
& \Rightarrow b \circ a \circ k_1' = b \circ k_2 && \text{(composition)}
\end{aligned}
$$

In this way we have $c \circ k_1' = h$ (assumption) and $b \circ a \circ k_1' = b \circ k_2$. Since (1)+(2) is a pullback, we have that $k_1 : P \to A$ is the unique morphism which satisfies these properties. Hence, as k_1' also does the job, we have that $k_1 = k_1'$.

$\square$

Example 4.7.5 In **Set**, the pullback of two functions $f : A \to C$ and $g : B \to C$ can be constructed as follows:

$P = \{\langle a, b\rangle : a \in A, b \in B, \text{ and } f(a) = g(b)\}$

with $f' : P \to B$ and $g' : P \to A$ defined as:

$f'(\langle a, b\rangle) = b$
$g'(\langle a, b\rangle) = a$

Therefore, P is a subset of $A \times B$. It is sometimes denoted $A \times_C B$, the product of A and B **over C**, or **fibred product**. This definition clearly satisfies the condition for P being a pullback of f and g, since we have $g \circ f'(\langle a, b\rangle) = g(f'(\langle a, b\rangle)) = g(b) = f(a) = f(g'(\langle a, b\rangle)) = f \circ g(\langle a, b\rangle)$. Now, if there are $i : X \to A$ and $j : X \to B$ such that $f \circ i = g \circ j$, we define the unique $k : X \to B$ as $k = \langle i, j\rangle$ and then we have $g' \circ \langle i, j\rangle = i$ and $f' \circ \langle i, j\rangle = j$.

4.8 Pushouts

The dual notion of a pullback is represented by the following

Definition 4.8.1 *A* **pushout** *of a pair of morphisms $f \in \mathbf{C}(C, A)$ and $g \in \mathbf{C}(C, B)$ is a* **C***-object P and a pair of morphisms $g' \in \mathbf{C}(A, P)$ and $f' \in \mathbf{C}(B, P)$, such that*

1. *$g' \circ f = f' \circ g$;*
2. *for any other* **C***-object X and pair of morphisms $i \in \mathbf{C}(A, X)$, $j \in \mathbf{C}(B, X)$ with $i \circ f = j \circ g$, there exists only one morphism $k : P \to X$ such that $i = k \circ g'$ and $j = k \circ f'$, as shown by the following diagram:*

$$\begin{tikzcd}
& B \arrow[dl, dashed, "j"'] \arrow[d, "f'"] & \\
X & P \arrow[l, "k"'] \arrow[r, phantom, "="] & C \arrow[ul, "g"'] \arrow[dl, dashed, "f"] \\
& A \arrow[ul, "i"] \arrow[u, "g'"'] &
\end{tikzcd}$$

Proposition 4.8.2 *Pushouts are unique up to isomorphism.*

Proof: $\boxed{\text{D}}$

Proposition 4.8.3 *Pushouts can be constructed by coproducts and coequalizers.*

Proof: $\boxed{\text{D}}$

Proposition 4.8.4 *Consider the following diagram:*

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "c"'] \arrow[dr, phantom, "(1)"] & B \arrow[r, "b"] \arrow[d, "d"'] \arrow[dr, phantom, "(2)"] & C \arrow[d, "e"'] \\
D \arrow[r, "f"] & E \arrow[r, "g"] & F
\end{tikzcd}$$

1. *If both squares are pushouts, then the outer rectangle is also a pushout.*
2. *If the outer rectangle is a pushout, (2) commutes and (1) is also a pushout, then (2) is also a pushout.*

Proof: $\boxed{\text{D}}$

Remark 4.8.5 Unfortunately, one can not claim, in general, that (1) is a pushout if (1)+(2) and (2) are pushouts. However, at least in **Set** and **Graph** this is true, provided that $b : B \to C$ is injective (see [EK76a] and [EK76b], for instance).

Example 4.8.6 Pushouts as amalgamations. If the effect of coequalizers is that of forcing identifications, that of pushouts is to form the so called **amalgamated sums**. First, consider the case of a set S and three subsets S_0, S_1 and S_2 with $S_0 = S_1 \cap S_2$ and $S = S_1 \cup S_2$. Then the diagram,

$$\begin{tikzcd}
S_0 \arrow[r, "a"] \arrow[d, "b"'] & S_1 \arrow[d, "c"] \\
S_2 \arrow[r, "d"'] & S
\end{tikzcd}$$

where all the morphisms are inclusions (i.e., $d \circ b = c \circ a$), is a pushout. The definition of a pushout translates into the obvious fact that given the functions $f_1 : S_1 \to T$ and $f_2 : S_2 \to T$ and $f_1|S_0 = f_2|S_0$, (i.e, $f_2 \circ b = f_1 \circ a$), then there is a unique function $f : S \to T$ with $f|S_1 = f_1$ and $f|S_2 = f_2$.

Now consider a little more general situation. We begin with the sets S_0, S_1 and S_2 and the functions $a : S_0 \to S_1$ and $b : S_0 \to S_2$. If a and b are injective, then, up to isomorphism, this case can be viewed like the previous one. Obviously, we can't form the union of S_1 and S_2, but rather first the disjoint union and then, for $s \in S_0$, identify the element $a(s) \in S_1$ with $b(s) \in S_2$. In this case, the pushout is called **amalgamated sum** of S_1 and S_2.

Example 4.8.7 In **Set**, a pushout of two functions $f : B \to A$ and $g : C \to B$ is obtained by, firstly constructing the disjoint union of A and B, and after by identifying $\langle f(c), g(c)\rangle$, for all $c \in C$, by a coequalizer. Here, the construction is a simple dualization of the one presented in theorem 4.7.3, i.e., we need a coequalizer for the pair of morphisms $in_A \circ f : C \to A + B$, $in_B \circ g : C \to A + B$ (where in_A and in_B are the injections), and hence $c : A + B \to P$, as shown by the following diagram.

$$\begin{tikzcd}
& A \arrow[dl, dashed, "c \circ in_A"'] \arrow[d, "in_A"] & \\
P & A + B \arrow[l, "c"'] & C \arrow[ul, "f"'] \arrow[l, shift right, "in_A \circ f"'] \arrow[l, shift left, "in_B \circ g"] \arrow[dl, dashed, "g"] \\
& B \arrow[ul, "c \circ in_B"] \arrow[u, "in_B"'] &
\end{tikzcd}$$

Remark 4.8.8 In categories like **Set** and **Graph** the following pullback diagram

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "b"'] & B \arrow[d, "c"] \\
C \arrow[r, "d"'] & D
\end{tikzcd}$$

(where c and d are injective) is also a pushout (where a and b are also injective). These diagrams are normally called **Doolittle** diagrams.

In Table 4.2 we summarize the categorical concepts seen up to now with their respective dual ones.

CONCEPT	*DUAL CONCEPT*
Category	Category
Monomorphism	Epimorphism
Isomorphism	Isomorphism
Retraction	Section
Terminal Object	Initial Object
Zero Object	Zero Object
Product	Coproduct
Product morphism	Coproduct morphism
Equalizer	Coequalizer
Pullback	Pushout

Table 4.2: Categorical concepts and its dual ones

4.9 Bibliographic notes

A highly readable presentation of the constructions introduced in this chapter can be found in [Gol86], pp. 43-68, [BW90], chapters 6 and 8, and [EP 72], chapter 3. The latter is highly recommended for readers interested in detailed proofs for main properties related to each construction. Also, [EP72] makes heavy use of diagrams both in presentation of concepts and demonstration of results, much more than any other textbook concerned with category theory.

Chapter 5

General limit constructions and completeness

The concept of limit embodies the general idea of a universal construction, that is, of an entity which has a privileged behavior among a class of objects that satisfy a certain property. The only way to define a property in the categorical language is by specifying the existence and equality of certain morphisms, that is, essentially by imposing the existence of a particular commutative diagram among objects inside the category.

Terminal objects, products, equalizers, and pullbacks are examples of universal constructions. These are all specific instances of the more general notion of a limit of a diagram.

5.1 Limits

Definition 5.1.1 *Let* $\mathbf{C}$ *be a category and* $\mathbf{D}$ *a diagram in* $\mathbf{C}$. *A* **cone** *for* $\mathbf{D}$ *is a* $\mathbf{C}$*-object* X *and morphisms* $f_i : X \to D_i$ *(one for each* D_i *in* $\mathbf{D}$*), such that for each* $\mathbf{C}$*-morphism* $g : D_i \to D_j$ *in* $\mathbf{D}$, *the diagram*

$$\begin{tikzcd} & X \arrow[dl, dashed, "f_i"'] \arrow[dr, "f_j"] & \\ D_i \arrow[rr, "g"'] & & D_j \end{tikzcd}$$

commutes. We use the notation $\{f_i : X \to D_i\}$ for cones.

Definition 5.1.2 *A* **limit** *for a diagram* $\mathbf{D}$ *is a cone* $\{f_i : X \to D_i\}$ *with the property that if* $\{f_i' : X \to D_i\}$ *is another cone for* $\mathbf{D}$, *then there is a unique morphism* $k : X' \to X$ *such that the diagram*

$$\begin{tikzcd}
X' \arrow[rr, "k"] \arrow[dr, "f'_i"'] & & X \arrow[dl, dotted, "f_i"] \\
& D_i &
\end{tikzcd}$$

commutes for every D_i in $\mathbf{D}$.

We now give two different proofs for the next

Proposition 5.1.3 *Limits are unique up to isomorphism.*

Proof 1: Note that the cones for a diagram $\mathbf{D}$ form a category, where the objects are cones and the morphisms are morphisms $k : \{f_i : X \to D_i\} \to \{f'_i : X \to D_i\}$ such that $f_i \circ k = f'_i$ for each $D_i \in \mathbf{D}$. Now, a limit in this category is a terminal object, and since terminal objects are unique up to isomorphism, so are limits.

$\square$

Proof 2: Consider the following diagram:

$$\begin{tikzcd}
X' \arrow[r, "k"] \arrow[dr, "g_i"'] & X \arrow[r, "k'"] \arrow[d, "f_i"] & X' \arrow[dl, dotted, "g_i"] \\
& D_i &
\end{tikzcd}$$

Suppose that $\{f_i : X \to D_i\}$ and $\{g_i : X' \to D_i\}$ are limits for a diagram $\mathbf{D}$. Now since $\{f_i : X \to D_i\}$ is a limit for $\mathbf{D}$ then there is a unique $k : X' \to X$ such that $f_i \circ k = g_i$ for each D_i in $\mathbf{D}$. Because $\{g_i : X' \to D_i\}$ is a limit for $\mathbf{D}$ then there is a unique $k' : X \to X'$ such that $g_i \circ k' = f_i$ for each D_i in $\mathbf{D}$.

To see that the above diagram commutes note that

$$\begin{array}{lr}
g_i \circ (k' \circ k) = & \\
= (g_i \circ k') \circ k & \text{(associativity)} \\
= f_i \circ k & (\{g_i : X' \to D_i\} \text{ is a limit for } \mathbf{D}) \\
= g_i & (\{f_i : X \to D_i\}) \text{ is a limit for } \mathbf{D})
\end{array}$$

However, consider this diagram:

$$\begin{tikzcd}
X' \arrow[rr, "s"] \arrow[dr, "g_i"'] & & X' \arrow[dl, dashed, "g_i"] \\
& D_i &
\end{tikzcd}$$

Since $(\{g_i : X' \to D_i\}$ is a limit for $\mathbf{D}$, then there is a unique $s : X' \to X'$ such that $g_i \circ s = g_i$ for each D_i in $\mathbf{D}$. We have already seen that $s = k' \circ k$ does the job. But $s = id_{X'}$ also does, since $g_i \circ id_{X'} = g_i$ for each D_i in $\mathbf{D}$. By the uniqueness requirement in the definition of a limit we have that $id_{X'} = s = k' \circ k$. Interchanging the roles of X and X' in the above argument leads to $k \circ k' = id_X$.

□

Example 5.1.4 Given two $\mathbf{C}$-objects A and B, let $\mathbf{D}$ be the diagram

$$A \qquad\qquad\qquad B$$

with the two vertices labeled A and B and no edges. Then a cone for this diagram is an object X with two morphisms f and g of the form

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

Note that, we have trivially $id_A \circ f = f$ and $id_B \circ g = g$. Now, a limiting $\mathbf{D}$-cone, if it exists, is a product of A and B, since if $\{f' : C \to A, g' : C \to B\}$ is another *cone*, then we have a unique $k : C \to X$ such that $f \circ k = f'$ and $g \circ k = g'$, which is exactly the definition of a product.

Example 5.1.5 Let $\mathbf{D}$ be the empty diagram with no vertices and no edges. A cone for $\mathbf{D}$ in a category $\mathbf{C}$ is any $\mathbf{C}$-object ($\mathbf{D}$ has no vertices so the cone has no morphisms). A limiting cone is then an object C with the additional requirement that for any $\mathbf{C}$-object C' (i.e., for any $\mathbf{D}$-cone) there is exactly one morphism from C' to C. In other words, C is a terminal object.

Example 5.1.6 Let $\mathbf{D}$ be the diagram

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

with three vertices and two edges. A cone for $\mathbf{D}$ is an object P and three morphisms f', g' and h such that the following diagram commutes:

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

But this is equivalent to the commutativity of

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

because h is completely determined as the common composite $f \circ g' = h = g \circ f'$.

If P, f', and g' form a limit, then they have the universal property among objects and morphisms that make this diagram commute – that is, given any object P' with morphisms f'', g'', and h' making the analogous diagram commute, there will be a unique morphism $k : P' \to P$ such that $f'' = f' \circ k, g'' = g' \circ k$, and $h' = h \circ k$ factor. Ignoring h' as above, this shows that a limit for $\mathbf{D}$ is a pullback of f and g.

Example 5.1.7 Let $\mathbf{D}$ be the diagram

$$\begin{tikzcd}
A \arrow[r, shift left, "f_1"] \arrow[r, shift right, "f_2"'] & B
\end{tikzcd}$$

A $\mathbf{D}$-cone for this diagram is an object C and two morphisms $h_1 : C \to A$ and $h_2 : C \to B$ such that the following diagrams commute:

$$\begin{tikzcd}
C \arrow[d, "h_1"'] \arrow[dr, "h_2"] & \\
A \arrow[r, "f_1"'] & B
\end{tikzcd}
\qquad
\begin{tikzcd}
C \arrow[d, "h_1"'] \arrow[dr, "h_2"] & \\
A \arrow[r, "f_2"'] & B
\end{tikzcd}$$

But this requires $h_2 = f_1 \circ h_1 = f_2 \circ h_1$ and so h_2 is totally determined by f_1 and h_1 (or by f_2 and h_2). So we can say that a $\mathbf{D}$-cone is a object C and a morphism $h_1 : C \to A$ such that $f_1 \circ h_1 = f_2 \circ h_1$. Now if C and h_1 form a limit for $\mathbf{D}$, then they have the universal property among objects and morphisms that make this diagram commute – that is, given an object C' with a morphism $e : C' \to A$ such that $f_1 \circ e = f_2 \circ e$, there will be a unique morphism k from C' to C, such that $h_1 \circ k = e$, i.e., a limit for $\mathbf{D}$ is an equalizer of f_1 and f_2

Definition 5.1.8 *Dually, a* **cocone** *for a diagram* $\mathbf{D}$ *in a category* $\mathbf{C}$ *is a* $\mathbf{C}$*-object* X *and a collection of* $\mathbf{C}$*-morphisms* $f_i : D_i \to X$ *(one for each* D_i *in* $\mathbf{D}$*) such that the following diagram commutes:*

$$\begin{tikzcd} & X & \\ D_i \arrow[ur, dotted, "f_i"] \arrow[rr, "g"'] & & D_j \arrow[ul, "f_j"'] \end{tikzcd}$$

Definition 5.1.9 *A* **colimit** *or* **inverse limit** *for* $\mathbf{D}$ *is then a cocone* $\{f_i : D_i \to X\}$ *with the couniversal property that for any other cocone* $\{f_i' : D_i \to X'\}$ *there is a unique morphism* $k : X \to X'$ *such that the diagram*

$$\begin{tikzcd} X' & & X \arrow[ll, "k"'] \\ & D_i \arrow[ul, "f_i'"] \arrow[ur, dotted, "f_i"'] & \end{tikzcd}$$

commutes for every D_i *in* $\mathbf{D}$.

Proposition 5.1.10 *Colimits are unique up to isomorphism.*

Proof: $\boxed{\mathrm{D}}$

We now dualize the examples 5.1.4–5.1.7.

Example 5.1.11 Given two $\mathbf{C}$-objects A and B, let $\mathbf{D}$ be the diagram

$$A \qquad\qquad\qquad B$$

with the two vertices labeled A and B and no edges. Then a $\mathbf{D}$-colimit, if it exists, is a coproduct of A and B.

Example 5.1.12 Let $\mathbf{D}$ be the empty diagram with no vertices and no edges. A cocone for $\mathbf{D}$ in a category $\mathbf{C}$ is any $\mathbf{C}$-object ($\mathbf{D}$ has no vertices so the cocone has no morphisms). A $\mathbf{D}$-colimit, if it exists, is a initial object in $\mathbf{C}$.

Example 5.1.13 Let $\mathbf{D}$ be the diagram

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

with three vertices and two edges. A **D**-colimit, if it exists, is a pushout of f and g.

Example 5.1.14 Let **D** be the diagram

$$\begin{tikzcd} A \arrow[r, shift left, "f_1"] \arrow[r, shift right, "f_2"'] & B \end{tikzcd}$$

A **D**-colimit, if it exists, is a coequalizer of f_1 and f_2.

Example 5.1.15 Let $(P, \leq_P)$ be a poset considered as a category. Then limits correspond to greater lower bounds, while colimits to least upper bounds.

5.2 Completeness

Definition 5.2.1 *A* **finite diagram** *is one that has a finite number of vertices and a finite number of edges between them.*

Definition 5.2.2 *A* **small diagram** *is a diagram whose collection of vertices and edges are really sets and not proper classes.*

Definition 5.2.3 *A category is said to*

1. **have (finite) products** *provided that for each (finite) set-indexed family of objects there exists a product;*

2. **have equalizers** *provided that for each pair of morphisms (with same domain and codomain) there exists an equalizer;*

3. **have pullbacks** *provided that for each pair of morphisms with the same codomain there exists an pullback.*

Remark 5.2.4 The dual notions **have (finite) coproducts**, **have coequalizers**, and **have pushouts** are straightforward.

Definition 5.2.5 *A category* **C** *is said to be*

1. **finitely complete** *if for each finite diagram in* **C** *there exists a limit;*

2. **complete** *if for each small diagram in* $\mathbf{A}$ *there exists a limit.*

Remark 5.2.6 The dual notions **finitely cocomplete** and **cocomplete** are immediate.

Theorem 5.2.7 *Let* $\mathbf{D}$ *be a diagram in a category* $\mathbf{C}$*, with sets* V *of vertices and* E *of edges. If every* V*-indexed and every* E*-indexed family of objects in* $\mathbf{C}$ *has a product and every pair of morphisms with the same domain and codomain in* $\mathbf{C}$ *has an equalizer, then* $\mathbf{D}$ *has a limit.*

Proof: Begin by forming the following V-indexed and E-indexed products and associated projections (note that $\Pi_{(I \xrightarrow{e} J)\in E} D_J$ is the product of the targets of the edges of $\mathbf{D}$):

$$\begin{tikzcd}
 & & D_J \\
\Pi_{I\in V} D_I \arrow[urr, "\pi_J"] \arrow[d, "\pi_I"'] & & \Pi_{(I \xrightarrow{e} J)\in E} D_J \arrow[u, "\pi_e"'] \arrow[d, "\pi_e"] \\
D_I \arrow[rr, "D_e"] & & D_J
\end{tikzcd}$$

Since each D_J at the top of the diagram is a component of $\Pi_{I\in V} D_I$, there exists, for each such D_J,a (projection) morphism $\pi_J : (\Pi_{I\in V} D_I) \to D_J$. By the universal property of indexed products, this implies the existence of a unique morphism $p : (\Pi_{I\in V} D_I) \to (\Pi_{(I \xrightarrow{e} J)\in E} D_J)$ such that $\pi_e \circ p = \pi_J$ for each edge $e : I \to J$. Similarly, for each D_J at the bottom right, there is a morphism $(D_e \circ \pi_I) : (\Pi_{I\in V} D_I) \to D_{jJ}$. This implies the existence of a unique morphism $q : (\Pi_{I\in V} D_I) \to (\Pi_{(I \xrightarrow{e} J)\in E} D_J)$ such that $\pi_e \circ q = D_e \circ \pi_I$ for each edge $e : I \to J$.

$$\begin{tikzcd}
 & & D_J \\
\Pi_{I\in V} D_I \arrow[urr, "\pi_J"] \arrow[d, "\pi_I"'] \arrow[rr, shift left, "p"] \arrow[rr, shift right, "q"'] & & \Pi_{(I \xrightarrow{e} J)\in E} D_J \arrow[u, "\pi_e"'] \arrow[d, "\pi_e"] \\
D_I \arrow[rr, "D_e"] & & D_J
\end{tikzcd}$$

Let h be an equalizer of p and q. Set $f_I = \pi_I \circ h$ for each $I \in V$.

$$\begin{tikzcd}
 & & & & D_J \\
X \arrow[rr, "h"] \arrow[drr, "f_I"'] & & \Pi_{I\in V} D_I \arrow[urr, "\pi_J"] \arrow[d, "\pi_I"'] \arrow[rr, shift left, "p"] \arrow[rr, shift right, "q"'] & & \Pi_{(I \xrightarrow{e} J)\in E} D_J \arrow[u, "\pi_e"'] \arrow[d, "\pi_e"] \\
 & & D_I \arrow[rr, "D_e"] & & D_J
\end{tikzcd}$$

We claim that $\{f_I : X \to D_I\}$ is a limit for $\mathbf{D}$. We must show first that it is a cone for $\mathbf{D}$, and furthermore that it is universal among cones for $\mathbf{D}$, i.e., if $\{f'_I : X' \to D_I\}$ is also a cone for $\mathbf{D}$, then there exists a unique morphism $k : X' \to X$ with $f_I \circ k = f'_I$ for every vertex I.

For each edge $e : I \to J$ in E, the commutativity of the diagram

$$
\begin{tikzcd}
D_I \arrow[rr, "D_e"] & & D_J \\
& X \arrow[ul, "f_I"] \arrow[ur, dotted, "f_J"'] &
\end{tikzcd}
$$

is established as follows (referring to the previous diagram):

$$
\begin{aligned}
& D_e \circ f_I = && \\
& = D_e \circ \pi_I \circ h && \text{(definition of } f_I\text{)} \\
& = \pi_e \circ q \circ h && \text{(commutativity of the lower rectangle)} \\
& = \pi_e \circ p \circ h && \text{(since } h \text{ equalizes } p \text{ and } q\text{)} \\
& = \pi_J \circ h && \text{(commutativity of the upper triangle)} \\
& = f_J && \text{(definition of } f_J\text{)}
\end{aligned}
$$

This shows that $\{f_I : X \to D_I\}$ is a cone for $\mathbf{D}$. We must now show that it is universal among cones for $\mathbf{D}$.

Assume that $\{f'_I : X' \to D_I\}$ is also a cone for $\mathbf{D}$. By the universal property of the product $\Pi_{I\in V} D_I$, there is a unique morphism $h' : X' \to (\Pi_{I\in V} D_I)$ such that $\pi_I \circ h' = f'_I$ for each $I \in V$. Now, for any edge $e : I \to J$ in E,

$$
\begin{aligned}
& \pi_e \circ p \circ h' = && \\
& = \pi_J \circ h' && (\Pi_{(I\xrightarrow{e}J)\in E} D_J \text{ is a product}) \\
& = f'_J && \text{(definition of } f'_J\text{)} \\
& = D_e \circ f'_I && (\{f'_I : X' \to D_I\} \text{ is a cone}) \\
& = D_e \circ \pi_I \circ h' && \text{(definition of } f'_I\text{)} \\
& = \pi_e \circ q \circ h' && (\Pi_{(I\xrightarrow{e}J)\in E} D_J \text{ is a product})
\end{aligned}
$$

This establishes the commutativity of the diagram

$$
\begin{tikzcd}
& X' \arrow[dl, "f'_J = \pi_e \circ p \circ h' = \pi_e \circ q \circ h'"'] \arrow[d, shift right, "p \circ h'"'] \arrow[d, shift left, "q \circ h'"] \\
D_J & \Pi_{(I\xrightarrow{e}J)\in E} D_J \arrow[l, "\pi_e"']
\end{tikzcd}
$$

which, by the universal property of the product $\Pi_{(I\xrightarrow{e}J)\in E} D_J$, implies that $p \circ h' = q \circ h'$, that is, that h' equalizes p and q.

Since h is an equalizer of p and q, the universal property of equalizers guarantees the existence of a unique $k : X' \to X$ such that $h \circ k = h'$. Now from the diagram

$$
\begin{tikzcd}
X \arrow[dr, "h"] \arrow[drr, "f_I"] & & \\
 & \Pi_{I\in V} D_I \arrow[r, "\pi_I"] & D_I \\
X' \arrow[uu, "k"] \arrow[ur, dashed, "h'"] \arrow[urr, "f'_I"'] & &
\end{tikzcd}
$$

we have:

$f_I \circ k =$
$= \pi_I \circ h \circ k$ (definition of f_I)
$= \pi_I \circ h'$ (h is an equalizer)
$= f'_I$ (definition of f'_I)

Finally, we must show that k is the *only* morphism from X' to X such that $f_I \circ k = f'_I$ for all $I \in V$. But if any morphism k' satisfies $f_I \circ k' = f'_I$, we have (by definition of f_I and f'_I) that $\pi_I \circ h \circ k' = \pi_I \circ h'$ for all $I \in V$, and so, by the universal property of the product $\Pi_{I\in V} D_I$, that $h \circ k' = h'$. But, as h is an equalizer, we have that the only morphism with this property is k, so $k = k'$.

□

Remark 5.2.8 The relevance of theorem 5.2.7 is that, in general, it is simpler to check the existence of products and equalizers than to prove directly the existence of limits.

Proposition 5.2.9 *(Construction of equalizers via products and pullbacks) Let* $\mathbf{C}$ *be a category. If* $f, g : A \to B$ *are* $\mathbf{C}$*-morphisms,* $(A \times B, \pi_A, \pi_B)$ *is a product, and*

$$
\begin{tikzcd}
P \arrow[r, "p_2"] \arrow[d, "p_1"'] & A \arrow[d, "\langle id_A, f\rangle"] \\
A \arrow[r, "\langle id_A, g\rangle"'] & A \times B
\end{tikzcd}
$$

is a pullback square, then $p_1 = p_2$ *is an equalizer of* f *and* g*.*

Proof: Since $\langle id_A, f\rangle \circ p_1 = \langle id_A, g\rangle \circ p_2$, we have

$p_1 =$
$= id_A \circ p_1$ (identity)

$= (\pi_A \circ \langle id_A, f\rangle) \circ p_1$ (by projection of π_A)
$= \pi_A \circ (\langle id_A, f\rangle \circ p_1)$ (associativity)
$= \pi_A \circ (\langle id_A, g\rangle \circ p_2)$ (assumption)
$= (\pi_A \circ \langle id_A, g\rangle) \circ p_2$ (associativity)
$= id_A \circ p_2$ (definition of π_A)
$= p_2$ (identity)

Similarly we have:

$f \circ p_1 =$
$= (\pi_B \circ \langle id_A, f\rangle) \circ p_1$ (by projection of π_B)
$= \pi_B \circ (\langle id_A, f\rangle \circ p_1)$ (associativity)
$= \pi_B \circ (\langle id_A, g\rangle \circ p_2)$ (assumption)
$= g \circ p_2$ (definition of π_B)
$= g \circ p_1$ ($p_1 = p_2$ from above)

which shows that p_1 equalizes f and g.

Now we must show that p_1 is universal among all others morphisms $k : K \to A$ such that $f \circ k = g \circ k$. To see this note that

$$\pi_B \circ \langle id_A, f\rangle \circ k = f \circ k = g \circ k = \pi_B \circ \langle id_A, g\rangle \circ k, \text{ and}$$

$$\pi_A \circ \langle id_A, f\rangle \circ k = id_A \circ k = k = id_A \circ k = \pi_A \circ \langle id_A, g\rangle \circ k.$$

Now, by the universal property of the product $A \times B$, we obtain $\langle id_A, f\rangle \circ k = \langle id_A, g\rangle \circ k$ which shows that k equalizes $\langle id_A, f\rangle$ and $\langle id_B, g\rangle$. Now since the square is a pullback, there is a unique $h : K \to P$ such that $k = p_1 \circ h$, which finally shows that p_1 is an equalizer of f and g.

Proposition 5.2.10 *If* 1 *is a terminal object, then the following are equivalent:*

(1)

$$\begin{tikzcd}
P \arrow[r, "\pi_A"] \arrow[d, "\pi_B"'] & A \arrow[d, "t_A"] \\
B \arrow[r, "t_B"'] & 1
\end{tikzcd}$$

is a pullback square.

(2) (P, π_A, π_B) *is a product of* A *and* B.

(1) $\Rightarrow$ (2): Suppose that there is a pair of morphism $f : C \to A$ and $g : C \to B$. Then, since

both $t_B \circ g$ and $t_A \circ f$ are morphisms from C to 1 and 1 is a terminal object, the square below commutes.

$$\begin{tikzcd} C \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "t_A"] \\ B \arrow[r, "t_B"'] & 1 \end{tikzcd}$$

Now, by the uniqueness requirement in the definition of a pullback, there is a unique morphism $h : C \to P$ such that $f = \pi_A \circ h$ and $g = \pi_B \circ h$, which shows that P is a product of A and B.

(2) $\Rightarrow$ (1): Since 1 is a terminal object, we have that $t_B \circ \pi_B = t_A \circ \pi_A$, because both are morphisms from P to 1. Since P is a product, if there is a pair of morphisms, say $f : C \to A$ and $g : C \to B$, we have trivially $t_B \circ g = t_A \circ f$ since 1 is terminal. By the uniqueness requirement in the definition of a product, there is exactly one $h : C \to P$ such that $\pi_A \circ h = f$ and $\pi_B \circ h = g$, which shows that the first square is a pullback.

$\square$

Theorem 5.2.11 *For each category* **C***, the following conditions are equivalent:*

1. **C** *is complete.*
2. **C** *has products and equalizers.*
3. **C** *has products and pullbacks.*

Proof:

1 $\Rightarrow$ 3: Products and pullbacks are special cases of limits.
3 $\Rightarrow$ 2: By proposition 5.2.9.
2 $\Rightarrow$ 1: By theorem 5.2.7.

Theorem 5.2.12 *For each category* **C***, the following conditions are equivalent:*

1. **C** *is cocomplete.*
2. **C** *has coproducts and coequalizers.*
3. **C** *has coproducts and pushouts.*

Proof: $\boxed{\text{D}}$

Theorem 5.2.13 *For each category* **C***, the following conditions are equivalent:*

1. **C** *is finitely complete.*

2. **C** *has finite products and equalizers.*

3. **C** *has pullbacks and a terminal object.*

Proof:

$1 \Rightarrow 3$: Pullbacks and terminal objects are limits of finite diagrams.
$3 \Rightarrow 2$: By proposition 5.2.10 we have that products of pair of objects can be formed via pullbacks and morphisms to a terminal object. By proposition 5.2.9, equalizers can be constructed via products and pullbacks.
$2 \Rightarrow 1$: By theorem 5.2.7.

Theorem 5.2.14 *For each category* **C**, *the following conditions are equivalent:*

1. **C** *is finitely cocomplete.*

2. **C** *has finite coproducts and coequalizers.*

3. **C** *has pushouts and a initial object.*

Proof: $\boxed{\text{D}}$

5.3 Bibliographic notes

The concepts of limit and completeness are very basic in category theory. For a very nice presentation of limits the reader may look at [AL91], [Gol86], [MaC71] and [AHS90]. The concept of completeness is treated in deep in [AHS90]. Theorem 5.2.7 is an adaptation from [Pie90]. A more interesting construction of limits/colimits, especially for practical applications, is when one first constructs the product/coproduct of all objects in the diagram and then iteratively computes equalizers/coequalizers (see [Pad91] for details).

Chapter 6

Functors and natural Transformations

The starting point of Category Theory is the premise that every kind of mathematically structured object comes equipped with a notion of "acceptable" transformation or construction, i.e., a morphism that preserves the structure of the object. This premise holds for categories themselves: a **functor** is the "natural" notion of morphisms between categories, i.e., a functor F from a category $\mathbf{C}$ to a category $\mathbf{D}$ 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 a category has. By a further step, the question about what a morphism between functors should look like suggest the notion of a **natural transformation**.

6.1 Functors

6.1.1 Basic concepts and examples

Definition 6.1.1.1 *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be categories. A (covariant)* **functor** $F : \mathbf{C} \to \mathbf{D}$ *is a pair of mappings,* $F_{Ob} : Ob(\mathbf{C}) \to Ob(\mathbf{D})$, $F_{Mor} : Mor(\mathbf{C}) \to Mor(\mathbf{D})$ *for which*

1. *If* $f : A \to B$ *in* $\mathbf{C}$, *then* $F_{Mor}(f) : F_{Ob}(A) \to F_{Ob}(B)$ *in* $\mathbf{D}$.
2. $F_{Mor}(id_A) = id_{F_{Ob}(A)}$;
3. $F_{Mor}(g \circ f) = F_{Mor}(g) \circ F_{Mor}(f)$.

Definition 6.1.1.2 *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be categories. A* **contravariant** *functor* $F : \mathbf{C} \to \mathbf{D}$ *is a pair of mappings* $F_{Ob} : Ob(\mathbf{C}) \to Ob(\mathbf{D})$, $F_{Mor} : Mor(\mathbf{C}) \to Mor(\mathbf{D})$ *for which*

1. *If* $f : A \to B$ *in* $\mathbf{C}$, *then* $F_{Mor}(f) : F_{Ob}(B) \to F_{Ob}(A)$ *in* $\mathbf{D}$.
2. $F_{Mor}(id_A) = id_{F_{Ob}(A)}$;
3. $F_{Mor}(g \circ f) = F_{Mor}(f) \circ F_{Mor}(g)$.

Remarks 6.1.1.3

1. Informally, a contravariant functor is a functor which reverses the direction of morphisms and hence, also the order of composition.

2. It is usual practice to omit the subscripts "Ob" and "Mor" as it always clear from context whether the functor is meant to operate on objects or on morphisms.

3. From now on the term "functor" will used always to denote a covariant functor. Contravariance will be stated explicitly.

Proposition 6.1.1 *If* $F : \mathbf{A} \to \mathbf{B}$ *and* $G : \mathbf{B} \to \mathbf{C}$ *are covariant functors, then the* **composition** $(G \circ F) : \mathbf{A} \to \mathbf{C}$ *defined by* $(G \circ F)(A) = G(F(A))$ *and* $(G \circ F)(f) = G(F(f))$ *is also a covariant functor.*

Proof:

$1.(G \circ F)(id_A) =$
$= G(F(id_A))$ (definition of $G \circ F$)
$= G(id_{F(A)})$ (F is a functor)
$= id_{G(F(A))}$ (G is a functor)
$= id_{(G \circ F)(A)}$ (definition of $G \circ F$)

$1.(G \circ F)(g \circ f) =$
$= G(F(g \circ f))$ (definition of $G \circ F$)
$= G(F(g) \circ F(f))$ (F is a functor)
$= G(F(g)) \circ G(F(f))$ (G is a functor)
$= (G \circ F)(g) \circ (G \circ F)(f)$ (definition of $G \circ F$)

$\square$

Remark 6.1.1.4 We note that the composition circle is usually omitted when composing functors so that we sometimes write GF for $G \circ F$.

We leave to the reader the trivial task of proving the following proposition:

Proposition 6.1.2

1. *The composition of two contravariant functors* $F : \mathbf{A} \to \mathbf{B}, G : \mathbf{B} \to \mathbf{C}$, *is a covariant functor* $G \circ F : \mathbf{A} \to \mathbf{C}$.

2. *If* $F : \mathbf{A} \to \mathbf{B}$ *is a contravariant functor and* $G : \mathbf{B} \to \mathbf{C}$ *is a covariant functor, then the composition* $G \circ F : \mathbf{A} \to \mathbf{C}$ *is a contravariant functor.*

3. *If* $F : \mathbf{A} \to \mathbf{B}$ *is a covariant functor and* $G : \mathbf{B} \to \mathbf{C}$ *is a contravariant functor, then the composition* $G \circ F : \mathbf{A} \to \mathbf{C}$ *is a contravariant functor.*

Proof: $\square$

Example 6.1.1.5 It is easy to see that a monoid homomorphism $f : M \to N$ determines a functor from $\mathbf{C}(M)$ to $\mathbf{C}(N)$. On objects, a homomorphism f must take the single object to M to the single object of N, and 6.1.1.1.1 is trivially verified since all morphisms in M have the same domain and codomain, and similarly for N. Then 6.1.1.1.2 and 6.1.1.1.3 says precisely that f is a monoid homomorphism.

Example 6.1.1.6 Let $\mathbf{C}(P)$ and $\mathbf{C}(Q)$ be the category determined by the posets $(P, \leq_P)$ and $(Q, \leq_Q)$ respectively. Remember that there is exactly one morphism from x to y in S (or in T) if and only if $x \leq y$; otherwise there is no morphism from x to y.

Let $f : S \to T$ be the functor. 6.1.1.1.1 says if there is a morphism from x to y, then there is a morphism from $f(x)$ to $f(y)$; in other words,

$$\text{if } x \leq_P y \text{ then } f(x) \leq_Q f(y)$$

Thus f is a monotone map and both 6.1.1.1.2 and 6.1.1.1.3 impose no additional conditions on f because they each assert the equality of two specified morphisms between two specified objects, and in a poset as a category, all morphisms between two objects are equal.

Example 6.1.1.7 Given a set S, we can form the set $List(S)$ of finite lists with elements drawn from S. This defines a mapping $List : \mathbf{Set} \to \mathbf{Set}$, which is the object part of a functor. The morphism part takes a function $f : S \to S'$ to a function $List(f) : List(S) \to List(S')$ that, given a list $L = [s_1, \ldots, s_n]$, "maps" f over the elements of L:

$$List(f)(L) = f^*(L) = [f(s_1), \ldots, f(s_n)]$$

The set of lists with elements drawn from S has some additional structure that we have yet not taken into account: an associative binary concatenation operation $\diamond$ over $List(S)$ and a empty list $[]$ that act as an identity for $\diamond$, i.e., $[] \diamond L = L = L \diamond []$. Thus, $(List(S), \diamond, [])$ is a monoid, and $List : \mathbf{Set} \to \mathbf{Mon}$ is indeed a functor (see proposition 6.1.3) taking each set S to the monoid of lists with elements drawn from S.

The morphism part of $List$ takes a set function f to a monoid homomorphism $List(f) = f^*$. The fact that f^* is a homomorphism corresponds exactly to the first two lines in the definition of f^*:

$$\begin{aligned}
&f^*([]) = [] \\
&f^*(L \diamond L') = f^*(L) \diamond f^*(L') \\
&f^*([s]) = [f(s)]
\end{aligned}$$

$List(S)$ is (often) called the **free monoid generated by** S (see 2.3.8). Thus any set function between sets induces a monoid homomorphism between the corresponding free monoids. f^* is called αf in [Bac81] and is often called **applytoall** or **maplist** in the computing science literature.

Proposition 6.1.3 *The map $List$: **Set** $\to$ **Mon** defined in 6.1.1.7 is a functor.*

Proof: Let S be a set, $[s_1, \ldots, s_n] = L \in List(S)$, $f \in \mathbf{Set}(A, B)$, $g \in \mathbf{Set}(B, C)$, $id_S \in \mathbf{Set}(S, S)$. Then we have, for all $L \in List(S)$:

1. Preservation of identities:

$List(id_S)(L) =$
$= [id_S(s_1), \ldots, id_S(s_n)]$ (definition of $List(id_S)$)
$= [s_1, \ldots, s_n] = L$ (definition of id_S and L)
$= id_{List(S)}(L)$ (definition of $id_{List(S)}$)

2.Preservation of composition

2.$List((g \circ f))(L) =$
$= [(g \circ f)(s_1), \ldots, (g \circ f)(s_n)]$ (definition of L and $List(g \circ f)$)
$= [g(f(s_1)), \ldots, g(f(s_n))]$ (composition)
$= g^*([f(s_1), \ldots, f(s_n)])$ (definition of g^*)
$= g^*(f^*(L))$ (definition of L and f^*)
$= g^* \circ f^*(L)$ (composition)
$= List(g) \circ List(f)(L)$ (definition of g^* and f^*)

$\square$

Example 6.1.1.8 Generalizing example 6.1.1.7 we have the **free monoid functor** from **Set** to **Mon** that takes a set A to the free monoid $F(A)$, which is the Kleene closure A^* with concatenation as operation, and a function $f : A \to B$ to the monoid homomorphism $F(f) : F(A) \to F(B)$.

Example 6.1.1.9 For every category $\mathbf{C}$, we have the identity functor $I_{\mathbf{C}}$ which takes each $\mathbf{C}$-object and every $\mathbf{C}$-morphism to itself. More precisely we have that $I_{\mathbf{C}}(g \circ f) = g \circ f = I_{\mathbf{C}}(g) \circ I_{\mathbf{C}}(f)$, and $I_{\mathbf{C}}(id_A) = id_A = id_{I_{\mathbf{C}}(A)}$.

Example 6.1.1.10 The **duality** functor $(-)^{op}_{\mathbf{C}} : \mathbf{C} \to \mathbf{C^{op}}$ is a contravariant functor such that for each $A \in \mathbf{C}, (A)^{op} = A$ and for each $f : A \to B \in \mathbf{C}, (f : A \to B)^{op} = f^{op} : B \to A$. We omit subscripts in the duality functor whenever this is clear from context. Note that we also have the inverse $(-)^{op^{-1}}_{\mathbf{C}} : \mathbf{C^{op}} \to \mathbf{C}$ such that $(-)^{op} \circ (-)^{op^{-1}} = I_{\mathbf{C^{op}}}$ and $(-)^{op^{-1}} \circ (-)^{op} = I_{\mathbf{C}}$.

Example 6.1.1.11 (The dual functor) If $F : \mathbf{A} \to \mathbf{B}$ is a functor, its object function $A \mapsto F(A)$ and its mapping function $f \mapsto F(f)$, rewriten as $f^{op} \mapsto (F(f))^{op}_{\mathbf{B}} = ((-)^{op}_{\mathbf{B}} \circ F)(f)$, together define a functor from $\mathbf{A^{op}}$ to $\mathbf{B^{op}}$, which we denote as $F^{op} : \mathbf{A^{op}} \to \mathbf{B^{op}}$ Note that $(F^{op})^{op^{-1}} = F$ since we have:

$$
\begin{aligned}
(F^{op})^{op^{-1}} &= (-)^{op^{-1}} \circ F^{op} \\
&= (-)^{op^{-1}} \circ (-)^{op} \circ F \\
&= I_{\mathbf{C}} \circ F \\
&= F
\end{aligned}
$$

Remark 6.1.1.12 Recall that we have formulated a duality principle related to objects, morphisms, and categories. To form the dual of a categorical statement that involves functors, make the same statement, but with each category and each functor replaced by its dual. Then translate this back into a statement about the original categories and functors.

Example 6.1.1.13 Consider now a functor $F : \mathbf{C^{op}} \rightarrow \mathbf{D}$. By definition of a functor, it assigns to each object $C \in \mathbf{C^{op}}$ a object $F(C)$ in $\mathbf{D}$, and to each morphism $f^{op} : B \rightarrow A \in \mathbf{C^{op}}$ a morphism $F(f^{op}) : F(B) \rightarrow F(A)$ in $\mathbf{D}$, with $F(f^{op} \circ g^{op}) = F(f^{op}) \circ F(g^{op})$ whenever $f^{op} \circ g^{op}$ is defined in $\mathbf{C^{op}}$ (i.e, when $g \circ f$ is defined in $\mathbf{C}$). The functor so described may be expressed directly in terms of the original category $\mathbf{C}$ as a functor $F' : \mathbf{C} \rightarrow \mathbf{D}$ as follows:

$$F' = F \circ (-)^{op}$$

so that for each $f : A \rightarrow B$ in $\mathbf{C}$

$$
\begin{aligned}
F'(f : A \rightarrow B) &= (F \circ (-)^{op})(f : A \rightarrow B) \\
&= F((-)^{op}(f)) \\
&= F(f^{op} : B \rightarrow A) \\
&= F(f^{op}) : F(B) \rightarrow F(A) \\
&= F \circ (-)^{op}(f) : F \circ (-)^{op}(B) \circ F \circ (-)^{op}(A) \\
&= F'(f) : F'(B) \rightarrow F'(A)
\end{aligned}
$$

To see that $F' : \mathbf{C} \rightarrow \mathbf{D}$ preserve identities and compositions note that

$$
\begin{aligned}
F'(id_A) &= F \circ (-)^{op}(id_A) \\
&= F(id_A^{op}) \\
&= id_{F(A)} \\
&= id_{F \circ (-)^{op}(A)} \\
&= id_{F'(A)}
\end{aligned}
$$

while for each $f : A \rightarrow B, g : B \rightarrow C \in \mathbf{C}$

$$
\begin{aligned}
F'(g \circ f) &= (F \circ (-)^{op})(g \circ f) \\
&= F \circ ((g \circ f)^{op}) \\
&= F(f^{op} \circ g^{op}) \\
&= F(f^{op}) \circ F(g^{op}) \\
&= F \circ (-)^{op}(f) \circ F \circ (-)^{op}(g) \\
&= F'(f) \circ F'(g)
\end{aligned}
$$

Remark 6.1.1.14 An immediate consequence of the above discussion is that any contravariant functor $F : \mathbf{C} \to \mathbf{D}$ me be expressed covariantly as $F : \mathbf{C}^{\mathbf{op}} \to \mathbf{D}$ and vice-versa. Contravariant functors $F : \mathbf{C} \to \mathbf{D}$ which are expressed covariantly as $F : \mathbf{C}^{\mathbf{op}} \to \mathbf{D}$ are refered to as "contravariant fuctors" throughout the category theory literature ...

Example 6.1.1.15 For each subcategory $\mathbf{D}$ of $\mathbf{C}$ we have the inclusion functor $Inc_{\mathbf{D}} : \mathbf{D} \to \mathbf{C}$ which has the same behavior as the identity functor.

Example 6.1.1.16 Let $\mathbf{C}$ and $\mathbf{D}$ be categories and let K be an object of $\mathbf{D}$. The functor which takes every $\mathbf{C}$-object A to K and every $f : A \to B$ to id_K is called the **functor constantly** K, denoted $\Delta(K)$.

Example 6.1.1.17 Forgetful functors. Forgetting some of the structure in a category of structures and structure-preserving functions gives a functor called **underlying functor** or **forgetful functor**. The functor $U : \mathbf{Mon} \to \mathbf{Set}$ which sends each monoid $(M, \cdot, e)$ to the set M and each monoid homomorphism $h : (M, \cdot, e) \to (M', \cdot'.e')$ to the corresponding function $h : M \to M'$ on the underlying sets is a forgetful functor. Note that we have $U(id_{(M,\cdot,e)}) = id_M = id_{U(M,\cdot,e)}$ and $U(g \circ f) = g \circ f = U(g) \circ U(f)$ for every $f : (M, \cdot, e) \to (M', \cdot', e')$, $g : (M', \cdot', e') :\to (M'', \cdot'', e'')$ and $id_{(M,\cdot,e)} : (M, \cdot, e) \to (M, \cdot, e)$.

Example 6.1.1.18 Let $\mathbf{C}$ be a category with all binary products. Then each $\mathbf{C}$-object A determines a functor $(- \times A) : \mathbf{C} \to \mathbf{C}$, which takes each object B to $B \times A$, and each morphism $f : B \to C$ to $f \times id_A : B \times A \to C \times A$. The "-" is used to show where the argument object or morphism goes. A computer scientist might write $(- \times A)$ as $\lambda x.x \times A$. To see that this does really defines a functor, note that

$$
\begin{aligned}
&1.(- \times A)(id_A) = && \\
&= id_A \times id_A && \text{(definition of } (- \times A)) \\
&= id_{A\times A} && \text{(proposition 4.3.13)} \\
&= id_{(-\times A)(A)} && \text{(definition of } (- \times A))
\end{aligned}
$$

$$
\begin{aligned}
&2.(- \times A)(g \circ f) = && \\
&= (g \circ f) \times id_A && \text{(definition of } (- \times A)) \\
&= (g \circ f) \times (id_A \circ id_A) && \text{(identity)} \\
&= (g \times id_A) \circ (f \times id_A) && \text{(proposition 4.3.17)} \\
&= (- \times A)(g) \circ (- \times A)(f) && \text{(definition of } (- \times A))
\end{aligned}
$$

This functor is called the **right product functor**. Similarly, there is the **left product functor** $(A \times -) : \mathbf{C} \to \mathbf{C}$.

Example 6.1.1.19 If $\mathbf{C}$ is a category with all binary products, then we can (generalizing the above example) define the product functor $\times : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ as the pair of operations that takes $\langle A, B\rangle \in Ob_{\mathbf{C}\times\mathbf{C}}$ to $\times((A, B)) = A \times B$ and $\langle f, g\rangle \in \mathbf{C} \times \mathbf{C}(\langle A, B\rangle, \langle C, D\rangle)$ to $\times(\langle f, g\rangle) = f \times g : A \times B \to C \times D$. To see that this really defines a functor note that

$$
\begin{aligned}
&1. \times (\langle id_A, id_B\rangle) = && \\
&= id_A \times id_B && \text{(definition of } \times)
\end{aligned}
$$

$= id_{A \times B}$ (proposition 4.3.13)
$= id_{\times(\langle A,B \rangle)}$ (definition of $\times$)

$2. \times (\langle f,g \rangle \circ (h,k))$
$= \times((f \circ h, g \circ k))$ (composition in $\mathbf{C} \times \mathbf{C}$)
$= (f \circ h) \times (g \circ k)$ (definition of $\times$)
$= (f \times g) \circ (h \times k)$ (proposition 4.3.17)
$= \times(\langle f,g \rangle) \circ \times(\langle g,k \rangle)$ (definition of $\times$)

Example 6.1.1.20 Given two functors $F : \mathbf{A} \to \mathbf{B}, F' : \mathbf{A}' \to \mathbf{B}'$ we define the the functor $F \times F' : \mathbf{A} \times \mathbf{A}' \to \mathbf{B} \times \mathbf{B}'$ such that for each $\langle A, A' \rangle \in \mathbf{A} \times \mathbf{A}', \langle A, A' \rangle \mapsto \langle F(A), F(A') \rangle$, and for each $\langle f, g \rangle : \langle A, A' \rangle \to \langle B, B' \rangle, \langle f, g \rangle \mapsto \langle F(f), F(g) \rangle$.

Example 6.1.1.21 The diagonal funtor $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ takes each $\mathbf{C}$-object A to the object $\langle A, A \rangle$ in the category $\mathbf{C} \times \mathbf{C}$, and each $\mathbf{C}$-morphism $f : A \to B$ to the morphism $\langle f, f \rangle$ in the category $\mathbf{C} \times \mathbf{C}$. To see that this really defines a functor note that

$1.\Delta(id_A) =$
$= \langle id_A, id_A \rangle$ (definition of Δ)
$= id_{\langle A,A \rangle}$ (definition of identity in $\mathbf{C} \times \mathbf{C}$)
$= id_{\Delta(A)}$ (definition of Δ)

$1.\Delta(g \circ f) =$
$= \langle g \circ f, g \circ f \rangle$ (definition of Δ)
$= \langle g, g \rangle \circ \langle f, f \rangle$ (composition in $\mathbf{C} \times \mathbf{C}$)
$= \Delta(g) \circ \Delta(f)$ (definition of Δ)

Example 6.1.1.22 Let $\mathbf{C}$ be a category. Each $\mathbf{C}$-object A determines a functor $Hom_{\mathbf{C}}(A, -) : \mathbf{C} \to \mathbf{Set}$ (also denoted in the literaure of category theory as $\mathbf{C}(A, -) : \mathbf{C} \to \mathbf{Set}$). This functor takes each $\mathbf{C}$-object B to the set $Hom_{\mathbf{C}}(A, B)$ of morphisms from A to B and each $\mathbf{C}$-morphism $f : B \to C$ to the function $Hom_{\mathbf{C}}(A, f) : Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{C}}(A, C)$ defined for each $g : A \to B$ by:

$$Hom_{\mathbf{C}}(A, f)(g : A \to B) = f \circ g.$$

$Hom_{\mathbf{C}}(A, B)$ is called a **hom-functor**. The set $Hom_{\mathbf{C}}(A, B)$ is often called a **hom-set**. On the other hand, note that, for this definition makes sense, $\mathbf{C}$ must be a locally small category (see 2.3.52).

The following calculations shows that $Hom_{\mathbf{C}}(C, -)$ is a functor.

1. For an object A, $Hom_{\mathbf{C}}(C, id_A) : Hom_{\mathbf{C}}(C, A) \to Hom_{\mathbf{C}}(C, A)$ takes a morphism $f : C \to A$ to $id_A \circ f = f$; hence $Hom_{\mathbf{C}}(C, id_A) = id_A \circ \_ = id_{Hom_{\mathbf{C}}(C,A)}$ (i.e., that $id_A \circ \_$ is the identity for any $f \in Hom_{\mathbf{C}}(C, A)$).

2. Now suppose $f : A \to B$ and $g : B \to C$. Then for any morphism $k : C \to A$,

$(Hom_{\mathbf{C}}(C,g) \circ Hom_{\mathbf{C}}(C,f))(k) =$
$= Hom_{\mathbf{C}}(C,g)(Hom_{\mathbf{C}}(C,f)(k))$ (composition)
$= Hom_{\mathbf{C}}(C,g)(f \circ k)$ (definition of $Hom_{\mathbf{C}}(C,f)$)
$= g \circ (f \circ k)$ (definition of $Hom_{\mathbf{C}}(C,g)$))
$= (g \circ f) \circ k$ (associativity)
$= Hom_{\mathbf{C}}(C, g \circ f)(k)$ (definition of $Hom_{\mathbf{C}}(C, g \circ f)$)

Example 6.1.1.23 For a given object D, there's another covariant hom-functor $Hom_{\mathbf{C}}(-,D) : \mathbf{C}^{op} \to \mathbf{Set}$ (or equivalently, contravariant hom-functor $Hom_{\mathbf{C}}(-,D) \circ (-)^{op} : \mathbf{C} \to \mathbf{Set}$, see 6.1.1.13) which is defined for each object A by $Hom_{\mathbf{C}}(-,D)(A) = Hom_{\mathbf{C}}(A,D)$ and for each morphism $f^{op} : B \to A$ in $\mathbf{C^{op}}$ by

$$Hom_{\mathbf{C}}(-,D)(f^{op}) = Hom_{\mathbf{C}}(f^{op},D) : Hom_{\mathbf{C}}(B,D) \to Hom_{\mathbf{C}}(A,D)$$

such that for each $k : B \to D, Hom_{\mathbf{C}}(f^{op},D)(k) = k \circ f$ which is really a morphism from A to D (note that f in $k \circ f$ is a morphism in $\mathbf{C}$!). The following calculation shows that $Hom_{\mathbf{C}}(-,D)$ is a functor.

1. For each object A, $Hom_{\mathbf{C}}(id_A^{op}, D) : Hom_{\mathbf{C}}(A,D) \to Hom_{\mathbf{C}}(A,D)$ takes a morphism $f : A \to D$ to $f \circ id_A = f$; hence $Hom_{\mathbf{C}}(id_A^{op}, D) = \_ \circ id_A = id_{Hom_{\mathbf{C}}(A,D)}$.

2. Now suppose $f^{op} : B \to A$ and $g^{op} : C \to B$. Then $Hom_{\mathbf{C}}(f^{op} \circ g^{op}, D) : Hom_{\mathbf{C}}(C,D) \to Hom_{\mathbf{C}}(A,D)$. Then for any $k : C \to D$ we have:

$Hom_{\mathbf{C}}(f^{op} \circ g^{op}, D) \circ k =$
$= k \circ (g \circ f)$ (definition of $Hom_{\mathbf{C}}(f^{op} \circ g^{op}, D)$)
$= (k \circ g) \circ f$ (associativity)
$= Hom_{\mathbf{C}}(f^{op}, D)(k \circ g)$ (definition of $Hom_{\mathbf{C}}(f^{op}, D)(k \circ g)$)
$= Hom_{\mathbf{C}}(f^{op}, D)(Hom_{\mathbf{C}}(g^{op}, D)(k))$ (definition of $Hom_{\mathbf{C}}(g^{op}, D)(k)$)
$= Hom_{\mathbf{C}}(f^{op}, D) \circ Hom_{\mathbf{C}}(g^{op}, D)(k)$ (composition)

Example 6.1.1.24 The **powerset functor** $\wp : \mathbf{Set} \to \mathbf{Set}$ takes a set to the powerset $\wp(S) = \{S' | S' \subseteq S\}$ and each function $f : S \to T$ to the function $\wp(f) : \wp(S) \to \wp(T)$, where $\wp(f)(S') = \{f(s') | s' \in S'\}$ (see 2.1.6). To see that this really defines a functor note that for every $A' \subseteq A$ we have:

1.$\wp(id_A)(A') =$
$= \{id_A(a') | a' \in A'\}$ (definition of $\wp(id_A)$)
$= \{a' | a' \in A'\}$ (definition of id_A)
$= A'$ (by definition)
$= id_{\wp(A)}(A')$ (by definition of $id_{\wp(A)}$)

2. For every $f : A \to B$ $g : B \to C$, $A' \subseteq \wp(A)$ we have:

$$
\begin{aligned}
&\wp(g) \circ \wp(f)(A') = \\
&= \wp(g)(\wp(f)(A')) && \text{(composition)} \\
&= \wp(g)\{f(a')|a' \in A'\} && \text{(definition of } \wp(f)) \\
&= \{g(f(a'))|f(a') \in \{f(a')|a' \in A'\}\} && \text{(definition of } \wp(g)) \\
&= \{(g \circ f)(a')|a' \in A'\} && \text{(composition)} \\
&= \wp(g \circ f)(A') && \text{(definition of } \wp(g \circ f))
\end{aligned}
$$

Example 6.1.1.25 The two variable hom-functor $Hom_{\mathbf{C}}(-,-) : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{Set}$ ($\mathbf{C}^{op} \times \mathbf{C}$ is the product category in the sense of 3.4.3.1) takes a pair $\langle C, D\rangle$ of objects such that we have

$$Hom_{\mathbf{C}}(-,-)(\langle C, D\rangle) = Hom_{\mathbf{C}}(C, D)$$

and a pair of morphisms $\langle f^{op} : A \to C, g : B \to D\rangle \in \mathbf{C^{op}} \times \mathbf{C}$, (and hence $f : C \to A \in \mathbf{C}$), such that

$$Hom_{\mathbf{C}}(-,-)\langle f^{op}, g\rangle = Hom_{\mathbf{C}}(\langle f^{op}, g\rangle) : Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{C}}(C, D)$$

Now, for any $h : A \to B$, $Hom_{\mathbf{C}}\langle f^{op}, g\rangle(h) = g \circ h \circ f$. Note that this composition really defines a morphism from C to D.

To see that this definition makes, for a given category $\mathbf{C}$, $Hom_{\mathbf{C}}(-,-)$ a functor note that we have:

1. *Preservation of identities*: given $id_A^{op} : A \to A$ in $\mathbf{C}^{op}$ and hence, $id_B : B \to B$ in $\mathbf{C}$, we have that $Hom_{\mathbf{C}}(id_A^{op}, id_B) : Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{C}}(A, B)$. Thus, given any $f : A \to B$ in $\mathbf{C}$ we have

$$
\begin{aligned}
Hom_{\mathbf{C}}(id_A^{op}, id_B)(f) &= id_B \circ f \circ id_A && \text{(by definition)} \\
&= f && \text{(by definition)} \\
&= id_{Hom_{\mathbf{C}}(A,B)}(f) && (id_B \circ - \circ id_A \text{ is identity for } Hom_{\mathbf{C}}(A, B))
\end{aligned}
$$

2. *Preservation of composition*: Given $\langle f^{op}, f'\rangle, \langle g^{op}, g'\rangle \in \mathbf{C^{op}} \times \mathbf{C}, f^{op} : A \to B, g^{op} : B \to C$ (hence $f : B \to A, g : C \to B$ in $\mathbf{C}$), $f' : A' \to B', g' : B' \to C'$, we have that $Hom_{\mathbf{C}}(\langle g^{op}, g'\rangle \circ \langle f^{op}, f'\rangle) : Hom_{\mathbf{C}}(A, A') \to Hom_{\mathbf{C}}(C, C')$. Now, given any $h : A \to A'$ we have:

$$
\begin{aligned}
Hom_{\mathbf{C}}(\langle g^{op}, g'\rangle \circ \langle f^{op}, f'\rangle)(h) &= Hom_{\mathbf{C}}(\langle g^{op} \circ f^{op}, g' \circ f'\rangle)(h) && \text{(composition in } \mathbf{C^{op}} \times \mathbf{C}) \\
&= (g' \circ f') \circ h \circ (f \circ g) && \text{(by definition)} \\
&= g' \circ (f' \circ h \circ f) \circ g && \text{(associativity)} \\
&= Hom_{\mathbf{C}}(g^{op}, g')(f' \circ h \circ f) && \text{(by definition)} \\
&= Hom_{\mathbf{C}}(g^{op}, g') \circ Hom_{\mathbf{C}}(f^{op}, f')(h) && \text{(by definition)}
\end{aligned}
$$

Note that this functor can be described contravariantly in the first argument by the composite functor

$$Hom_{\mathbf{C}}(-,-) \circ ((-)^{op} \times I_{\mathbf{C}}) : \mathbf{C} \times \mathbf{C} \to \mathbf{Set}$$

Example 6.1.1.26 There is another "variant" of the powerset functor, namely $\wp^{\sharp} : \mathbf{Set}^{op} \to \mathbf{Set}$, which takes a set S to its powerset $\wp(S)$, and a set function $f : A \to B$ (that is, a morphism from B to A in $\mathbf{Set}^{op}$) to the inverse image $f^{-1} : \wp(B) \to \wp(A)$, which is defined for all $B' \subseteq \wp(B)$ by

$$f^{-1}(B') =_{def} \{a \in A | f(a) \in B'\}$$

To see that this definition really provides a functor, we calculate as follows:

1. Preservation of identities. Now, for all $A' \in \wp(A)$ we have

$$\begin{array}{rcll} \wp^{\sharp}(id_A)(A') & = & id_A^{-1}(A) & \text{(by definition of } \wp^{\sharp}) \\ & = & \{a \in A | id_A(a) \in A'\} & \text{(by definition of } id_A^{-1}) \\ & = & A' & \text{(by definition)} \\ & = & id_{\wp^{\sharp}(A)}(A') & \text{(by definition of } id_{\wp^{\sharp}}) \end{array}$$

2. Preservation of composition. Thus, given $C' \in \wp^{\sharp}(C)$ and two composable morphisms $f : A \to B, g : B \to C$, we have:

$$\begin{array}{rcll} \wp^{\sharp}(f) \circ \wp^{\sharp}(g)(C') & = & (f^{-1} \circ g^{-1})(C') & \text{(by definition of } \wp^{\sharp}) \\ & = & f^{-1}(g^{-1}(C')) & \text{(composition)} \\ & = & f^{-1}(\{b \in B | g(b) \in C'\}) & \text{(by definition of } g^{-1}) \\ & = & \{a \in A | f(a) \in \{b \in B | g(b) \in C'\}\} & \text{(by definition of } f^{-1}) \\ & = & \{a \in A | g(f(a)) \in C'\} & \text{(take } b =_{def} f(a)) \\ & = & \{a \in A | g \circ f(a)) \in C'\} & \text{(composition)} \\ & = & (g \circ f)^{-1}(C') & \text{(by definition of } (g \circ f)^{-1}) \\ & = & \wp^{\sharp}(g \circ f)(C') & \text{(by definition of } \wp^{\sharp}) \end{array}$$

Example 6.1.1.27 The category **Cat** has small categories as objects and functors as morphisms. The identity functor for each category $\mathbf{C}$ is the one presented in 6.1.1.9, and the composition of functors the one established in proposition 6.1.1.

Example 6.1.1.28 If you forget you can compose morphisms in a category and you forget which morphisms are the identities, then would have remembered only that the category is a graph. This gives an underlying set functor $U : \mathbf{Cat} \to \mathbf{Graph}$, since every functor is a graph morphism although not vice versa.

6.1.2 Types of functors

Definition 6.1.2.1 *Any functor $F : \mathbf{C} \to \mathbf{D}$ induces a set mapping*

$$Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{D}}(F(A), F(B))$$

for each pair of objects A and B of $\mathbf{C}$. This mapping takes a morphism $f : A \to B$ to $F(f) : F(A) \to F(B)$.

Definition 6.1.2.2 *A functor $F : \mathbf{C} \to \mathbf{D}$ is* **faithful** *if the induced mapping is injective on every hom set. Thus, if $f : A \to B$ and $g : A \to B$ are different morphisms then $F(f) \neq F(g)$.*

Example 6.1.2.3 Underlying functors are typically faithful. Two different monoid homomorphisms between the same two monoids must be different set functions

On the other hand, consider the set $\{0, 1, 2\}$. It has two different monoid structures via addition and multiplication (mod 3), but the two corresponding identity homomorphisms are the same as set functions (have the same underlying function). Thus underlying functors need not to be injective.

Example 6.1.2.4 The free monoid functor is faithful, because if $f \neq g : S \to T$ then $F(f) \neq F(g) : S^* \to T^*$ since on strings of length 1, $F(f)$ is essentially the same as f.

Definition 6.1.2.5 *A functor $F : \mathbf{C} \to \mathbf{D}$ is* **full** *if the induced mapping is surjective for every hom set.*

Example 6.1.2.6 A full functor need not be surjective on either objects or arrows. A full subcategory is exactly one whose embedding is a full and faithful functor.

Example 6.1.2.7 Consider the following diagrams (we omit identities):

$$\begin{tikzcd} A \arrow[d, shift right, "f"'] & B \arrow[d, shift right, "g"'] & & C \arrow[d, shift right, "h"'] \\ X \arrow[u, shift right, "f'"'] & Y \arrow[u, shift right, "g'"'] & & Z \arrow[u, shift right, "h'"'] \end{tikzcd}$$

The functor $F : \mathbf{C} \to \mathbf{D}$ which takes A and B to C, X and Y to Z, f, g to h, and f', g' to h' is not full. This is because $Hom_{\mathbf{D}}(C, C) = Hom_{\mathbf{D}}(F(A), F(B))$ has one arrow, namely the identity, while $Hom_{\mathbf{C}}(A, B)$ is the empty set. However, it is faithful (although not injective), since two arrows between the same two objects do not get identified.

6.1.3 Preservation of properties

Definition 6.1.3.1 A functor $F : \mathbf{C} \to \mathbf{D}$ **preserves** a property P of morphisms if whenever f has a property P, so does $F(f)$.

Proposition 6.1.3.2 *Every functor preserves isomorphisms.*

Proof: If $f : A \to B$ is an isomorphism with an inverse f^{-1} we have:

$$
\begin{aligned}
& F(f) \circ F(f^{-1}) = \\
&= F(f \circ f^{-1}) && (F \text{ is a functor}) \\
&= F(id_B) && (f \text{ is an isomorphism}) \\
&= id_{F(B)} && (F \text{ is a functor})
\end{aligned}
$$

Similarly, we can show that $F(f^{-1}) \circ F(f) = id_{F(A)}$.

$\square$

Proposition 6.1.3.3 *Every functor preserves sections.*

Proof: If $f : A \to B$ is a section then there exists a $h : B \to A$ such that $h \circ f = id_A$. Hence,

$$
\begin{aligned}
& F(h) \circ F(f) = \\
&= F(h \circ f) && (F \text{ is a functor}) \\
&= F(id_A) && (f \text{ is a section}) \\
&= id_{F(A)} && (F \text{ is a functor})
\end{aligned}
$$

$\square$

Proposition 6.1.3.4 *Every functor preserves retractions.*

Proof: $\boxed{\text{D}}$

6.1.4 Comma Categories

There are many instances of comma categories in computer science. One whole family of examples arises from the fact that indexed families of sets form comma categories. Indexed families of sets occurs frequently, for example, in the mathematical study of data structures (see our discussion on (algebraic) signatures and specifications in chapter 1). Before introducing the technical machinery, we make here an explicit construction of a special kind of comma category. Let S be a set (a set of indices) and let **Set** be our usual category of sets and functions. Let now $\mathbf{Set} \downarrow S$ (such denotation will be clear later on) be the category where the objects are functions $c : A \to S$, where $A \in \mathbf{Set}$, which assigns to each $a \in A$ its index $c(a)$. Thus, the corresponding indexed family is $(c^{-1}(s) | s \in S)$. The morphisms of $\mathbf{Set} \downarrow S$ from $c : A \to S$ to $d : B \to S$ are functions $f : A \to B$ such that $d \circ f = c$, i.e., such that the following diagram commutes

$$
\begin{tikzcd}
A \arrow[rr, "f"] \arrow[dr, "c"'] & & B \arrow[dl, "d"] \\
& S &
\end{tikzcd}
$$

Note that these functions $f : A \to B$ correspond bijectively with families $(f_s : c^{-1}(s) \to d^{-1}(s) | s \in S)$ with f_s being the restriction of f to $c^{-1}(s)$, and f being the union of all the f_s, and the above commutativity says essentially that the changes of names from A to B should preserve indexes or in terms of computer science, types. The above example motivate us to present the following

Definition 6.1.4.1 *If $F : \mathbf{A} \to \mathbf{C}$ and $G : \mathbf{B} \to \mathbf{C}$ are functors, then the* **comma-category** *$(F \downarrow G)$ is the category whose objects are triples (A, f, B) with $A \in Ob(\mathbf{A})$, $B \in Ob(\mathbf{B})$, and $f : F(A) \to G(B) \in \mathbf{C}$; whose morphisms from (A, f, B) to (A', f', B') are pairs (a, b) with $a : A \to A' \in \mathbf{A}$, and $b : B \to B' \in \mathbf{B}$ such that the square*

$$\begin{tikzcd} FA \arrow[r, "Fa"] \arrow[d, "f"'] & FA' \arrow[d, "f'"] \\ GB \arrow[r, "Gb"'] & GB' \end{tikzcd}$$

commutes; whose identities are $id_{(A,f,B)} = (id_A, id_B)$; and whose composition is defined componentwise, i.e., by $(a', b') \circ (a, b) = (a' \circ a, b' \circ b)$.

In the following examples of comma-categories let $\Delta(K) : \mathbf{1} \to \mathbf{A}$ be the functor constantly K and $I_{\mathbf{A}} : \mathbf{A} \to \mathbf{A}$ the identity functor on A.

Example 6.1.4.2 The comma-category $(I_{\mathbf{A}} \downarrow \Delta(K))$, also denoted by $(\mathbf{A} \downarrow K)$, is called the **category of objects over** K as follows:

1. An object in $(\mathbf{A} \downarrow K)$ is a triple $(A, f, 1)$ where $A \in Ob(\mathbf{A})$, $1 \in Ob(\mathbf{1})$ and $f : I_{\mathbf{A}}(A) \to \Delta(K)(1) = A \to K \in Mor(\mathbf{A})$.
2. A morphism from $(A, f, 1)$ to $(A', g, 1)$ is a pair (a, id_1) where $a : A \to A'$ and $id_1 : 1 \to 1$ such that the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & A' \arrow[d, "g"] \\ K \arrow[r, "id_K"'] & K \end{tikzcd}$$

3. For any $(\mathbf{A} \downarrow K)$-morphisms $(a, id_1) : (A, f, 1) \to (A', g, 1)$ and $(b, id_1) : (A', g, 1) \to (A'', h, 1)$ the composition $(b, id_1) \circ (a, id_1) : (A, f, 1) \to (A'', h, 1)$ is defined (componentwise) as $(b \circ a, id_1 \circ id_1)$, or such that the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "f"'] \arrow[dr, phantom, "(1)"] & A' \arrow[r, "b"] \arrow[d, "g"] \arrow[dr, phantom, "(2)"] & A'' \arrow[d, "h"] \\
K \arrow[r, "id_K"'] & K \arrow[r, "id_K"'] & K
\end{tikzcd}$$

To see that it really commutes note that

$h \circ b \circ a =$
$= id_K \circ g \circ a$ ((2) is a $(\mathbf{A} \downarrow K)$-morphism)
$= id_K \circ id_K \circ f$ ((1) is a $(\mathbf{A} \downarrow K)$-morphism)
$= f$ (identity)

Moreover, for any $(a, id_1) : (A, f, 1) \to (A', g, 1)$, $(b, id_1) : (A', g, 1) \to (A'', h, 1)$, $(c, id_1) : (A'', h, 1) \to (A''', i, 1)$ we have that $(c, id_1) \circ ((b, id_1) \circ (a, id_1)) = ((c, id_1 \circ (b, id_1)) \circ (a, id_1)$ by the commutativity of the following diagram:

$$\begin{aligned}
(c, id_1) \circ ((b, id_1) \circ (a, id_1)) &= (c, id_1) \circ (b \circ a, id_1 \circ id_1) \\
&= (c \circ (b \circ a), id_1 \circ (id_1 \circ id_1)) \\
&= ((c \circ b) \circ a, id_1)
\end{aligned}$$

4. For every $(\mathbf{A} \downarrow K)$-morphism $(a, id_1) : (A, f, 1) \to (A', g, 1)$ we have identities (id_A, id_1) and $(id_{A'}, id_1)$ such that $(a, id_1) \circ (id_A, id_1) = (a, id_1)$ and $(id_{A'}, id_1) \circ (a, id_1) = (a, id_1)$.

Remark 6.1.4.3 Usually, textbooks on category theory always omit the identity arrows id_K, since they are just identities in relation to composition. Thus, the diagram defining a morphism in $(\mathbf{A} \downarrow K)$ is usually represented like the following one:

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "f"'] & A' \arrow[dl, dotted, "g"] \\
K &
\end{tikzcd}$$

Example 6.1.4.4 The comma-category $(\Delta(K) \downarrow I_{\mathbf{A}})$, also denoted by $(K \downarrow \mathbf{A})$, is called the **category of objects under** K. The verification that $(K \downarrow \mathbf{A})$ is indeed a category is analogous to the one done in 6.1.4.2. A morphism in $(K \downarrow \mathbf{A})$ from $(1, f, A)$ to $(1, g, B)$ is an $\mathbf{A}$-morphism $a : A \to A'$ such that the following diagram commutes:

$$\begin{tikzcd}
& K \arrow[dl, dotted, "f"'] \arrow[d, "g"] \\
A \arrow[r, "a"'] & A'
\end{tikzcd}$$

Example 6.1.4.5 The comma category $(I_{\mathbf{A}} \downarrow I_{\mathbf{A}})$, also denoted $\mathbf{A}^2$ is called the arrow category of $\mathbf{A}$. In this category a morphism $(a, b) : (A, f, B) \to (A', f', B')$ is defined by the commutativity of the following diagram:

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & A' \arrow[d, "f'"] \\ B \arrow[r, "b"'] & B' \end{tikzcd}$$

However, this is just the category $\mathbf{C}^{\to}$ we have presented in 3.4.3.4!

6.1.5 Diagrams

Here we give a more rigorous definition of the concept of a diagram, using just categorical concepts.

Definition 6.1.5.1

1. A **diagram** in a category A is a functor $D : \mathbf{I} \to \mathbf{A}$ with codomain $\mathbf{A}$. The domain $\mathbf{I}$ is called the **scheme** of the diagram.
2. A diagram with a small (or finite) scheme is said to be **small** (or **finite**).

Example 6.1.5.2 A diagram in $\mathbf{A}$ with a discrete scheme is essentially just a family of $\mathbf{A}$-objects.

Example 6.1.5.3 A diagram in $\mathbf{A}$ with the scheme

$$\begin{tikzcd} \bullet \arrow[r, shift left] \arrow[r, shift right] & \bullet \end{tikzcd}$$

is essentially just a pair of $\mathbf{A}$-morphisms with common domain and codomain.

Example 6.1.5.4 Consider the scheme represented by (1). Then if $D : \mathbf{I} \to \mathbf{A}$ is a functor, the corresponding diagram in A can be represented by (2).

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "(1)"] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(2)"] & B \arrow[d, "i"] \\ C \arrow[r, "h"'] & D \end{tikzcd}$$

6.2 Natural transformations

Having originally defined categories as collections of objects with morphisms between them, by introducing functors we took a step up the ladder of abstraction to consider categories as objects, with functors as morphisms between them. Now we climb even higher, to regard functors themselves as objects!

Given two categories $\mathbf{C}$ and $\mathbf{D}$ we are going to construct a category, denoted $\mathbf{D}^{\mathbf{C}}$, whose objects are the functors from $\mathbf{C}$ to $\mathbf{D}$. We need a definition of morphism from one functor to another. Taking $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{C} \to \mathbf{D}$, we think of the functors F and G as providing different "pictures" of $\mathbf{C}$ inside $\mathbf{D}$. A reasonably intuitive idea of "transformation" from F to G comes if we imagine ourselves trying to superimpose or "slide" the F-picture onto the G-picture, i.e., we use the structure of $\mathbf{D}$ to translate the former into the latter. This could be done by assigning to each $\mathbf{C}$-object A a morphism in $\mathbf{D}$ from the F-image of A to the G-image of A. Denoting this morphism by η_A, we have $\eta_A : F(A) \to G(A)$. However we still need that this process be "structure-preserving" .

6.2.1 Basic concepts and examples

The formal definition is as follows:

Definition 6.2.1.1 *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be categories and* F *and* G *be functors from* $\mathbf{C}$ *to* $\mathbf{D}$. *A* **natural transformation** η *from* F *to* G*, written* $\eta : F \dot{\to} G$*, is a function that assigns to every* $\mathbf{C}$*-object* A *a* $\mathbf{D}$*-morphism* $\eta_A : F(A) \to G(A)$ *such that for any* $\mathbf{C}$*-arrow* $f : A \to B$ *the diagram on the right commutes in* $\mathbf{D}$*:*

$$\begin{tikzcd} A \arrow[d, "f"'] \\ B \end{tikzcd} \qquad \begin{tikzcd} F(A) \arrow[r, "\eta_A"] \arrow[d, "F(f)"'] \arrow[dr, phantom, "="] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\eta_B"] & G(B) \end{tikzcd}$$

Example 6.2.1.2 For every functor $F : \mathbf{C} \to \mathbf{D}$, the components of the identity natural transformation $\iota_F : F \dot{\to} F$ are the identity morphisms of the objects in the image of F, i.e., $\iota_F(A) = id_{F(A)}$. To see this, note that the following diagram commutes for every morphism $f : A \to B$, i.e. $F(f) \circ \iota_F(A) = F(f) \circ id_{F(A)} = F(f) = id_{F(B)} \circ F(f) = \iota_F(B) \circ F(f)$.

$$\begin{tikzcd} A \arrow[d, "f"'] \\ B \end{tikzcd} \qquad \begin{tikzcd} F(A) \arrow[r, "\iota_F(A)"] \arrow[d, "F(f)"'] \arrow[dr, phantom, "="] & F(A) \arrow[d, "F(f)"] \\ F(B) \arrow[r, "\iota_F(B)"] & F(B) \end{tikzcd}$$

Example 6.2.1.3 Recall the *List* functor of example 6.1.1.7. Let rev be the "reverse" operation on lists, i.e., $rev_S : List(S) \to List(S)$ takes any list with elements in S to its reverse. For example,

$$rev_S([5,6,7]) = [7,6,5]$$

Indeed, we can apply any mapping whatsoever to the individual elements of the argument to rev, even one that change their types. If $f : S \to T$, then

$$\begin{array}{lr} List(f) \circ rev_S([5,6,7]) = & \\ = List(f)(rev_S([5,6,7])) & \text{(composition)} \\ = List(f)([7,6,5]) & \text{(by definition of } rev_S) \\ = [f(7), f(6), f(5)] & \text{(by definition of } List(f)) \end{array}$$

while

$$\begin{array}{lr} rev_T \circ List(f)([5,6,7]) = & \\ = rev_T(List(f)([5,6,7])) & \text{(composition)} \\ = rev_T([f(5), f(6), f(7)]) & \text{(definition of } List(f)) \\ = [f(7), f(6), f(5)] & \text{(by definition of } rev_T) \end{array}$$

In general, if $f : S \to T$, then

$$rev_T \circ List(f) = List(f) \circ rev_S$$

which states the commutativity of the following diagram:

$$\begin{tikzcd} S \arrow[d, "f"'] & List(S) \arrow[r, "rev_S"] \arrow[d, "List(f)"'] \arrow[dr, phantom, "="] & List(S) \arrow[d, "List(f)"] \\ T & List(T) \arrow[r, "rev_T"] & List(T) \end{tikzcd}$$

But this says exactly that rev is a natural transformation.

Example 6.2.1.4 $\mathbf{C}^{\to}$. Consider the category $\mathbf{2}$ of example 2.3.48. A functor $F : \mathbf{2} \to \mathbf{C}$ comprises two $\mathbf{C}$-objects $F(A)$ and $F(B)$, and a function $F(f) = g : F(A) \to F(B)$. Thus F is essentially a morphism g in $\mathbf{C}$, i.e., an object in $\mathbf{C}^{\to}$. Now given another such functor G, construed as $h : G(A) \to G(B)$, then $\eta : F \dot{\to} G$ has components η_A and η_B that make the following diagram commute:

$$\begin{tikzcd}
A \arrow[d, "f"'] & F(A) \arrow[r, "\eta_A"] \arrow[d, "F(f)=g"'] \arrow[dr, phantom, "="] & G(A) \arrow[d, "G(f)=h"] \\
B & F(B) \arrow[r, "\eta_B"] & G(B)
\end{tikzcd}$$

Compare this description with the one given in 3.4.3.4.

6.2.2 Vertical composition of natural transformations

Definition and proposition 6.2.2.1 *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be categories. Let* F, G *and* H *be functors from* $\mathbf{C}$ *to* $\mathbf{D}$. *Let* $\sigma : F \dot{\to} G$ *and* $\tau : G \dot{\to} H$ *be natural transformations. We define the* **composite** *natural transformation* $(\tau \circ \sigma) : F \dot{\to} H$ *by* $(\tau \circ \sigma)_A = \tau_A \circ \sigma_A$.

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

$$\begin{tikzcd}
A \arrow[d, "f"'] & F(A) \arrow[r, "\sigma_A"] \arrow[d, "F(f)"'] & G(A) \arrow[r, "\tau_A"] \arrow[d, "G(f)"] & H(A) \arrow[d, "H(f)"] \\
B & F(B) \arrow[r, "\sigma_B"] & G(B) \arrow[r, "\tau_B"] & H(B)
\end{tikzcd}$$

To see this note that

$\tau_B \circ \sigma_B \circ F(f) =$
$= \tau_B \circ G(f) \circ \sigma_A$ (by naturality of σ)
$= H(f) \circ \tau_A \circ \sigma_A$ (by naturality of τ)

$\square$

Remark 6.2.2.2 Another possible formulation for composition will be given in 6.2.5.1. Pictorially, the vertical composition of two natural transformations σ and τ is usually represented by the following diagram:

$$\begin{tikzcd}
\mathbf{C} \arrow[r, bend left=50, "F"] \arrow[r, "G" description] \arrow[r, bend right=50, "H"'] & \mathbf{D}
\end{tikzcd} \quad (\Downarrow\sigma,\ \Downarrow\tau)$$

Example 6.2.2.3 The category $\mathbf{D}^{\mathbf{C}}$ has functors from $\mathbf{C}$ to $\mathbf{D}$ as objects and natural transformations bewteen such functors as morphisms. The composition of natural transformations is the one defined in 6.2.2.1 and the identity natural transformation the one defined 6.2.1.2. Associativity and identity are inherited from $\mathbf{D}$, since the components of such natural transformations are morphisms in $\mathbf{D}$.

Definition and proposition 6.2.2.4 *Let* $\mathbf{C}$ *and* $\mathbf{J}$ *be categories (*$\mathbf{J}$ *for index or scheme category). The* diagonal functor $\Delta : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ *sends each object* $A \in \mathbf{C}$ *to the constant functor* $\Delta(A) : \mathbf{J} \to \mathbf{C}$ *- the functor which has the value* A *at each object* $I \in \mathbf{J}$ *and the value* id_A *at each morphism of* $\mathbf{J}$*, or more precisely,* $\Delta(A)(I) = A, \Delta(A)(f)$*, respectively. If* $f : A \to A'$ *is a morphism in* $\mathbf{C}$*, then* $\Delta(f)$ *is the natural transformation* $\Delta(f) : \Delta(A) \to \Delta(A')$*, which has the value* f *at each object* $I \in \mathbf{J}$*.*

Proof: First note that, given an object $A \in \mathbf{C}$, the constant functor $\Delta(A)$, by its definition (constantly A) can be identified with the object A itself. By the same token, given a morphism $f : A \to A'$, the natural transformation $\Delta(f) : \Delta(A) \to \Delta(A')$ can be identified with the morphism f itself. Now we have $\Delta(id_A) = id_A = id_{\Delta(A)}$ and given $f : A \to B, g : B \to C$ in $\mathbf{C}$, $\Delta(g \circ f) = g \circ f = \Delta(g) \circ \Delta(f)$, as required to show, respectively, preservation of identities and composition. $\square$

Remark 6.2.2.5 Recal the diagonal functor $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ of example 6.1.1.21. Note that if $\mathbf{J}$ is the scheme categorie $\bullet \quad \bullet$ (i.e., a category with only two objects), then a functor from $\mathbf{J}$ to $\mathbf{C}$ is just a pair of objects of $\mathbf{C}$. Therefore, the functor category $\mathbf{C}^{\mathbf{J}}$ is isomorphic to the product category $\mathbf{C} \times \mathbf{C}$.

6.2.3 Natural isomorphism

Definition 6.2.3.1 *Let* F, G *be functors from* $\mathbf{C}$ *to* $\mathbf{D}$*. If each component* η_A *of* $\eta : F \dot{\to} G$ *is an isomorphism in* $\mathbf{D}$*, we call* η *a* **natural isomorphism***.*

Remark 6.2.3.2 In case η is a natural isomorphism, we can interpret this as meaning that the F-picture and the G-picture of $\mathbf{C}$ look the same in $\mathbf{D}$. Each $\eta_A : F(A) \dot{\to} G(A)$ then has an inverse $\eta_A^{-1} : G(A) \to F(A)$, and each such inverse is the component of a natural isomorphism $\eta^{-1} : G \dot{\to} F$ (proposition 6.2.3.3).

Proposition 6.2.3.3 *Suppose* $F : \mathbf{C} \to \mathbf{D}$ *and* $G : \mathbf{C} \to \mathbf{D}$ *are functors and* $\eta : F \dot{\to} G$ *is natural isomorphism. Then there is a unique natural transformation* $\eta^{-1} : G \dot{\to} F$ *such that the composites* $\eta \circ \eta^{-1} = \iota_G$ *and* $\eta^{-1} \circ \eta = \iota_F$*. Hence* η *is an isomorphism of the category* $\mathbf{D}^{\mathbf{C}}$*.*

Proof: First we define $\eta_A^{-1} = (\eta_A)^{-1}$, which is the only possible definition. Now we have to show is that the commutativity of (A) implies that of (B).

$$
\begin{tikzcd}
A \arrow[d, "f"'] \\
B
\end{tikzcd}
\qquad
\begin{tikzcd}
F(A) \arrow[r, "\eta_A"] \arrow[d, "F(f)"'] \arrow[dr, phantom, "\text{(A)}"] & G(A) \arrow[d, "G(f)"] \\
F(B) \arrow[r, "\eta_B"] & G(B)
\end{tikzcd}
\qquad
\begin{tikzcd}
F(A) \arrow[d, "F(f)"'] \arrow[dr, phantom, "\text{(B)}"] & G(A) \arrow[l, "(\eta_A)^{-1}"'] \arrow[d, "G(f)"] \\
F(B) & G(B) \arrow[l, "(\eta_B)^{-1}"']
\end{tikzcd}
$$

To see this, note that

$F(f) \circ (\eta_A)^{-1}$
$= id_{F(B)} \circ (F(f) \circ (\eta_A)^{-1})$ (identity)
$= ((\eta_B)^{-1} \circ \eta_B) \circ (F(f) \circ (\eta_A)^{-1})$ (η is a natural isomorphism)
$= (\eta_B)^{-1} \circ (\eta_B \circ F(f)) \circ (\eta_A)^{-1}$ (associativity)
$= (\eta_B)^{-1} \circ (G(f) \circ \eta_A) \circ (\eta_A)^{-1}$ (by naturality of η)
$= (\eta_B)^{-1} \circ G(f) \circ (\eta_A \circ (\eta_A)^{-1})$ (associativity)
$= (\eta_B)^{-1} \circ G(f) \circ id_{G(A)}$ (η is a natural isomorphism)
$= (\eta_B)^{-1} \circ G(f)$ (identity)

□

Example 6.2.3.4 In **Set**, we have trivially $A \cong 1 \times A$ (this is an instance of proposition 4.3.18). This isomorphism is a natural one between the identity functor $I_{\mathbf{Set}} : \mathbf{Set} \to \mathbf{Set}$ and the left-product functor $(1 \times -) : \mathbf{Set} \to \mathbf{Set}$ (see example 6.1.1.18). Given $f : A \to B$ then the diagram

$$
\begin{tikzcd}
A \arrow[d, "f"'] \\
B
\end{tikzcd}
\qquad
\begin{tikzcd}
I_{\mathbf{Set}}(A) \arrow[r, "\eta_A"] \arrow[d, "I_{\mathbf{Set}}(f)"'] \arrow[dr, phantom, "="] & 1 \times A \arrow[d, "id_1 \times f"] \\
I_{\mathbf{Set}}(B) \arrow[r, "\eta_B"] & 1 \times B
\end{tikzcd}
$$

must commute. We define $\eta_A(a) = \langle 0, a \rangle$ where $1 = \{0\}$ and similarly for η_B. This clearly defines a bijection. To see it is natural, note that for any $a \in A$ we have:

$(id_1 \times f) \circ \eta_A(a) =$
$= (id_1 \times f)(\eta_A(a))$ (composition)
$= (id_1 \times f)(\langle 0, a \rangle)$ (by definition of η)
$= \langle 0, f(a) \rangle$ (by definition of $id_1 \times f$)
$= \eta_B(f(a))$ (by definition of η)
$= \eta_B \circ f(a)$ (composition)

Example 6.2.3.5 Again in **Set**, we have $A \times B \cong B \times A$ by the "twist" map $tw_B : A \times B \to B \times A$ given by the rule $tw_B(\langle a, b \rangle) = (\langle b, a \rangle)$. But this function can be seen as a natural isomorphism between the left-product functor $(A \times -) : \mathbf{Set} \rightarrow \mathbf{Set}$ and the right-product functor $(- \times A) : \mathbf{Set} \rightarrow \mathbf{Set}$. What we have to show is that the diagram

$$\begin{tikzcd} B \arrow[d, "f"'] & A \times B \arrow[r, "tw_B"] \arrow[d, "id_A \times f"'] \arrow[dr, phantom, "="] & B \times A \arrow[d, "f \times id_A"] \\ C & A \times C \arrow[r, "tw_C"] & C \times A \end{tikzcd}$$

commutes, showing then that the bijections tw_B are the components of a natural isomorphism from $(A \times -)$ to $(- \times A)$. To see this note that for any $\langle a, b\rangle \in A \times B$ we have:

$tw_C \circ (id_A \times f)(\langle a, b\rangle) =$
$= tw_C((id_A \times f)(\langle a, b\rangle))$ (composition)
$= tw_C(\langle a, f(b)\rangle)$ (definition of $id_A \times f$)
$= \langle f(b), a\rangle$ (by definition of tw_C)
$= (f \times id_A)(\langle b, a\rangle)$ (by definition of $f \times id_A$)
$= (f \times id_A) \circ tw_B(\langle a, b\rangle)$ (by definition of tw_B)

6.2.4 Cones and cocones as natural transformations

Example 6.2.4.1 Consider now a diagram as a functor from a scheme category $\mathbf{I}$ to a category $\mathbf{C}$ (see 6.1.5.1). First, note that any $\mathbf{C}$-object C is the image of a functor (diagram) constantly C, $\Delta(C) : \mathbf{I} \rightarrow \mathbf{C}$, and therefore that $\Delta(C)$ can be considered as a (degenerated) diagram of type $\mathbf{I}$ in $\mathbf{C}$. Once diagrams are defined as functors, it makes sense to consider natural transformations between these diagrams. In this way, if $D, D' : \mathbf{I} \rightarrow \mathbf{C}$ are two diagrams (of type $\mathbf{I}$), a natural transformation from D to D' is a family of $\mathbf{C}$-morphisms τ_I indexed by the objects of $\mathbf{I}$ such that, for each morphism $e : I \rightarrow J$ in $\mathbf{I}$, the following diagram commutes:

$$\begin{tikzcd} I \arrow[d, "e"'] & D_I = D(I) \arrow[r, "\tau_I"] \arrow[d, "D_e = D(e)"'] & D'(I) = D'_I \arrow[d, "D'(e) = D'_e"] \\ J & D_J = D(J) \arrow[r, "\tau_J"] & D'(J) = D'_J \end{tikzcd}$$

This means that $\tau : D \rightarrow D'$ is a just a natural transformation.

Now, a cone L for a diagram $D : \mathbf{I} \rightarrow \mathbf{C}$ is then a natural transformation from the diagram constantly L, $\Delta(L) : \mathbf{I} \rightarrow \mathbf{C}$, to the diagram $D : \mathbf{I} \rightarrow \mathbf{C}$ such that the following diagram commutes:

$$
\begin{tikzcd}
I \arrow[d, "e"'] & \Delta(L)(I) = L \arrow[r, "\tau_I"] \arrow[d, "\Delta(L)(e) = id_L"'] & D_I \arrow[d, "D_e"] \\
J & \Delta(L)(J) = L \arrow[r, "\tau_J"] & D_J
\end{tikzcd}
$$

Dually, a cocone C for a diagram $D : \mathbf{I} \to \mathbf{C}$ is a natural transformation from $D : \mathbf{I} \to \mathbf{C}$ to the "diagram" constantly C, $\Delta(C) : \mathbf{I} \to \mathbf{C}$.

Remark 6.2.4.2

1. If the cone, describe above as a natural transformation, is a limiting cone in the sense of 5.1.2, then we may write from now on $(\Delta(L), \tau : \Delta(L) \dot{\to} D)$, or equivalently, $(L, \tau : L \dot{\to} D)$, to denote a limit of the diagram $D : \mathbf{I} \to \mathbf{C}$ in the category $\mathbf{C}$.
2. Dually, $(\Delta(C), \tau : D \dot{\to} \Delta(C))$, or equivalently, $(C, \tau : D \dot{\to} C)$, to denote a colimit of the diagram $D : \mathbf{I} \to \mathbf{C}$ in the category $\mathbf{C}$.

6.2.5 Horizontal composition of natural transformations

In 6.2.2.1 we have defined a "vertical" composite $\tau \circ \sigma$ of two natural transformations. There is another "horizontal" composition for natural transformations. This is reflected in the following

Definition and proposition 6.2.5.1 *Given functors $F, F' : \mathbf{C} \to \mathbf{B}, G, G' : \mathbf{B} \to \mathbf{A}$ and natural transformations $\tau : F \dot{\to} F', \eta : G \dot{\to} G'$ according to the following diagram,*

$$
\begin{tikzcd}
C \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow \tau"] & B \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow \eta"] & A
\end{tikzcd}
$$

the **vertical** *composition $\eta \bullet \tau : G \circ F \dot{\to} G' \circ F'$ is defined, at each component $C \in \mathbf{C}$, as $(\eta \bullet \tau)_C =_{def} G'(\tau_C) \circ \eta_{F(C)} = \eta_{F'(C)} \circ G(\tau_C)$.*

Proof: Consider the following diagram:

$$
\begin{tikzcd}
F(C) \arrow[d, "\tau_C"'] & G(F(C)) \arrow[r, "\eta_{F(C)}"] \arrow[d, "G(\tau_C)"'] & G'(F(C)) \arrow[d, "G'(\tau_C)"] \\
F'(C) & G(F'(C)) \arrow[r, "\eta_{F'(C)}"'] & G'(F'(C))
\end{tikzcd}
$$

Now, note that the above diagram commutes by naturality of η for the morphisms $\tau_C : F(C) \to F'(C)$ of B (which are indeed morphisms in $\mathbf{B}$ by naturality of τ). This shows that the two definitions for the composition above are equal. To see that this definition is also natural with respect to the morphisms $f : C \to B$ of $\mathbf{C}$ consider the following diagram:

$$\begin{tikzcd}
C \arrow[d, "f"'] & G(F(C)) \arrow[r, "G(\tau_C)"] \arrow[d, "G(F(f))"'] & G(F'(C)) \arrow[r, "\eta_{F'(C)}"] \arrow[d, "G(F'(f))"'] & G'(F'(C)) \arrow[d, "G'(F'(f))"] \\
B & G(F(C')) \arrow[r, "G(\tau_B)"'] & G(F'(C')) \arrow[r, "\eta_{F'(C')}"'] & G'(F'(C'))
\end{tikzcd}$$

To see that the left diagram commutes, consider the following diagram:

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

Note that it commutes by naturality of τ for the morphisms of $\mathbf{C}$. Therefore, the left diagram of the previous rectangle commutes because G is a functor and functors preserve commutative diagrams, i.e., composition. On the other hand, the right rectangle commutes because of naturality of η for the morphisms of $\mathbf{B}$, where $F'(f) : F'(C) \to F'(C')$ is indeed a morphism in $\mathbf{B}$ because F' is a functor.

This shows that the above rectangle commutes and hence that $\eta \bullet \tau$ is a natural transformation from $G \circ F$ to $G' \circ F'$.

$\square$

Remark 6.2.5.2 As an exercise in the use of natural transformations as well as in the use of the above definition for vertical composition, we show now that this composition is associative. To see this, consider the following diagram of functors and natural transformations:

$$\begin{tikzcd}
C \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow \tau"] & B \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow \eta"] & A \arrow[r, bend left, "H"] \arrow[r, bend right, "H'"'] \arrow[r, phantom, "\Downarrow \sigma"] & D
\end{tikzcd}$$

Now, see that for each $C \in \mathbf{C}$ we have:

$$\begin{aligned}
(\sigma \bullet (\eta \bullet \tau))_C &= H'((\eta \bullet \tau)_C \circ \sigma_{GF(C)} && \text{(by 6.2.5.1)} \\
&= H'(G'(\tau_C) \circ \eta_{F(C)}) \circ \sigma_{G(F(C))} && \text{(by 6.2.5.1)} \\
&= H'(G'(\tau_C)) \circ H'(\eta_{F(C)}) \circ \sigma_{G(F(C))} && \text{(H' is a functor)}
\end{aligned}$$

while

$$\begin{aligned}
((\sigma \bullet \eta) \bullet \tau)_C &= H'(G'(\tau_A)) \circ (\sigma \bullet \eta)_{F(C)} && \text{(by 6.2.5.1)} \\
&= H'(G'(\tau_C)) \circ H'(\eta_{F(C)}) \circ \sigma_{G(F(C))} && \text{(by 6.2.5.1)}
\end{aligned}$$

To see that $\sigma \bullet \eta \bullet \tau$ is natural, look at the following diagram:

$$
\begin{tikzcd}
C \arrow[d, "f"'] & HGF(C) \arrow[r, "\sigma_{GF(C)}"] \arrow[d, "HGF(f)"'] & H'GF(C) \arrow[r, "H'(\eta_{F(C)})"] \arrow[d, "H'GF(f)"'] & H'G'F(C) \arrow[r, "H'G'(\tau_C)"] \arrow[d, "H'G'F(f)"'] & H'G'F'(C) \arrow[d, "H'G'F'(f)"] \\
C' & HGF(C') \arrow[r, "\sigma_{GF(C')}"'] & H'GF(C') \arrow[r, "H'(\eta_{F(C')})"'] & H'G'F(C') \arrow[r, "H'G'(\tau_{C'})"'] & H'G'F'(C')
\end{tikzcd}
$$

where the first commutes by naturality of σ and the fact that $GF(f) : GF(C) \to GF(C')$ is a morphism in $\mathbf{A}$, the second because H', being a functor, preserves the naturality of η, and the third because $H' \circ G'$, being a functor, preserves the naturality of τ.

Remark 6.2.5.3 The previous calculations may have (hopefully) suggested us that the composition of a natural transformation with a functor (and vice-versa) turns out to be again a natural transformation. This may be precisely understood by first taking into consideration the following diagram of functors and natural transformations

$$
\begin{tikzcd}
C \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow \tau"] & B \arrow[r, bend left, "G"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow \iota_G"] & A \arrow[r, bend left, "H"] \arrow[r, bend right, "H'"'] \arrow[r, phantom, "\Downarrow \sigma"] & D
\end{tikzcd}
$$

where $\iota_G : G \dot{\to} G$ is the identity natural transformation. The precise formulation for the composition of a functor, say G, with a natural transformation, say $\sigma : H \to H'$, is given by the composition $(\sigma \bullet \iota_G) : HG \dot{\to} H'G$ where each component $B \in \mathbf{B}$ is given by:

$$
\begin{array}{rcll}
(\sigma \bullet \iota_G)_B & = & H'(\iota_G(B)) \circ \sigma_{G(B)} & \text{(by 6.2.5.1)} \\
 & = & H'(id_{G(B)}) \circ \sigma_{G(B)} & \text{(by definition of } \iota_G) \\
 & = & id_{H'(G(B))} \circ \sigma_{G(B)} & (H' \text{ is a functor}) \\
 & = & \sigma_{G(B)} & \text{(identity)}
\end{array}
$$

which is indeed a natural transformation by 6.2.5.1, or by inspection of the following diagram:

$$
\begin{tikzcd}
B \arrow[d, "f"'] & G(B) \arrow[d, "G(f)"'] & HG(B) \arrow[r, "\sigma_{G(B)}"] \arrow[d, "HG(f)"'] & H'G(B) \arrow[d, "H'G(B)"] \\
B' & G(B') & HG(B') \arrow[r, "\sigma_{G(B')}"'] & H'G(B)
\end{tikzcd}
$$

On the other hand, the composition of a natural transformation, say $\tau : F \to F'$, with a functor, say G, is given by the composition $\iota_G \bullet \tau : GF \dot{\to} GF'$, where each component $C \in \mathbf{C}$ is given by:

$$
\begin{array}{rcll}
(\iota_G \bullet \tau)_C & = & G(\tau_C) \circ \iota_G F(C) & \text{(by 6.2.5.1)} \\
 & = & G(\tau_C) \circ id_{G(F(C))} & \text{(definition of } \iota_G) \\
 & = & G(\tau_C) & \text{(identity)}
\end{array}
$$

which is also a natural transformation by definition 6.2.5.1, or by inspection of the following diagram:

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

It is sometimes useful to let the symbol G for a functor also denote the identity transformation $G \dot{\to} G$, so that the above described situation can now be given by the following diagram:

$$\begin{tikzcd} C \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\tau"] & B \arrow[r, "G"] & A \arrow[r, bend left, "H"] \arrow[r, bend right, "H'"'] \arrow[r, phantom, "\Downarrow\sigma"] & D \end{tikzcd}$$

where we now have $(G \bullet \tau)_C : GF(C) \to GF'(C) =_{def} G(\tau_C)$ and $(\sigma \bullet G)_B : HG(B) \to H'G(B) =_{def} \sigma_{G(B)}$.

6.2.6 Yoneda Lemma

Consider, for example the category **Graph** of graphs and graph morphisms of 2.3.39. Let the functor $F : \mathbf{Graph} \to \mathbf{Set}$ be such that for each graph $G = (E_G, V_G, s_G, t_G)$ we have $F(G) =_{def} V_G$ and each graph morphism $f = (f_E, f_V) : G \to H$, $F(f) =_{def} f_V$. Now we pick a graph with one vertex $*$ and denote it by $\{*\}$. Let $Hom_{\mathbf{Graph}}(\{*\}, -)$ be the corresponding (covariant) hom-functor. Now, a graph homomorphism from the graph $\{*\}$ to an arbitrary graph G is evidently determined by the image of $\{*\}$ and that can by any vertex from G. In other words, vertices from G are "essentially the same thing" as graph homomorphisms from $\{*\}$ to G, that is to say, as the elements from $Hom_{\mathbf{Graph}}(\{*\}, G)$. Now we can define a natural transformation $\alpha : Hom_{\mathbf{Graph}}(\{*\}, -) \dot{\to} F$ by defining each component $\alpha_G : Hom_{\mathbf{Graph}}(\{*\}, G) \to F(G)$ for each $f \in Hom_{\mathbf{Graph}}(\{*\}, G)$ (note that each graph morphism in this hom set has just the "vertex" component) by

$$\alpha_G(f) =_{def} F(f)(*) = f_V(*)$$

To see that α is natural, we have to show that for every graph morphism $g : G \to G'$ the following diagram commutes:

$$\begin{tikzcd} G \arrow[d, "g"'] & Hom_{\mathbf{Graph}}(\{*\}, G) \arrow[r, "\alpha_G"] \arrow[d, "Hom_{\mathbf{Graph}}(\{*\}, g)"'] & F(G) \arrow[d, "F(g)"] \\ G' & Hom_{\mathbf{Graph}}(\{*\}, G') \arrow[r, "\alpha_{G'}"] & F(G') \end{tikzcd}$$

Now, picking any $f : \{*\} \to G$, and chasing it counterclockwise we have:

$$
\begin{aligned}
\alpha_{G'} \circ (Hom_{\mathbf{Graph}}(\{*\}, g)(f) &= \alpha_{G'}(g \circ f) && \text{(by definition of } Hom_{\mathbf{Graph}}(\{*\}, g)) \\
&= (g \circ f)_V(*) && \text{(by the above definition)} \\
&= (g_V \circ f_V)(*) && \text{(by definition)} \\
&= (g_V(f_V(*)) && \text{(composition)} \\
&= g_V(\alpha_G(f)) && \text{(by the above definition)} \\
&= F(g)(\alpha_G(f)) && \text{(by definition of } F) \\
&= F(g) \circ \alpha_G(f) && \text{(composition)}
\end{aligned}
$$

This shows that α is natural. According to the above discussion, we have shown that the vertex "*" uniquely determines the natural transformation α. Conversely, each natural transformation $\alpha : Hom_{\mathbf{Graph}}(\{*\}, -) \dot{\to} F$ determines a element of the set $F(G) =_{def} G_V$, i.e., a vertex, by the map $\alpha_{\{*\}}(id_{\{*\}}) : Hom_{\mathbf{Graph}}(\{*\}, \{*\}) \to F(\{*\})$ (which is, in this case, just the vertex "*").

This discussion may (hopefully) illuminate the following remarkable

Theorem 6.2.6.1 *(***Yoneda Lemma***) Let $F : \mathbf{C} \to \mathbf{Set}$ be a set-valued functor and C an object of $\mathbf{C}$ (for $\mathbf{C}$ a category with small hom-sets). Then there is a natural bijection*

$$
y_{\langle F,C\rangle} : Nat(Hom_{\mathbf{C}}(C, -), F) \cong F(C)
$$

which sends each natural transformation $\alpha : Hom_{\mathbf{C}}(C, -) \dot{\to} F$ to the element $\alpha_C(id_C) \in F(C)$, the image of the identity $id_C : C \to C$.

Proof: To define a natural transformation $\alpha \in Nat(Hom_{\mathbf{C}}(C, -), F)$, we have to define a family of functions $\alpha_{C'} : Hom_{\mathbf{C}}(C, C') \to F(C')$ for each $C' \in \mathbf{C}$. This means that we are looking, for each $f : C \to C'$, the right side of

$$
\alpha_{C'}(f) =?
$$

Now observe, by definition of $Hom_{\mathbf{C}}(C, f)$, that $Hom_{\mathbf{C}}(C, f)(id_C) = f$. This gives the final clue to define the transformation α. Now, look at the following commutative diagram (by naturality of α):

$$
\begin{array}{ccc}
C & & Hom_{\mathbf{C}}(C, C) \xrightarrow{\alpha_C} F(C) \\
{\scriptstyle f}\downarrow & & {\scriptstyle Hom_{\mathbf{C}}(C,f)}\downarrow \qquad\qquad \downarrow{\scriptstyle F(f)} \\
C' & & Hom_{\mathbf{C}}(C, C') \xrightarrow[\alpha_{C'}]{} F(C')
\end{array}
$$

We will chase id_C in the above diagram. Chasing counterclockwise we have:

$$
\begin{aligned}
\alpha_{c'} \circ Hom_{\mathbf{C}}(C, f)(id_C) &= \alpha_{C'}(f \circ id_C) && \text{(by definition of } Hom_{\mathbf{C}}(C, f)) \\
&= \alpha_{C'}(f) && \text{(identity)}
\end{aligned}
$$

while clockwise gives us:

$$F(f)(\alpha_C(id_C))$$

The above calculations are expressed in the following diagram:

$$\begin{tikzcd}
id_C \arrow[r, mapsto] \arrow[d, mapsto] & \alpha_C(id_C) \arrow[d, mapsto] \\
Hom_{\mathbf{C}}(C,f)(id_C) = f \arrow[r, mapsto] & \alpha_{C'}(f) = F(f)(\alpha_C(id_C))
\end{tikzcd}$$

Therefore, the above chasing says that $\alpha_{C'}(f) = F(f)(\alpha_C(id_C))$, that is to say, given $f : C \to C' \in Hom_{\mathbf{C}}(C, C')$, then the component of the natural transformation α at C' ($\alpha_{C'}$) is the function that takes f to the evaluation of the induced function $F(f)$ at $\alpha_C(id_C)$, which is of course an element in $F(C')$.

To say that $\alpha_{C'} = F(f)(\alpha_C(id_C))$ is indeed the component of a natural transformation α in $Nat(Hom_{\mathbf{C}}(C, -), F)$ at C', we must show that for any $g : C' \to B$ in $\mathbf{C}$ the following diagram commutes:

$$\begin{tikzcd}
C' \arrow[d, "g"'] & Hom_{\mathbf{C}}(C,C') \arrow[r, "\alpha_{C'}"] \arrow[d, "Hom_{\mathbf{C}}(C',g)"'] & F(C') \arrow[d, "F(g)"] \\
B & Hom_{\mathbf{C}}(C,B) \arrow[r, "\alpha_B"'] & F(B)
\end{tikzcd}$$

Now, we pick any $f \in Hom_{\mathbf{C}}(C, C')$ and we have (going counterclockwise):

$$\begin{array}{rcllr}
\alpha_B \circ Hom_{\mathbf{C}}(C,g)(f) & = & \alpha_B(Hom_{\mathbf{C}}(C,g)(f)) & & \text{(composition)} \\
& = & \alpha_B(g \circ f) & & \text{(by definition of } Hom_{\mathbf{C}}(C,g)) \\
& = & F(g \circ f)(\alpha_C(id_C)) & & \text{(by definition of } \alpha_B(g \circ f)) \\
& = & F(g) \circ F(f)(\alpha_C(id_C)) & & (F \text{ is a functor}) \\
& = & F(g) \circ \alpha_{C'}(f) & & \text{(by definition of } \alpha_{C'}(f))
\end{array}$$

This shows that the family $\alpha_{C'}(f) =_{def} F(f)(\alpha_C(id_C)) \in F(C')$ for every $C' \in \mathbf{C}$ is the natural transformation α and more important, that each natural transformation $\alpha : Hom_{\mathbf{C}}(C, -) \dot{\to} F$ is fully determined by the image of α at the identity id_C. Thus, for each $x \in F(C), C', f : C \to C' \in \mathbf{C}, (y^{-1}_{\langle F,C \rangle}(x))_{C'} = \alpha_{C'}(f) =_{def} F(f)(x)$ and $y_{\langle F,C \rangle}(\alpha) =_{def} \alpha_C(id_C)$ is the wanted bijection. To see this, let β be the natural transformation determined by an element $x \in F(C)$ under the bijection $y_{\langle F,C \rangle}$. Now we have:

$$\begin{array}{rcllr}
y_{\langle F,C \rangle} \circ y^{-1}_{\langle F,C \rangle}(x) & = & y_{\langle F,C \rangle}(\beta) & & \text{(by definition)} \\
& = & \beta_C(id_C) & & \text{(by definition of } y_{\langle F,C \rangle}) \\
& = & F(id_C)(x) & & \text{(by definition of } \beta) \\
& = & id_{F(C)}(x) & & (F \text{ is a functor}) \\
& = & x & & \text{(identity)}
\end{array}$$

This shows that $y_{\langle F,C\rangle} \circ y^{-1}_{\langle F,C\rangle} = id_{F(C)}$. On the other hand, let $\alpha : Hom_{\mathbf{C}}(C,-) \dot{\rightarrow} F$. Then, by definition of $y_{\langle F,C\rangle}$,

$$y_{\langle F,C\rangle}(\alpha) = \alpha_C(id_C)$$

Now assume that $\beta = y^{-1}_{\langle F,C\rangle}(\alpha_C(id_C))$. Then by definition we have

$$\beta_C(id_C) = F(id_C)(\alpha_C(id_C)) = id_{F(C)}(\alpha_C(id_C)) = \alpha_C(id_C)$$

Now, let C' be any $\mathbf{C}$-object and $f \in Hom_{\mathbf{C}}(C,C')$. Then consider the following calculation:

$$\begin{array}{rcll}
\beta_{C'}(f) & = & \beta_{C'}(f \circ id_C) & \text{(identity)} \\
 & = & \beta_{C'} \circ Hom_{\mathbf{C}}(C,f)(id_C) & \text{(by definition of } Hom_{\mathbf{C}}(C,f)) \\
 & = & F(f) \circ \beta_C(id_C) & \text{(by naturality of } \beta) \\
 & = & F(f) \circ \alpha_C(id_C) & \text{(calculation above)} \\
 & = & \alpha_{C'} \circ Hom_{\mathbf{C}}(C,f)(id_C) & \text{(naturality of } \alpha) \\
 & = & \alpha_{C'}(f \circ id_C) & \text{(by definition of } Hom_{\mathbf{C}}(C,f)) \\
 & = & \alpha_{C'}(f) & \text{(identity)}
\end{array}$$

Hence, since f and C' were arbitrary, $\alpha = \beta$ and thus $y^{-1}_{\langle F,C\rangle} \circ y_{\langle F,C\rangle} = id_{Nat(Hom_{\mathbf{C}}(C,-),F)}$, as required.

The proof of naturality of y is postponed to 6.2.6.10.

□

Take now, in the above theorem, instead of the functor $F : \mathbf{C} \rightarrow \mathbf{Set}$, a covariant hom functor from $\mathbf{C}$ to $\mathbf{Set}$, say, $Hom_{\mathbf{C}}(D,-) : \mathbf{C} \rightarrow Set$. Then, by the Yoneda Lemma above, we have a bijective correspondence

$$Nat(Hom_{\mathbf{C}}(C,-), Hom_{\mathbf{C}}(D,-)) \cong Hom_{\mathbf{C}}(D,-)(C) = Hom_{\mathbf{C}}(D,C)$$

or in other words, the mapping

$$Hom_{\mathbf{C}}(D,C) \mapsto Nat(Hom_{\mathbf{C}}(C,-), Hom_{\mathbf{C}}(D,-))$$

is both injective and surjective. That means that each morphism in $Hom_{\mathbf{C}}(D,C)$ (thus an element of $Hom_{\mathbf{C}}(D,-)(C)$) uniquely defines a natural transformation $\alpha : Hom_{\mathbf{C}}(C,-) \dot{\rightarrow} Hom_{\mathbf{C}}(D,-)$ and vice versa. If we are able to describe this mapping by a functor, say, Y, then we have trivially that Y becomes both faithful and full. However, this is nothing else than the work of the next

Definition and proposition 6.2.6.2 (**Yoneda Embedding**) *Let* $\mathbf{C}$ *be a category. There is a (contravariant) functor* $Y : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathbf{C}}$, *the* **Yoneda functor**, *defined as follows:*

- *For an object* C *of* $\mathbf{C}$, $Y(C) =_{def} Hom_{\mathbf{C}}(C, -)$.
- *If* $f : D \to C$ *in* $\mathbf{C}$ *and* A *is an object of* $\mathbf{C}$, *then the component* $Y(f)_A : Hom_{\mathbf{C}}(C, A) \to Hom_{\mathbf{C}}(D, A)$ *of the natural transformation* $Y(f) : Hom_{\mathbf{C}}(C, -) \dot{\to} Hom_{\mathbf{C}}(D, -)$ *is defined for every* $h : C \to A$ *by*

$$Y(f)_A(h) = h \circ f$$

Proof: To see that Y is well-defined, i.e., to show that $Y(f)$ is a natural transformation, requires to show, given $f : D \to C$ in $\mathbf{C}$, that the next diagram commutes for every morphism $k : A \to B$.

$$\begin{tikzcd}
A \arrow[d, "k"'] & Hom_{\mathbf{C}}(C,A) \arrow[r, "Y(f)_A"] \arrow[d, "Hom_{\mathbf{C}}(C,k)"'] & Hom_{\mathbf{C}}(D,A) \arrow[d, "Hom_{\mathbf{C}}(D,k)"] \\
B & Hom_{\mathbf{C}}(C,B) \arrow[r, "Y(f)_B"'] & Hom_{\mathbf{C}}(D,B)
\end{tikzcd}$$

To see that it commutes, start with $h : C \to A$, an arbitrary element of the northwest corner. The lower route takes this to $k \circ h$, then to $(k \circ h) \circ f$. The upper route takes it to $k \circ (h \circ f)$, so the fact that the diagram commutes is simply a statement of the associative law.

To show that Y is a functor, we have to show that it preserves identities and composition. To see that it preserves identities, note that for each component $B \in \mathbf{C}$, we have that $Y(id_A)_B$ is, by definition, a function $Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{C}}(A, B)$ such that for any $f : A \to B$, $Y(id_A)_B(f) = f \circ id_A$, and this is certainly the identity for any $f \in Hom_{\mathbf{C}}(A, B)$, or in other words, $Y(id_A)_B = - \circ id_A = id_{Hom_{\mathbf{C}}(A,B)} = id_{Hom_{\mathbf{C}}(A,-)(B)} = id_{Y(A)(B)} = \iota_{Y(A)}(B)$, i.e., at each component $B \in \mathbf{C}$, $\iota_{Y(A)}(B)$ is an identity for the set $Hom_{\mathbf{C}}(A, B)$. To see that Y preserves composition, consider $f : A \to B, g : B \to C \in \mathbf{C}$. Then, at each component B, $Y(g \circ f)_B$ is a function from $Hom_{\mathbf{C}}(C, B)$ to $Hom_{\mathbf{C}}(A, B)$ (since $g \circ f : A \to C$). Now, for any $h : C \to B$ we have:

$$\begin{array}{rcll}
Y(g \circ f)_B(h) & = & h \circ (g \circ f) & \text{(by definition)} \\
& = & (h \circ g) \circ f & \text{(associativity)} \\
& = & Y(f)_B(h \circ g) & (h \circ g : B \to B, Y(f) : Hom_{\mathbf{C}}(B, -) \dot{\to} Hom_{\mathbf{C}}(A, -)) \\
& = & Y(f)_B \circ Y(g)_B(h) & (h : C \to B, Y(g) : Hom_{\mathbf{C}}(C, -) \dot{\to} Hom_{\mathbf{C}}(B, -))
\end{array}$$

$\square$

Remark 6.2.6.3 Note that $Y(C)$ is a covariant hom-functor and that $Y(f)_A$ is a component of a contravariant hom-functor.

The main result concerning Y is the following

Theorem 6.2.6.4 *$Y : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathbf{C}}$ is a full and faithful functor.*

Proof: Directly from the discussion above. □

Corollary 6.2.6.5 *Every natural transformation $\alpha : Hom_{\mathbf{C}}(C, -) \dot{\to} Hom_{\mathbf{D}}(D, -)$ is given by composition with a unique morphism $f : D \to C$.*

Proof: Directly by definition of $Y : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathbf{C}}$ and the above theorem. □

Definition and proposition 6.2.6.6 *By replacing $\mathbf{C}$ by $\mathbf{C}^{\mathrm{op}}$ in 6.2.6.2, we derive a second Yoneda functor $J : \mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$. For an object C of $\mathbf{C}$, $J(C) = Hom_{\mathbf{C}}(-, C)$, the contravariant hom-functor. If $f : C \to D$ in $\mathbf{C}$ and A is an object of $\mathbf{C}$, then the component*

$$J(f)_A : Hom_{\mathbf{C}}(A, C) \to Hom_{\mathbf{C}}(A, D)$$

of the natural transformation $J(f) : Hom_{\mathbf{C}}(-, C) \dot{\to} Hom_{\mathbf{C}}(-, D)$ is

$$Hom_{\mathbf{C}}(A, f) : Hom_{\mathbf{C}}(A, C) \to Hom_{\mathbf{C}}(A, D)$$

Remark 6.2.6.7 We leave the reader with the task of showing the the above definition gives rise to a functor which is also full and faithful!

Before delving into the proof of the natural bijection stated in the Yoneda lemma, we need to introduce two non-trivial functors, which are introduced and discussed in detail in the next two examples.

Example 6.2.6.8 (The evaluation functor) We now define a set-valued functor $Ev : \mathbf{C} \times \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}$ such that for each object $\langle A, F \rangle$ in $\mathbf{C} \times \mathbf{Set}^{\mathbf{C}}$, where A is an object in $\mathbf{C}$ and $F : \mathbf{C} \to \mathbf{Set}$ a functor (i.e., an object in $\mathbf{Set}^{\mathbf{C}}$), $Ev(\langle A, F \rangle) = F(A)$, and for a morphism $\langle f, \alpha \rangle$ in $\mathbf{C} \times \mathbf{Set}^{\mathbf{C}}$, where $f : A \to A'$ in $\mathbf{C}$ and $\alpha : F \dot{\to} F'$ in $\mathbf{Set}^{\mathbf{C}}$, $Ev(\langle f, \alpha \rangle) : F(A) \to F'(A)$ is taken to be one of the two (equal) paths from $F(A)$ to $F'(A')$ in the following commutative diagram (since α is natural):

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

This means that $Ev(\langle f, \alpha\rangle) = \alpha_{A'} \circ F(f) = F'(f) \circ \alpha_A$.

It remains to show that $Ev : \mathbf{C} \times \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}$ preserves identities and composition.

1. Preservation of identities: given $id_A : A \to A$ in $\mathbf{C}$ and the identity natural transformation $\iota : F \dot{\to} F$, where $F : \mathbf{C} \to \mathbf{Set}$, we have:

$$\begin{array}{rcll} Ev(id_A, \iota_F) & = & \iota_F(A) \circ F(id_A) & \text{(by definition of } Ev) \\ & = & id_{F(A)} \circ id_{F(A)} & \text{(by definition of } \iota_F;\ F \text{ is a functor)} \\ & = & id_{F(A)} & \text{(identity)} \\ & = & id_{(Ev(A,F))} & \text{(by definition of } Ev) \end{array}$$

2. Preservation of composition: now, given $\mathbf{C}$-morphisms $f : A \to B, g : B \to C$, and natural transformations (i.e., morphisms in $\mathbf{Set}^{\mathbf{C}}$) $\alpha : F \dot{\to} G, \beta : G \dot{\to} H$ we have, at each component $C \in \mathbf{C}$

$$\begin{array}{rcll} Ev(\langle g \circ f, \beta \circ \alpha\rangle) & = & (\beta \circ \alpha)_C \circ F(g \circ f) & \text{(by definiton of } Ev) \\ & = & \beta_C \circ \alpha_C \circ F(g) \circ F(f) & \text{(by definition of } (\beta \circ \alpha)_C;\ F \text{ is a functor)} \\ & = & \beta_C \circ G(g) \circ \alpha_B \circ F(f) & \text{(by naturality of } \alpha \text{ in (1) below)} \\ & = & Ev(\langle g, \beta\rangle) \circ Ev(\langle f, \alpha\rangle) & \text{(by definition of } Ev) \end{array}$$

$$\begin{tikzcd} B \arrow[d, "g"'] & F(B) \arrow[r, "\alpha_B"] \arrow[d, "F(g)"'] \arrow[dr, phantom, "(1)"] & G(B) \arrow[d, "G(g)"] \\ C & F(C) \arrow[r, "\alpha_C"'] & G(C) \end{tikzcd}$$

Example 6.2.6.9 Remember that in the proof of the Yoneda lemma, we have, given a category $\mathbf{C}$, an object C in $\mathbf{C}$, and two set-valued functors $Hom_{\mathbf{C}}(C, -), F : \mathbf{C} \to \mathbf{Set}$, constructed a bijection $Nat(Hom_{\mathbf{C}}(C, -), F) \cong F(C)$ between the set $F(C)$ and the collection of natural transformations in $Nat(Hom_{\mathbf{C}}(C, -), F)$, which shows that this collection is indeed a set. This fact allows us to defined a set-valued functor $Nt : \mathbf{C} \times \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}$ in the following way:

- given C in $\mathbf{C}$ and $F : \mathbf{C} \to \mathbf{Set}$, we put $Nt(\langle C, F\rangle) = Nat(Hom_{\mathbf{C}}(C, -), F)$;
- given $f : A \to A'$ in $\mathbf{C}$ and $\alpha : F \dot{\to} G$ then $Nt(\langle f, \alpha\rangle)$ is the function $Nat(\langle f, \alpha\rangle) : Nat(Hom_{\mathbf{C}}(A, -), F) \to Nat(Hom_{\mathbf{C}}(A', -), F')$, such that for every β in $Nat(Hom_{\mathbf{C}}(A, -), F)$, $Nat(\langle f, \alpha\rangle)(\beta) = \alpha \circ \beta \circ Y(f)$, where $Y(f) : Hom_{\mathbf{C}}(A', -) \dot{\to} Hom_{\mathbf{C}}(A, -)$ is the application of the Yoneda functor (see 6.2.6.2) to $f : A \to A'$), i.e., $Y(f)$ is a natural transformation between hom-functors. Note that this composition is a well-defined natural transformation $Hom_{\mathbf{C}}(A', -) \dot{\to} F'$.

It remains to show preservation of identities and composition.

1. Preservation of identities: given $id_A : A \to A$ in $\mathbf{C}$, $\iota_F : F \to F'$, and $\alpha : Hom_{\mathbf{C}}(A, -) \dot{\to} F$, then $Nt(\langle id_A, \iota_F\rangle) : Nat(Hom_{\mathbf{C}}(A, -), F) \to Nat(Hom_{\mathbf{C}}(A, -), F)$ and we have:

$$
\begin{aligned}
Nt(\langle id_A, \iota_F\rangle)(\alpha) &= \iota_F \circ \alpha \circ Y(id_A) && \text{(by definition of } Nt) \\
&= \iota_F \circ \alpha \circ id_{Y(A)} && (Y \text{ is a functor}) \\
&= \iota_F \circ \alpha \circ id_{Hom_{\mathbf{C}}(A,-)} && \text{(by definition of } Y(A)) \\
&= \iota_F \circ \alpha \circ \iota_{Hom_{\mathbf{C}}(A,-)} && \text{(our convention)}
\end{aligned}
$$

that is to say, $\iota_F \circ - \circ \iota_{Hom_{\mathbf{C}}(A,-)}$ is the identity for every α in $Nat(Hom_{\mathbf{C}}(A,-), F)$, or more precisely,

$$
Nt(\langle id_A, \iota_F\rangle) = \iota_F \circ - \circ \iota_{Hom_{\mathbf{C}}(A,-)} = id_{Nat(Hom_{\mathbf{C}}(A,-),F)} = id_{Nt(\langle A,F\rangle)}.
$$

2. Preservation of composition: given $f : A \to B, g : B \to C$ in $\mathbf{C}$, $\alpha : F \dot{\to} G, \beta : G \dot{\to} H$ in $\mathbf{Set}^{\mathbf{C}}$, then $Nat(\langle g \circ f, \beta \circ \alpha\rangle) : Nat(Hom_{\mathbf{C}}(A,-), F) \to Nat(Hom_{\mathbf{C}}(C,-), H)$, such that for each γ in $Nat(Hom_{\mathbf{C}}(A,-), F)$ we have:

$$
\begin{aligned}
Nt(\langle g \circ f, \beta \circ \alpha\rangle)(\gamma) &= (\beta \circ \alpha) \circ \gamma \circ (Y(g \circ f)) && \text{(by definition of } Nt) \\
&= (\beta \circ \alpha) \circ \gamma \circ (Y(f) \circ Y(g)) && (Y \text{ is a contravariant functor}) \\
&= \beta \circ (\alpha \circ \gamma \circ Y(f)) \circ Y(g) && \text{(associativity)} \\
&= Nt(\langle g, \beta) \circ (\alpha \circ \gamma \circ Y(f)) && \text{(by definition of } Nt) \\
&= Nt(\langle g, \beta\rangle) \circ Nt(\langle f, \alpha\rangle)(\gamma) && \text{(by definition of } Nt)
\end{aligned}
$$

Proposition 6.2.6.10 *The bijection $y_{\langle C,F\rangle} : Nat(Hom_{\mathbf{C}}(C,-), F) \to F(C)$ is the component at $\langle C, F\rangle$ of a natural isomomorphism $y : Nt \dot{\to} Ev$, between the functors $Nt, Ev : \mathbf{C} \times \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}$ introduced above.*

Proof: The diagram which must be shown commutative is the following one, where $f : C \to C'$ is a morphism in $\mathbf{C}$ and $\alpha : F \to F'$ is a morphism (i.e., a natural transformation) in $\mathbf{Set}^{\mathbf{C}}$:

$$
\begin{array}{ccccc}
\langle C, F\rangle & \qquad & Nat(Hom_{\mathbf{C}}(C,-), F) & \xrightarrow{y_{\langle C,F\rangle}} & F(C) \\
\downarrow{\scriptstyle \langle f,\alpha\rangle} & & \downarrow{\scriptstyle Nt(\langle f,\alpha\rangle)} & & \downarrow{\scriptstyle Ev(\langle \alpha,f\rangle)} \\
\langle C', F'\rangle & & Nat(Hom_{\mathbf{C}}(C',-), F') & \xrightarrow[y_{\langle C',F'\rangle}]{} & F'(C')
\end{array}
$$

Therefore, let β be a natural transformation in $Nat(Hom_{\mathbf{C}}(C,-), F)$. Then we have:

$$
\begin{array}{rcll}
(y_{\langle C',F'\rangle} \circ Nt(\langle f,\alpha\rangle))(\beta) & = & y_{\langle C',F'\rangle}(Nt(\langle f,\alpha\rangle)(\beta)) & \text{(composition)} \\
& = & y_{\langle C',F'\rangle}(\alpha \circ \beta \circ Y(f)) & \text{(by definition of } Nt) \\
& = & (\alpha \circ \beta \circ Y(f))_{C'}(id_{C'}) & \text{(by definition of } y_{\langle C',F'\rangle}) \\
& = & (\alpha_{C'} \circ \beta_{C'} \circ Y(f)_{C'})(id_{C'}) & \text{(by definition)} \\
& = & (\alpha_{C'} \circ \beta_{C'})(Y(f)_{C'}(id_{C'})) & \text{(associativity and composition)} \\
& = & (\alpha_{C'} \circ \beta_{C'})(id_{C'} \circ f) & \text{(by definition of } Y) \\
& = & (\alpha_{C'} \circ \beta_{C'})(f) & \text{(identity)} \\
& = & (\alpha_{C'} \circ \beta_{C'})(f \circ id_C) & \text{(identity)} \\
& = & (\alpha_{C'} \circ \beta_{C'})(J(f)_C(id_C)) & \text{(by definition of } J\text{, see 6.2.6.6)} \\
& = & (\alpha_{C'} \circ \beta_{C'} \circ J(f)_C)(id_C) & \text{(composition and associativity)} \\
& = & (\alpha_{C'} \circ F(f) \circ \beta_C)(id_C) & \text{(naturality of } \beta \text{ in (1) below)} \\
& = & (F'(f) \circ \alpha_C \circ \beta_C)(id_C) & \text{(naturality of } \alpha \text{ in (2) below)} \\
& = & (F'(f) \circ \alpha_C)(\beta_C(id_C)) & \text{(composition)} \\
& = & (F'(f) \circ \alpha_C)(y_{\langle C,F\rangle}(\beta)) & \text{(by definition of } y_{\langle C,F\rangle}) \\
& = & Ev(\langle f,\alpha\rangle)(y_{\langle C,F\rangle}(\beta)) & \text{(by definition of } Ev) \\
& = & Ev(\langle f,\alpha\rangle) \circ y_{\langle C,F\rangle}(\beta) & \text{(by composition)}
\end{array}
$$

$$
\begin{tikzcd}
C \arrow[d, "f"'] & Hom_{\mathbf{C}}(C,C) \arrow[r, "\beta_C"] \arrow[d, "J(f)_C"'] \arrow[dr, phantom, "(1)"] & F(C) \arrow[d, "F(f)"] \\
C' & Hom_{\mathbf{C}}(C,C') \arrow[r, "\beta_{C'}"'] & F(C')
\end{tikzcd}
$$

$$
\begin{tikzcd}
C \arrow[d, "f"'] & F(C) \arrow[r, "\alpha_C"] \arrow[d, "F(f)"'] \arrow[dr, phantom, "(2)"] & F'(C) \arrow[d, "F'(f)"] \\
C' & F(C') \arrow[r, "\alpha_{C'}"'] & F'(C')
\end{tikzcd}
$$

Naturality in the other direction, i.e., naturality of y^{-1}, follows directly from 6.2.3.3.

□

6.3 Bibliographic notes

The choice of examples and propositions presented in this chapter are based mainly in [BW90], [Pie90] and [Gol96]. Especially, the presentation of comma-categories is an adaptation from [AHS90], whose notation mention explicitly the functors which are present in the constructions, a detail that is usually transparent in some presentation of this subject by other authors. The more advanced notions of horizontal composition of natural transformations and the Yoneda Lemma are adaptations from [MaC71] and [BW90], respectively.

Chapter 7

Free constructions and adjoint functors

Adjoints are about one of the most important ideas in category theory. However, we first examine the notion of a "free construction" in detail, and then we show also that it can, under one specific condition, to be extended to a functor which is called "free functor". As examples of free constructions, we present the "special" cases of limits we have encountered so far. The corresponding "dual" construction, i.e., cofree constructions, is also presented along the same lines. The section on adjoints is meant to provide a deeper understanding between the relation of free and cofree constructions, as well as some characterization "theorems" that express the existence of such adjoint situations. Moreover, these constructions are more intricate that anything we have encountered so far. The reader must not expect to understand its relevance upon first reading the definitions or the few examples presented in this chapter: only by systematic use of adjunctions will she or he become competent on the subject.

7.1 Free constructions

Let **Mon** denote the category of monoids and let $U : \mathbf{Mon} \to \mathbf{Set}$ denote the underlying set functor. In a monoid $(U(M), \cdot, e)$ (see definition 2.3.6), $U(M)$ is a set called the **underlying set** of M. Now, given a set S, we can construct the free monoid $(List(S), \diamond, [])$ (see example 6.1.1.7). This free monoid is characterized by the property given in the following proposition: the property is called the **universal mapping property** of the free monoid. In the proposition, $i : S \to List(S)$ takes $s \in S$ to the list $[s]$ of length 1. Also, note the difference between the free monoid $F(S) = (List(S), \diamond, [])$ and its underlying set, the Kleene closure $S^* = List(S) = U(F(S))$.

Proposition 7.1.1 *Let $f : S \to M$ be any function from a set S to the underlying set $U(M)$ of a monoid $(M, \cdot, e)$ and let i be the injection taking an element $s \in S$ to the list $[s]$ of length 1. Then there is exactly one monoid homomorphism*

$$f^\sharp : (List(S), \diamond, []) \to (M, \cdot, e)$$

such that the following diagram commutes:

$$\begin{tikzcd} S \arrow[r, "i"] \arrow[dr, "f"'] & List(S) \arrow[d, "f^\sharp"] \\ & M \end{tikzcd}$$

Proof: We define $f^\sharp$ to be the monoid homomorphism taking each list $[s_1, s_2, \ldots, s_n]$ to the product $f(s_1) \cdot f(s_2) \cdot \ldots \cdot f(s_n)$ in $(M, \cdot, e)$ and taking the empty list $[]$ to e. This definition clearly satisfies the condition for being a monoid homomorphism (conditions 1 and 2) and makes the above diagram commute (condition 3):

1. $f^\sharp([]) = e$
2. $f^\sharp([s_1, s_2, \ldots, s_n] \diamond [t_1, t_2, \ldots, t_m]) = f^\sharp([s_1, \ldots, s_n]) \cdot f^\sharp([t_1, t_2, \ldots, t_m])$
3. $f^\sharp \circ i = f$

Now assume that some f' also satisfies these conditions. For any $L \in List(S)$, we show, by induction on the length of L, that $f'(L) = f^\sharp(L)$, and thus that $f' = f^\sharp$.

If $L = []$, the first condition forces $f'(L) = e = f^\sharp(L)$. If $L = [s_1, s_2, \ldots, s_n]$, $(n \geq 1)$, then $L = [s_1] \diamond [s_2, \ldots, s_n]$. By the third condition, $f'([s_1]) = f'(i(s_1)) = f(s_1) = f^\sharp(i(s_1)) = f^\sharp([s_1])$. By induction hypothesis we have that $f'([s_2, \ldots, s_n]) = f^\sharp([s_2, \ldots, s_n])$. Now we have:

$$\begin{aligned}
&f'(L) = \\
&= f'([s_1] \diamond [s_2, \ldots, s_n]) && \text{(definition of } L) \\
&= f'([s_1]) \cdot f'([s_2, \ldots, s_n]) && \text{(condition 2)} \\
&= f'(i(s_1)) \cdot f'([s_2, \ldots, s_n]) && \text{(definition of } [s_1]) \\
&= f^\sharp(i(s_1)) \cdot f'([s_2, \ldots, s_n]) && \text{(assumption)} \\
&= f^\sharp([s_1]) \cdot f^\sharp([s_2, \ldots, s_n]) && \text{(inductive hypothesis and definition of } [s_1]) \\
&= f^\sharp([s_1] \diamond [s_2, \ldots, s_n]) && \text{(condition 2)} \\
&= f^\sharp(L) && \text{(definition of } L)
\end{aligned}$$

□

Remark 7.1.2 The homomorphism $f^\sharp$ is called **extension** of f because it agrees with f on the elements of S, i.e., $f^\sharp([s]) = f(s)$.

The above proposition motivate us to present the following

Definition 7.1.3 *Let $G : \mathbf{D} \to \mathbf{C}$ be a functor. We call an object D in $\mathbf{D}$ a* **free construction over** *A* **with respect to** *G, if there is a morphism $\eta_A : A \to G(D)$ in $\mathbf{A}$, called* **universal morphism***, such that for any morphism $f : A \to G(B)$ ($A \in \mathbf{C}, B \in \mathbf{D}$) there is a unique morphism $f^\sharp : D \to B$ in $\mathbf{D}$ such that $G(f^\sharp) \circ \eta_A = f$. In this case we say that the following diagram commutes:*

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\eta_A"'] & G(B) \\
G(D) \arrow[ur, dashed, "G(f^\sharp)"'] &
\end{tikzcd}
$$

Remark 7.1.4 The pair (D, η_A) is normally used to denote the fact that D is a free construction over A with respect to G. The pair is also called **universal** from A to G in the literature of category theory.

Proposition 7.1.5 *Free constructions are unique up to isomorphism.*

Proof: Consider the following diagram:

$$
\begin{tikzcd}
 & GD \\
A \arrow[ur, dashed, "\eta_A"] \arrow[r, "\eta'_A"] \arrow[d, "\eta_A"'] \arrow[dr, phantom, "(1)"] & GD' \arrow[u, "G(\eta_A^\sharp)"'] \arrow[u, phantom, "(2)", shift right=4ex] \\
GD \arrow[ur, dashed, "G((\eta'_A)^\sharp)"'] &
\end{tikzcd}
$$

Suppose that D and D' are free over A with respect to $G : \mathbf{D} \to \mathbf{C}$. Since D is free over A there is a unique $(\eta'_A)^\sharp : D \to D'$ such that (1) commutes. Also, since D' is free over A, there is a unique $\eta_A^\sharp : D' \to D$ such that (2) commutes. Now consider the following diagram:

$$
\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[d, "\eta_A"'] & GD \\
GD \arrow[ur, dashed, "G(k)"'] &
\end{tikzcd}
$$

Since D is free over A, then there must be a unique $k : D \to D$ such that the above diagram commutes. Clearly $k = id_D$ does the job since

$G(id_D) \circ \eta_A =$

$= id_{GD} \circ \eta_A$ (G is a functor)

$= \eta_A$ (identity)

Now putting $k = \eta_A^\sharp \circ (\eta'_A)^\sharp$ we have:

$$
\begin{aligned}
&G(\eta_A^\sharp \circ (\eta'_A)^\sharp) \circ \eta_A = \\
&= G(\eta_A^\sharp) \circ G((\eta'_A)^\sharp) \circ \eta_A && (G \text{ is a functor}) \\
&= G(\eta_A^\sharp) \circ \eta'_A && (D \text{ is free over } A) \\
&= \eta_A && (D' \text{ is free over } A)
\end{aligned}
$$

Thus both id_D and $\eta_A^\sharp \circ (\eta'_A)^\sharp$ are solutions to the equation

$$G(?) \circ \eta_A = \eta_A$$

Since, by definition of free construction, this equation has a unique solution, they are equal, i.e., $\eta_A^\sharp \circ (\eta'_A)^\sharp = id_D$. Interchanging the roles of D and D' in the above argument leads to $(\eta'_A)^\sharp \circ \eta_A^\sharp = id_{D'}$.

□

Example 7.1.6 In proposition 7.1.1 we have, according to definition 7.1.3, that $i = \eta_A$, $S = A$, $M = G(B)$, $D = (List(S), \diamond, [])$, $GD = List(S)$.

Definition 7.1.7 *Let C be a class of Σ-algebras, X_s be a set of variables of sort s for $s \in S$, and $X = (X_s | s \in S)$ be a family of S-sorted pairwise disjoint sets. A Σ-algebra $F(X)$ is called* **free over X in C** *if $F(X) \in C$ and there is an assignment $u : X \to F(X)$, called* **universal mapping**, *such that for every assignment $h : X \to A$ into a Σ-algebra $A \in C$ there is one and only one Σ-homomorphism $h^\sharp$ such that the following diagram commutes, i.e., $h = h^\sharp \circ u$.*

$$
\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "u"'] & A \\
F(X) \arrow[ur, dotted, "h^\sharp"'] &
\end{tikzcd}
$$

Remark 7.1.8 The free construction D over A with respect to G and the universal morphism $\eta_A : A \to GD$ generalizes the notion of a free algebra $F(X)$ over X in C and the universal mapping $u : X \to F(X)$, respectively. We obtain the especial case of 7.1.7 by taking $\mathbf{B}$ to be the class C of Σ-algebras together with all Σ-homomorphisms between algebras in C, $\mathbf{A}$ the category Σ_0-$\mathbf{Alg}$, where Σ_0 is the signature Σ with empty set OP consisting of the sorts of Σ, $G = U : \Sigma - \mathbf{Alg} \to \Sigma_0\text{-}\mathbf{Alg}$ (the forgetful functor) and $A = X$. Note that, since by the above definition, the set of sorts in $\Sigma_0 = (S, \emptyset)$ is equal to to set of sorts in $\Sigma = (S, OP)$, we have that $U(h^\sharp) = h^\sharp$, since the forgetful functor does not forget or change any base set.

Example 7.1.9 The definition of a coproduct of two objects is a special case of the notion of free construction. To see this, consider the diagonal functor $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ of 6.1.1.21 and the product category of 3.4.3.1. Now consider an object $\langle A, B\rangle$ of $\mathbf{C} \times \mathbf{C}$. A free object over $\langle A, B\rangle$ is an object S of C, with a morphism $\langle in_A, in_B\rangle : \langle A, B\rangle \to \Delta(S) = \langle S, S\rangle$ satisfying the universal property.

The universal property says that for any C in $\mathbf{C}$ and any $\mathbf{C} \times \mathbf{C}$-morphism

$$\langle f, g\rangle : \langle A, B\rangle \to \Delta(C) = \langle C, C\rangle$$

there is a unique $\mathbf{C}$-morphism $\langle f, g\rangle^\sharp : S \to C$ such that

$$\Delta(\langle f, g\rangle^\sharp) \circ \langle in_A, in_B\rangle = \langle \langle f, g\rangle^\sharp, \langle f, g\rangle^\sharp\rangle \circ \langle in_A, in_B\rangle = \langle f, g\rangle$$

That is, given any pair of morphism $f : A \to C$ and $g : B \to C$, there is a unique morphism $\langle f, g\rangle^\sharp : S \to C$ such that $\langle f, g\rangle^\sharp \circ in_A = f$ and $\langle f, g\rangle^\sharp \circ in_B = g$. But this is exactly the universal property of coproducts, i.e., S with injections in_A and in_B is the coproduct of A and B (see the next diagram).

$$\begin{tikzcd}
\langle A, B\rangle \arrow[r, "\langle f,g\rangle"] \arrow[d, "\langle in_A, in_B\rangle"'] & \langle C, C\rangle = \Delta(C) \\
\langle S, S\rangle = \Delta(S) \arrow[ur, "\Delta(\langle f,g\rangle^\sharp)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
C \\
S = A + B \arrow[u, "\langle f,g\rangle^\sharp"']
\end{tikzcd}$$

Example 7.1.10 Let $T : \mathbf{D} \to \mathbf{1}$ be the unique functor from $\mathbf{D}$ to $\mathbf{1}$, where $\mathbf{1}$ is the one-object category. A free object over 1 is a $\mathbf{D}$-object A together with an universal morphism $\eta_1 : 1 \to T(A)$, such that for every $\mathbf{D}$-object B and every $\mathbf{1}$-morphism $f : 1 \to T(B)$ there exists a unique $\mathbf{D}$-morphism $f^\sharp$ from A to B satisfying the universal property, i.e., $T(f^\sharp) \circ \eta_1 = f$. However, $\mathbf{1}$ is the category with just one morphism, namely $id_{\mathbf{1}}$. So we have $T(f^\sharp) = id_{\mathbf{1}}$, $\eta_{\mathbf{1}} = id_1$ and $f = id_{\mathbf{1}}$ and the universal property is trivially satisfied. Hence, saying that A is free over 1 is the same as saying that A is an initial object in $\mathbf{D}$ (see the next diagram).

$$\begin{tikzcd}
1 \arrow[r, "id_1 = f"] \arrow[d, "id_1 = \eta_1"'] & 1 = T(B) \\
1 = T(A) \arrow[ur, dashed, "id_1 = T(f^\sharp)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
B \\
A \arrow[u, "f^\sharp = id_1^\sharp"']
\end{tikzcd}$$

7.2 Cofree constructions

The dual notion of free construction is represented by the following

Definition 7.2.1 *Given categories* $\mathbf{C}$ *and* $\mathbf{D}$*, a functor* $F : \mathbf{C} \to \mathbf{D}$*, and an object* A *in* $\mathbf{D}$*, we call an object* D *in* $\mathbf{C}$ *a* **cofree construction over** A **with respect to** F *if there is a morphism* $\epsilon_A : F(D) \to A$ *in* $\mathbf{D}$*, called* **couniversal morphism***, such that for any* $\mathbf{D}$*-morphism* $f : F(D') \to A$ *(*$D' \in \mathbf{C}$*) there is a unique morphism* $f^\bullet : D' \to D \in \mathbf{C}$ *such that* $\epsilon_A \circ G(f^\bullet) = f$*. In this case we say that the following diagram commutes:*

$$
\begin{tikzcd}
A & F(D') \arrow[l, "f"'] \arrow[dl, dotted, "F(f^\bullet)"] \\
F(D) \arrow[u, "\epsilon_A"] &
\end{tikzcd}
$$

Remark 7.2.2 The pair (D, ϵ_A) is normally used to denote the fact that D is cofree over A with respect to F. In the literature of category theory, the pair is also called **(co)universal** from F to A.

Proposition 7.2.3 *Cofree constructions are unique up to isomorphism.*

Proof: $\boxed{\text{D}}$

Example 7.2.4 The definition of a product of two objects is a special case of the notion of a cofree construction. To see this, consider the diagonal functor Δ of 6.1.1.21 and the product category of 3.4.3.1. Now consider an object $\langle A, B\rangle$ of $\mathbf{C} \times \mathbf{C}$. A cofree object over $\langle A, B\rangle$ is a $\mathbf{C}$-object P with a $\mathbf{C} \times \mathbf{C}$-morphism $\langle \pi_A, \pi_B\rangle : \Delta P = \langle P, P\rangle \to \langle A, B\rangle$ satisfying the couniversal property.

The couniversal property says that for any C in $\mathbf{C}$ and any $\mathbf{C} \times \mathbf{C}$-morphism $\langle f, g\rangle : \Delta(C) = \langle C, C\rangle \to \langle A, B\rangle$ there is a unique morphism $\langle f, g\rangle^\bullet : C \to P$ such that

$$\langle \pi_A, \pi_B\rangle \circ \Delta(\langle f, g\rangle^\bullet) = \langle \pi_A, \pi_B\rangle \circ \langle \langle f, g\rangle^\bullet, \langle f, g\rangle^\bullet\rangle = \langle f, g\rangle$$

That is, given any pair of morphisms $f : C \to A$ and $g : C \to B$, there is a unique morphism $\langle f, g\rangle^\bullet : C \to P$ such that $\pi_A \circ \langle f, g\rangle^\bullet = f$ and $\pi_B \circ \langle f, g\rangle^\bullet = g$. But this is exactly the universal property of products, i.e., P with projections π_A and π_B is the product of A and B (see the next diagram).

$$
\begin{tikzcd}
\langle A, B\rangle & \langle C, C\rangle = \Delta(C) \arrow[l, "\langle f, g\rangle"'] \arrow[dl, dotted, "\Delta(\langle f, g\rangle^\bullet)"] & C \arrow[d, "\langle f, g\rangle^\bullet"] \\
\langle P, P\rangle = \Delta(P) \arrow[u, "\langle \pi_A, \pi_B\rangle"] & & P = A \times B
\end{tikzcd}
$$

The following example assume knowledge of what exponetial objects and cartesian closed categories are. It may be skipped on a nth-reading...

Example 7.2.5 Consider now the right product functor $(- \times A)$ of 6.1.1.18. Let $(- \times A) : \mathbf{C} \to \mathbf{C}$, $\mathbf{C}$ a cartesian closed category, and B an object of $\mathbf{C}$. A cofree object over B is an object E of $\mathbf{C}$ with a morphism $eval_{AB} : (- \times A)(E) \to B = E \times A \to B$ satisfying the couniversal property.

The couniversal property says that for any object C of $\mathbf{C}$ and any morphism $f : (- \times A)(C) = C \times A \to B$, there is a unique morphism $f^\bullet : C \to E$ such that

$$eval_{AB} \circ ((- \times A)(f^\bullet) = f$$

i.e., $eval_{AB} \circ (f^\bullet \times id_A) = f$

But this is precisely the definition of an exponential object, if we take $E = B^A$ and $f^\bullet = curry(f)$ (see the next diagram).

$$\begin{tikzcd}
B & & C \times A \arrow[ll, "f"'] \arrow[dll, dashed, "curry(f) \times id_A"] & & C \arrow[d, "curry(f)"] \\
B^A \times A \arrow[u, "eval_{AB}"] & & & & B^A
\end{tikzcd}$$

Remark 7.2.6 Since cofree constructions are unique up to isomorphism, it follows that exponentials objects are (also) unique up to isomorphism.

Example 7.2.7 Let $F : \mathbf{C} \to \mathbf{1}$ be the unique functor from the category $\mathbf{C}$ to the one-object category $\mathbf{1}$. A cofree object over 1 is a $\mathbf{C}$-object T with a morphism $\eta_1 : F(T) \to 1$ such that for every $\mathbf{C}$-object A and every $\mathbf{1}$-morphism $f : F(A) \to 1$ there is a unique $\mathbf{C}$-morphism $f^\bullet : A \to T$ satisfying the couniversal property, i.e., $\epsilon_1 \circ F(f^\bullet) = f$. However, since the only $\mathbf{1}$-morphism is the identity, we have $F(f^\bullet) = f = \epsilon_1 = id_1$ and the couniversal property is trivially satisfied. Hence, saying that T is cofree over 1 is the same as saying that T is a terminal object in $\mathbf{C}$ (see the next diagram, where we take $t_A = f^\bullet$).

$$\begin{tikzcd}
1 & & F(A) \arrow[ll, "f = id_1"'] \arrow[dll, dashed, "G(t_A)"] & & A \arrow[d, "t_A"] \\
F(T) \arrow[u, "\epsilon_1 = id_1"] & & & & T
\end{tikzcd}$$

7.3 Adjoints

The basic data for an **adjoint situation**, or **adjunction**, comprise two categories, $\mathbf{C}$ and $\mathbf{D}$, and functors F and G between them, i.e., $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$. Given a $\mathbf{C}$-object A and a $\mathbf{D}$-object B we obtain

$$\begin{tikzcd} A \arrow[rr, "F"] \arrow[d, dashed] & & F(A) \arrow[d, dashed] \\ G(B) & & B \arrow[ll, "G"] \\ \text{in } \mathbf{C} & & \text{in } \mathbf{D} \end{tikzcd}$$

$G(B)$ in $\mathbf{C}$ and $F(A)$ in $\mathbf{D}$. An adjunction occurs when there is an exact correspondence of morphisms between these objects in the directions indicated by the broken arrows in the above picture, so that any passage from A to $G(B)$ in $\mathbf{C}$ is matched uniquely by a passage from $F(A)$ to B in $\mathbf{D}$, that is to say, for any objects $A \in \mathbf{C}, B \in \mathbf{D}$ we have a bijection between the hom-sets $Hom_{\mathbf{C}}(A, G(B))$ and $Hom_{\mathbf{D}}(F(A), B)$, written $Hom_{\mathbf{C}}(A, G(B)) \cong Hom_{\mathbf{D}}(F(A), B)$. We will come back later on to this topic. We now want to establish a relation between the existence of free constructions and the functor F above, respectively, the existence of cofree constructions and the functor G above.

The first of these relations is established by the next

Theorem 7.3.1 *Consider the functor $G : \mathbf{D} \to \mathbf{C}$. Suppose that for each A of $\mathbf{C}$, a free object $D \in \mathbf{D}$ with universal morphism η_A exists. Then the free construction D can be extended to morphisms in $\mathbf{C}$ such that for each $h : A \to B$ in $\mathbf{C}$, $(\eta_B \circ h)^\sharp : D \to D'$ is uniquely defined by commutativity of the following diagram:*

$$\begin{tikzcd} I_{\mathbf{C}}(A) \arrow[r, "I_{\mathbf{C}}(h)"] \arrow[d, "\eta_A"'] & I_{\mathbf{C}}(B) \arrow[d, "\eta_B"] \\ G(D) \arrow[r, "G((\eta_B \circ h)^\sharp)"'] & G(D') \end{tikzcd}$$

Taking $F(A) = D, F(B) = D'$ we obtain a functor $F : \mathbf{C} \to \mathbf{D}$ called **free functor with respect to G**, *and a natural transformation $\eta : I_{\mathbf{C}} \dot{\to} G \circ F$, called* **universal transformation**, *such that $F(h) = (\eta_B \circ h)^\sharp$.*

Proof: Consider the next diagram (where we omit the identity functor):

$$\begin{tikzcd}
A \arrow[r, "h"] \arrow[d, "\eta_A"'] & B \arrow[r, "\eta_B"] & GF(B) \\
GF(A) \arrow[urr, "G(h^\sharp)=GF(h)"'] & &
\end{tikzcd}$$

We define $f = \eta_B \circ h$ and we obtain a unique $f^\sharp = (\eta_B \circ h)^\sharp : F(A) \to F(B)$ in $\mathbf{D}$ such that the above diagram commutes (i.e., $G(h^\sharp) \circ \eta_A = f$) by the universal property of $F(A)$. Now we define for each such $h, F(h) = (\eta_B \circ h)^\sharp$. Note that uniqueness of $F(h)$ with respect to the situation pictured in the above diagram shows that F is well-defined with respect to morphisms. To show that $F : \mathbf{C} \to \mathbf{D}$ is already a functor it is necessary to show preservation of identities and composition.

1. F preserves identities. Consider the following diagram:

$$\begin{tikzcd}
A \arrow[r, "id_A"] \arrow[d, "\eta_A"'] & A \arrow[d, "\eta_A"] \\
GF(A) \arrow[r, "GF(id_A)"'] & GF(A)
\end{tikzcd}$$

Since $F(A)$ is a free construction, $F(id_A) = (\eta_A \circ id_A)^\sharp$ is the unique $\mathbf{D}$-morphism that makes the above diagram commute. Now $id_{GF(A)}$ also does the job, since $id_{GF(A)} \circ \eta_A = \eta_A = \eta_A \circ id_A$. Now, since G is a functor we have that $id_{GF(A)} = G(id_{F(A)})$. So, both $id_{F(A)}$ and $F(id_A)$ are solutions to the equation

$$\eta_A \circ id_A = G(?) \circ \eta_A$$

Since, by definition of free construction, this equation has a unique solution, they are equal. This shows that $F(id_A) = id_{F(A)}$

2. F preserves compositions. Consider the following diagram:

$$\begin{tikzcd}
A \arrow[r, "h"] \arrow[d, "\eta_A"'] \arrow[dr, phantom, "(1)"] & B \arrow[r, "g"] \arrow[d, "\eta_B"] \arrow[dr, phantom, "(2)"] & C \arrow[d, "\eta_C"] \\
GF(A) \arrow[r, "GF(h)"'] & GF(B) \arrow[r, "GF(g)"'] & GF(C)
\end{tikzcd}$$

By the universal properties of $F(A)$ and $F(B)$, the diagrams (1) and (2) commute, respectively. Since (1) and (2) commute, we have that the whole diagram commute. To see this note that

$\eta_C \circ g \circ h =$
$= GF(g) \circ \eta_B \circ h$ by (2)

$= GF(G) \circ GF(h) \circ \eta_A$ by (1)

Thus we have shown that the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "g \circ h"] \arrow[d, "\eta_A"'] & C \arrow[d, "\eta_C"] \\
GF(A) \arrow[r, "GF(g) \circ GF(h)"'] & GF(C)
\end{tikzcd}$$

Now $GF(g) \circ GF(h) = G(F(g) \circ F(h))$, since G is a functor. However, by universal property of $F(A)$, $F(g \circ h)$ is the unique **D**-morphism that make the above diagram commute. Thus, both $F(g \circ h)$ and $F(g) \circ F(h)$ are solutions to the equation

$$\eta_C \circ (g \circ h) = G(?) \circ \eta_A$$

Since, by definition of free construction, this equation has a unique solution, they are equal. This shows that $F(g \circ h) = F(g) \circ F(h)$.

Now, to see that the family $\eta_A : A \to GF(A)$, for each $A \in \mathbf{C}$ constitute a natural transformation $\eta : I \to G \circ F$, it is necessary to show that the following diagram commutes (omitting the identity functor) for each $h : A \to B$:

$$\begin{tikzcd}
A \arrow[d, "h"'] & A \arrow[r, "\eta_A"] \arrow[d, "h"'] & G(F(A)) \arrow[d, "G(F(h))"] \\
B & B \arrow[r, "\eta_B"'] & G(F(B))
\end{tikzcd}$$

Since we have already shown that $F : \mathbf{C} \to \mathbf{D}$ is a functor, the above diagram commutes for each $h : A \to B \in \mathbf{C}$ by the claim made in the beginning of this proof, i.e., by the fact that $F(A)$ is a free construction.

$\square$

Conversely, by assuming the existence of a functor $F : \mathbf{C} \to \mathbf{D}$, we can state the next

Theorem 7.3.2 *Consider the functor $F : \mathbf{C} \to \mathbf{D}$. Suppose that for each A of $\mathbf{D}$, a cofree object $G(A) \in \mathbf{C}$ with couniversal morphism $\epsilon_A : FG(A) \to A$ exists. Then the cofree construction $G(A)$ can be extended to morphisms in $\mathbf{D}$ such that for each $h : B \to A$ in $\mathbf{D}$, $G(h) = (h \circ \epsilon_B)^\bullet : G(B) \to G(A)$ is uniquely defined by commutativity of the following diagram:*

$$\begin{array}{ccc} I_{\mathbf{D}}(A) & \xleftarrow{I_{\mathbf{D}}(h)} & I_{\mathbf{D}}(B) \\ \uparrow \epsilon_A & & \uparrow \epsilon_B \\ FG(A) & \xleftarrow[F \circ G(h)]{} & FG(B) \end{array}$$

In this way we obtain a functor $G : \mathbf{D} \to \mathbf{C}$ called **cofree functor with respect to G**, *and a natural transformation $\epsilon : F \circ G \dot{\to} I_{\mathbf{D}}$, called* **couniversal natural transformation**.

Proof: $\boxed{\text{D}}$

Definition 7.3.3 *Let $\mathbf{A}$ and $\mathbf{B}$ be categories. If $F : \mathbf{A} \to \mathbf{B}$ and $G : \mathbf{B} \to \mathbf{A}$ are functors, we say that F is a* **left adjoint** *to G and G is* **right adjoint** *to F provided there is a natural transformation $\eta : I_{\mathbf{C}} \dot{\to} G \circ F$ such that for any objects A of $\mathbf{A}$ and B of $\mathbf{B}$ and any morphism $f : A \to G(B)$, there is a unique morphism $f^{\sharp} : FA \to B$ such that $f = G(f^{\sharp}) \circ \eta_A$.*

Remarks 7.3.4

1. Note that the above definition states that $F(A)$ is a free construction over A with respect to G.

2. The characterization of an adjunction in terms of a bijection between hom-sets (see our discussion in the beginning of this section) as well as its equivalence with the above definition will be the work of theorem 7.3.11.

3. It is usual to write $F \dashv G$ to denote the situation described in the above definition. The data (F, G, η) constitute an adjunction. The natural transformation η is called the **unit** of adjunction.

4. A given functor F may or may not have a right adjoint. The forgetful functor on a variety of algebras has a left adjoint, the **free algebra functor**, which takes a set S to the free algebra generated by the elements of S and a function $f : S \to S'$ to f's unique extension to a homomorphism.

5. In some texts, the adjunction is written as a rule of inference like this:

$$\frac{A \to G(B)}{F(A) \to B}$$

   which calls attention to the symmetry present in the definition.

6. Associated with each adjunction is another natural transformation $\epsilon : F \circ G \dot{\to} I_{\mathbf{D}}$, called the **counit** of the adjunction, with the property that for each $\mathbf{B}$-morphism $g : F(A) \to B$, there is a unique $\mathbf{A}$-morphism $g^{\bullet} : A \to G(B)$ for which the following diagram commutes:

$$\begin{tikzcd} B & FG(B) \arrow[l, "\epsilon_B"'] \\ & F(A) \arrow[ul, "g"] \arrow[u, "F(g^\bullet)"'] \end{tikzcd}$$

7. It can be shown that the existence of a unit implies the existence of a counit, and vice versa. This is the task of our next

Proposition 7.3.5 *Suppose that $F : \mathbf{A} \to \mathbf{B}$ and $G : \mathbf{B} \to \mathbf{A}$ are functors such that $F \dashv G$. Then there is a natural transformation $\epsilon : F \circ G \dot{\to} I_{\mathbf{B}}$ such that for any $g : F(A) \to B$, there is a unique $g^\bullet : A \to G(B)$ such that the following diagram commutes:*

$$\begin{tikzcd} G(B) \\ A \arrow[u, "g^\bullet"'] \end{tikzcd} \qquad \begin{tikzcd} B & FG(B) \arrow[l, "\epsilon_B"'] \\ & F(A) \arrow[ul, "g"] \arrow[u, "F(g^\bullet)"'] \end{tikzcd}$$

Proof: It is a immediate consequence of categorical duality that this proposition is reversible and the adjunction is equivalent to the existence of either natural transformation η or ϵ with its appropriate universal mapping property.

Now take $A = G(B)$ in the definition of adjoint. Then corresponding to the identity morphism $id_{G(B)} : G(B) \to G(B)$, there is a unique $id^\sharp_{G(B)}$ such that the following diagram commutes:

$$\begin{tikzcd} G(B) \arrow[r, "id_{G(B)}"] \arrow[d, "\eta_{G(B)}"'] \arrow[dr, phantom, "(1)"] & G(B) \\ GFG(B) \arrow[ur, "G(id^\sharp_{G(B)} = \epsilon_B)"'] \end{tikzcd} \qquad \begin{tikzcd} B \\ FG(B) \arrow[u, "id^\sharp_{G(B)}"] \end{tikzcd}$$

We define $\epsilon_B = id^\sharp_{GB}$. To show that ϵ_B is the component at B of a natural transformation $\epsilon : F \circ G \dot{\to} I_{\mathbf{B}}$, we have to show that for every $k : B' \to B \in \mathbf{B}$ the following diagram commutes:

$$\begin{tikzcd} B' \arrow[d, "k"'] \\ B \end{tikzcd} \qquad \begin{tikzcd} FG(B') \arrow[r, "\epsilon_{B'}"] \arrow[d, "FG(k)"'] & B' \arrow[d, "k"] \\ FG(B) \arrow[r, "\epsilon_B"'] & B \end{tikzcd}$$

To see this, consider the following diagram:

$$\begin{tikzcd}
G(B') \arrow[r, "\eta_{G(B')}"] \arrow[rr, bend left, "id_{G(B')}"] \arrow[d, "\eta_{G(B')}"'] & GFG(B') \arrow[r, "G(\epsilon_{G(B')})"] & G(B') \arrow[r, "G(k)"] & G(B) \\
GFG(B') \arrow[urrr, "G(?)"'] & & &
\end{tikzcd}$$

where $G(\epsilon_{G(B')}) \circ \eta_{G(B')} = id_{G(B')}$ by the diagram (1) above, taking $G(B')$ instead of $G(B)$. We first put $? = k \circ \epsilon_{B'}$. Now we have:

$$\begin{aligned}
G(k \circ \epsilon_{B'}) \circ \eta_{G(B')} &= G(k) \circ G(\epsilon_{B'}) \circ \eta_{G(B')} && (G \text{ is a functor}) \\
&= G(k) \circ id_{G(B')} && (\text{observation above}) \\
&= G(k) && (\text{identity})
\end{aligned}$$

while if we take $? = \epsilon_B \circ FG(k)$ we have:

$$\begin{aligned}
G(\epsilon_B \circ FG(k)) \circ \eta_{G(B')} &= G(\epsilon_B) \circ GFG(k) \circ \eta_{G(B')} && (G \text{ is a functor}) \\
&= G(\epsilon_B) \circ \eta_{G(B)} \circ G(k) && (\text{naturality of } \eta \text{ in (2) below}) \\
&= id_{G(B)} \circ G(k) && (\text{by (1)}) \\
&= G(k) && (\text{identity})
\end{aligned}$$

$$\begin{tikzcd}
B' \arrow[d, "k"'] & G(B') \arrow[r, "\eta_{G(B')}"] \arrow[d, "k"'] \arrow[dr, phantom, "(2)"] & GFG(B') \arrow[d, "GFG(k)"] \\
B & G(B) \arrow[r, "\eta_{G(B)}"'] & GFG(B)
\end{tikzcd}$$

This shows that both $k \circ \epsilon_{B'}$ and $\epsilon_B \circ FG(k)$ are solutions to the equation

$$G(?) \circ \eta_{G(B')} = G(h)$$

Since this equation, by the universal property of $FG(B')$, has a unique solution, they must be equal. This shows that ϵ is a natural transformation from $F \circ G$ to $I_{\mathbf{B}}$.

It remains to show that given $g : F(A) \to B$ in $\mathbf{B}$, there is a unique $g^\bullet : A \to G(B) \in \mathbf{A}$ such that the following diagram commutes (i.e., we have to show that $G(B)$ is cofree over B with respect to $F : \mathbf{A} \to \mathbf{B}$):

$$\begin{tikzcd}
G(B) & B & FG(B) \arrow[l, "\epsilon_B"'] & (3) \\
A \arrow[u, "g^\bullet"'] & F(A) \arrow[u, "g"] \arrow[ur, "F(g^\bullet)"'] & &
\end{tikzcd}$$

We define $g^\bullet = G(g) \circ \eta_A$ according to the following diagram:

$$\begin{tikzcd} A \arrow[r, "g^\bullet"] \arrow[d, "\eta_A"'] & GB \\ GFA \arrow[ur, "G(g)"'] & \end{tikzcd} \qquad \begin{tikzcd} B \\ FA \arrow[u, "g"'] \end{tikzcd}$$

Then we have:

$$\begin{aligned} \epsilon_B \circ F(G(g) \circ \eta_A) &= \epsilon_B \circ F(G(g)) \circ \eta_A & (G \text{ is a functor}) \\ &= g \circ \epsilon_{F(A)} \circ F(\eta_A) & (\text{by naturality of } \epsilon \text{ below}) \end{aligned}$$

$$\begin{tikzcd} FGF(A) \arrow[r, "\epsilon_{F(A)}"] \arrow[d, "FGF(g)"'] & F(A) \arrow[d, "g"] \\ FG(B) \arrow[r, "\epsilon_B"'] & B \end{tikzcd}$$

It suffices to show that $\epsilon_{F(A)} \circ F(\eta_A) = id_{F(A)}$ (why ?). Therefore, consider the following diagram:

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[d, "\eta_A"'] & GF(A) \\ GF(A) \arrow[ur, "G(?)"'] & \end{tikzcd} \qquad \begin{tikzcd} F(A) \\ F(A) \arrow[u, "?"'] \end{tikzcd}$$

We put first $? = id_{F(A)}$ and we have trivially $G(id_{F(A)}) \circ \eta_A = id_{GF(A)} \circ \eta_A = \eta_A$. We now fix $? = \eta_{F(A)} \circ F(\eta_A)$ and we have:

$$\begin{aligned} G(\epsilon_{F(A)} \circ F(\eta_A)) \circ \eta_A &= G(\epsilon_{F(A)}) \circ GF(\eta_A) \circ \eta_A & (G \text{ is a functor}) \\ &= G(\epsilon_{F(A)}) \circ \eta_{GF(A)} \circ \eta_A & (\text{naturality of } \eta \text{ below}) \\ &= id_{GF(A)} \circ \eta_A & (\text{taking } F(A) = B \text{ and using (1)}) \\ &= \eta_A & (\text{identity}) \end{aligned}$$

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[d, "\eta_A"'] & GF(A) \arrow[d, "GF(\eta_A)"] \\ GF(A) \arrow[r, "\eta_{GF(A)}"'] & GFGF(A) \end{tikzcd}$$

Now both $id_{F(A)}$ and $\eta_{F(A)} \circ F(\eta_A)$ are solutions to the equation

$$G(?) \circ \eta_A = \eta_A$$

Since this equation, by the universal property of $F(A)$, has a unique solution, they must be equal. This shows that the next diagram commutes

$$\begin{tikzcd}
FA \arrow[r, "id_{FA}"] \arrow[d, "F(\eta_A)"'] \arrow[dr, phantom, "(3)"] & FA \\
FGFA \arrow[ur, "\eta_{FA}"'] &
\end{tikzcd}$$

and hence that $\epsilon_B \circ F(g^\bullet) = g$.

It remains to show uniqueness of $g^\bullet = G(g) \circ \eta_A : A \to G(B)$. Therefore, assume the existence of a $h : A \to G(B) \in \mathbf{A}$ such that $\epsilon_B \circ F(h) = g$. We must show that $g^\bullet = G(g) \circ \eta_A = h$. Now consider the following diagram:

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[d, "h"'] & GF(A) \arrow[d, "GF(h)"] \arrow[dd, bend left=60, "G(g)"] \\
G(B) \arrow[r, "\eta_{G(B)}"'] \arrow[dr, "id_{G(B)}"'] & GFG(B) \arrow[d, "G(\epsilon_B)"] \\
& G(B)
\end{tikzcd}$$

The following simple calculation yields the desired result:

$$\begin{array}{rcll}
g^\bullet & = & G(g) \circ \eta_A & \text{(by definition)} \\
& = & G(\epsilon_B \circ F(h)) \circ \eta_A & \text{(assumption)} \\
& = & G(\epsilon_B) \circ GF(h) \circ \eta_A & \text{(G is a functor)} \\
& = & G(\epsilon_B) \circ \eta_{G(B)} \circ h & \text{(naturality of η)} \\
& = & id_{G(B)} \circ h & \text{(by (1))} \\
& = & h & \text{(identity)}
\end{array}$$

Remark 7.3.6 Note that the above proposition states that each free functor $F : \mathbf{A} \to \mathbf{B}$ determines the existence of a cofree functor $G : \mathbf{B} \to \mathbf{A}$ and vice-versa.

Proposition 7.3.7 *Left adjoints are unique up to natural isomorphism.*

Proof: Given free functors $F : \mathbf{C} \to \mathbf{D}$ and $F' : \mathbf{C} \to \mathbf{D}$ with natural transformations η and η', we obtain unique morphisms $\tau_A : F(A) \to F'(A)$ and $\tau'_A : F'(A) \to F(A)$, for each $A \in \mathbf{C}$, such that the following diagram commutes:

$$\begin{tikzcd}
& A \arrow[dl, dotted, "\eta_A"'] \arrow[d, "\eta'_A"] \arrow[dr, "\eta_A"] & \\
GF(A) \arrow[r, "G(\tau_A)"'] & GF'(A) \arrow[r, "G(\tau'_A)"'] & GF(A)
\end{tikzcd}$$

To see that it really commutes note that we have:

$G(\tau'_A) \circ G(\tau_A) \circ \eta_A =$
$= G(\tau'_A) \circ \eta'_A$ $\qquad$ ($F(A)$ is a free construction)
$= \eta_A$ $\qquad$ ($F'(A)$ is a free construction)

However, $id_{F(A)}$ also makes the above diagram commute, i.e., $G(id_{F(A)}) \circ \eta_A = id_{GF(A)} \circ \eta_A = \eta_A$. Now note that $G(\tau'_A) \circ G(\tau_A) = G(\tau'_A \circ \tau_A)$, since G is a functor. Thus, both $id_{F(A)}$ and $\tau'_A \circ \tau_A$ are solutions to the equation

$$G(?) \circ \eta_A = \eta_A$$

Since, by definition of a left adjoint, this equation has a unique solution, they are equal. This shows that $\tau'_A \circ \tau_A = id_{F(A)}$ for each A in $\mathbf{C}$. Interchanging the roles of $F(A)$ and $F'(A)$ in the above argument leads to $\tau_A \circ \tau'_A = id_{F'(A)}$ for each $A \in \mathbf{C}$. To show that this family of isomorphisms is natural with respect to morphisms in $\mathbf{C}$ (and hence a natural isomorphism), we have to show that for each $f : A \to B \in \mathbf{C}$ the following diagram commutes:

$$\begin{tikzcd} A \arrow[d, "f"'] & F(A) \arrow[r, "\tau_A"] \arrow[d, "F(f)"'] & F'(A) \arrow[d, "F'(f)"] \\ B & F(B) \arrow[r, "\tau_B"'] & F'(B) \end{tikzcd}$$

Now consider the following diagram,where we use the knowledge of the existence of the natural transformations $\eta : I \dot{\to} G \circ F, \eta' : I \dot{\to} G \circ F'$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\eta_A"'] & B \arrow[r, "\eta'_B"] & G(F'(B)) \\ G(F(A)) \arrow[urr, "G(?)"'] & & \end{tikzcd}$$

We put $? = F'(f) \circ \tau_A$ and we have:

$$\begin{aligned} G(F'(f) \circ \tau_A) \circ \eta_A &= G(F'(f)) \circ G(\tau_A) \circ \eta_A && (G \text{ is a functor}) \\ &= G(F'(f)) \circ \eta'_A && (\text{diagram (1) below}) \\ &= \eta'_B \circ f && (\text{naturality of } \eta' \text{ in (2) below}) \end{aligned}$$

$$\begin{tikzcd} A \arrow[r, "\eta'_A"] \arrow[d, "\eta_A"'] \arrow[dr, phantom, "(1)"] & GF'(A) \\ GF(A) \arrow[ur, "G(\tau_A)"'] & \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\eta'_A"] \arrow[d, "f"'] \arrow[dr, phantom, "(2)"] & GF'(A) \arrow[d, "GF'(f)"] \\ B \arrow[r, "\eta'_B"'] & GF'(B) \end{tikzcd}$$

We now set $? = \tau_B \circ F(f)$ and we have:

$$\begin{array}{rcll} G(\tau_B \circ F(f)) \circ \eta_A & = & G(\tau_B) \circ GF(f) \circ \eta_A & (G \text{ is a functor}) \\ & = & G(\tau_B) \circ \eta_B \circ f & (\text{naturality of } \eta) \\ & = & \eta'_B \circ f & (\text{diagram (3) below}) \end{array}$$

$$\begin{tikzcd} B \arrow[r, "\eta'_B"] \arrow[d, "\eta_B"'] \arrow[dr, phantom, "(3)"] & GF'(B) \\ GF(B) \arrow[ur, "G(\tau_B)"'] & \end{tikzcd}$$

This shows that both $F'(f) \circ \tau_A$ and $\tau_B \circ F(f)$ are solutions to the equation

$$G(?) \circ \eta_A = \eta'_B \circ f$$

Since $F(A)$ is a free construction, they must be equal. Hence, $F'(f) \circ \tau_A = \tau_B \circ F(f)$ and we are done.

The fact that $\tau' : F' \dot{\rightarrow} F$ is also natural follows directly from 6.2.3.3.

□

Proposition 7.3.8 *Right adjoints are unique up to natural isomorphism.*

Proof: $\boxed{\text{D}}$

Proposition 7.3.9 *Left adjoints are closed under composition, i.e., if* $F : \mathbf{A} \rightarrow \mathbf{B}$ *and* $F' : \mathbf{B} \rightarrow \mathbf{C}$ *are left adjoints with respect to* $G : \mathbf{B} \rightarrow \mathbf{A}$ *and* $G' : \mathbf{C} \rightarrow \mathbf{B}$*, then* $F' \circ F : \mathbf{A} \rightarrow \mathbf{C}$ *is left adjoint with respect to* $G \circ G' : \mathbf{C} \rightarrow \mathbf{A}$*.*

Proof: Given left adjoints F and F' we have to show the universal property of $F' \circ F$ with respect to $G \circ G'$. Now, given a morphism $f : A \rightarrow G \circ G'(C)$ $(C \in \mathbf{C})$ in $\mathbf{A}$, we obtain, by the universal property of FA, a unique $f^\sharp : FA \rightarrow G'(C)$ in $\mathbf{B}$ such that the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\eta_A"'] \arrow[dr, phantom, "(1)"] & GG'(C) \\ GFA \arrow[ur, dashed, "G(f^\sharp)"'] & \end{tikzcd}$$

In the same way, given a $\mathbf{B}$-morphism $f^\sharp : FA \rightarrow G'(C)$, we obtain, by the universal property of $F'FA$, a unique $(f^\sharp)^\sharp : F'FA \rightarrow C$ such that the following diagram commutes:

$$
\begin{tikzcd}
FA \arrow[r, "f^\sharp"] \arrow[d, "\eta_{FA}"'] \arrow[dr, phantom, "(2)"] & G'(C) \\
G'F'FA \arrow[ur, dotted, "G'((f^\sharp)^\sharp)"'] &
\end{tikzcd}
$$

Now, since $G : \mathbf{B} \to \mathbf{A}$ is a functor, we can apply it to (2) to obtain the following commutative diagram.

$$
\begin{tikzcd}
GFA \arrow[r, "G(f^\sharp)"] \arrow[d, "G(\eta_{FA})"'] \arrow[dr, phantom, "(3)"] & GG'(C) \\
GG'F'FA \arrow[ur, dotted, "GG'((f^\sharp)^\sharp)"'] &
\end{tikzcd}
$$

Note that the commutativity is implied by the axiom $G(g \circ f) = G(g) \circ G(f)$.

Now if we paste the diagrams (1) and (3) together and define $\alpha_A = G(\eta_{FA}) \circ \eta_A$, we obtain:

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\alpha_A"'] \arrow[dr, phantom, "(4)"] & GG'(C) \\
GG'F'FA \arrow[ur, dotted, "GG'((f^\sharp)^\sharp)"'] &
\end{tikzcd}
$$

To see that this diagram commutes, note that

$G \circ G'((f^\sharp)^\sharp) \circ \alpha_A =$

$= G \circ G'((f^\sharp)^\sharp) \circ G(\eta_{FA}) \circ \eta_A$ (by definition of α_A)

$= G(f^\sharp) \circ \eta_A$ (by (3))

$= f$ (by (1))

It remains to show that $(f^\sharp)^\sharp : F'FA \to C$ is the unique $\mathbf{C}$-morphism such that (4) commutes.

Therefore, suppose there exists another $\mathbf{C}$-morphism $g : F'FA \to C$ such that (4) commutes, i.e., such that

$$G \circ G'(g) \circ G(\eta_{FA}) \circ \eta_A = f.$$

Now, since G is a functor we have that

$$G \circ G'(g) \circ G(\eta_{FA}) \circ \eta_A = G(G'(g) \circ \eta_{FA}) \circ \eta_A$$

Thus, both $f^\sharp$ and $G'(g) \circ \eta_{FA}$ are solutions to the equation

$$G(?) \circ \eta_A = f$$

However, since FA is a free construction (see (1)), this equation has a unique solution, and hence we must have $G'(g) \circ \eta_{FA} = f^\sharp$. Again, both $(f^\sharp)^\sharp$ (see (2)) and g are solutions to the equation

$$G'(?) \circ \eta_{FA} = f^\sharp$$

Since $F'FA$ is a free construction, this equation has a unique solution, and hence $g = (f^\sharp)^\sharp$, as required.

$\square$

Example 7.3.10 The composition $F' \circ F : \mathbf{Set} \to \mathbf{Mon}$ is the well-known free construction assigning to each set A the free monoid $F'FA = F'(A^+, \diamond) = (A^*, \lambda, \diamond)$, where $(A^+, \diamond)$ is the free semigroup generated by A.

The symmetry intrinsic to an adjoint situation, which was used as a motivation for the concept of adjunction, is explicit in the following

Theorem 7.3.11 *If $F : \mathbf{A} \to \mathbf{B}, G : \mathbf{B} \to \mathbf{A}$ are functors, then $F \vdash G$ if and only if $Hom_{\mathbf{B}}(F-, -)$ and $Hom_{\mathbf{A}}(-, G-)$ are naturally isomorphic as functors $\mathbf{A}^{\mathbf{op}} \times \mathbf{B} \to \mathbf{Set}$.*

Proof: "$\Rightarrow$" Let $F \vdash G, A \in \mathbf{A}, B \in \mathbf{B}$. Then, by definition, the following diagram commutes with a unique $f^\sharp : FA \to B \in \mathbf{B}$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\eta_A"'] & GB & & B \\ GFA \arrow[ur, "G(f^\sharp)"'] & & & FA \arrow[u, "f^\sharp"'] \end{tikzcd}$$

Now, given two bifunctors (see 6.1.1.25)

$$Hom_{\mathbf{B}}(F-, -) : \mathbf{A}^{\mathbf{op}} \times \mathbf{B} \to \mathbf{Set}$$
$$Hom_{\mathbf{A}}(-, G-) : \mathbf{A}^{\mathbf{op}} \times \mathbf{B} \to \mathbf{Set}$$

where, for each $\langle A, B\rangle \in \mathbf{A}^{\mathbf{op}} \times \mathbf{B}$, $Hom_{\mathbf{B}}(F-, -)(\langle A, B\rangle) = Hom_{\mathbf{B}}(-, -) \circ (F^{op} \times I_{\mathbf{B}})(\langle A, B\rangle)$ and $Hom_{\mathbf{A}}(-, G-)(\langle A, B\rangle) = Hom_{\mathbf{A}}(-, -) \circ (I_{\mathbf{A}} \times G)(\langle A, B\rangle)$. Now, we define the natural transformation $(-)^\bullet : Hom_{\mathbf{B}}(F-, -) \dot{\to} Hom_{\mathbf{A}}(-, G-)$ at each component $\langle A, B\rangle \in \mathbf{A}^{\mathbf{op}} \times \mathbf{B}$ for each $g : F(A) \to B \in Hom_{\mathbf{B}}(FA, B)$ by

$$(g)^{\bullet}_{\langle A,B\rangle} = G(g) \circ \eta_A$$

Now, according to 7.3.5, the existence of a unit η implies the existence of a counit ϵ, such that given $A \in \mathbf{A}, B \in \mathbf{B}$ the next diagram commutes with a unique $g^{\bullet} : A \to GB$:

$$\begin{tikzcd} GB & & B & FG(B) \arrow[l, "\epsilon_B"'] \\ A \arrow[u, "g^{\bullet}"'] & & FA \arrow[u, "g"] \arrow[ur, "F(g^{\bullet})"'] & \end{tikzcd}$$

Thus we also define the natural transformation $(-)^{\sharp} : Hom_{\mathbf{A}}(-, G-) \dot{\to} Hom_{\mathbf{B}}(F-, -)$ at each component $\langle A, B\rangle \in \mathbf{A}^{\mathbf{op}} \times \mathbf{B}$ for each $f : A \to GB \in Hom_{\mathbf{A}}(A, GB)$ by

$$(f)^{\sharp}_{\langle A,B\rangle} = \epsilon_B \circ F(f)$$

Now, to see that $((-)^{\sharp}_{\langle A,B\rangle})^{\bullet}_{\langle A,B\rangle} = I_{Hom_A(A,GB)}$, consider the following calculations, for some $f : A \to GB \in Hom_{\mathbf{A}}(A, GB)$

$$\begin{array}{rcll} ((f)^{\sharp}_{\langle A,B\rangle})^{\bullet}_{\langle A,B\rangle} & = & (\epsilon_B \circ F(f))^{\bullet}_{\langle A,B\rangle} & \text{(by definition of } (-)^{\sharp}) \\ & = & G(\epsilon_B \circ F(f)) \circ \eta_A & \text{(by definition of } (-)^{\bullet}) \\ & = & G(\epsilon_B) \circ GF(f) \circ \eta_A & (G \text{ is a functor}) \\ & = & G(\epsilon_B) \circ \eta_{GB} \circ f & \text{(diagram (1) below)} \\ & = & id_{GB} \circ f & \text{(diagram (1) in 7.3.5)} \\ & = & f & \text{(identity)} \end{array}$$

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[d, "f"'] \arrow[dr, phantom, "(1)"] & GF(A) \arrow[d, "GF(f)"] \\ G(B) \arrow[r, "\eta_{GB}"'] & GFG(B) \end{tikzcd}$$

On the other hand, to to see that $((-)^{\bullet}_{\langle A,B\rangle})^{\sharp}_{\langle A,B\rangle} = I_{Hom_B(FA,B)}$, consider the following calculations, for some $g : FA \to B \in Hom_{\mathbf{B}}(FA, B)$:

$$\begin{array}{rcll} ((g)^{\bullet}_{\langle A,B\rangle})^{\sharp}_{\langle A,B\rangle} & = & (G(g) \circ \eta_A)^{\sharp} & \text{(by definition of } (-)^{\bullet}) \\ & = & \epsilon_B \circ F(G(g) \circ \eta_A) & \text{(by definition of } (-)^{\sharp}) \\ & = & \epsilon_B \circ F(G(g)) \circ F(\eta_A) & (F \text{ is a functor}) \\ & = & g \circ \epsilon_{F(A)} \circ F(\eta_A & \text{(naturality of } \epsilon \text{ in (2) below)} \\ & = & g \circ id_{F(A)} & \text{(diagram (3) in 7.3.5)} \\ & = & g & \text{(identity)} \end{array}$$

$$
\begin{tikzcd}
F(A) \arrow[d, "g"] & FGF(A) \arrow[r, "\epsilon_{FA}"] \arrow[d, "FG(g)"'] \arrow[dr, phantom, "(2)"] & F(A) \arrow[d, "g"] \\
B & FG(B) \arrow[r, "\epsilon_B"'] & B
\end{tikzcd}
$$

This shows that the family $(-)^{\bullet}_{\langle A,B\rangle}$ for each $\langle A, B\rangle \in \mathbf{A^{op}} \times \mathbf{B}$ is a family of bijective functions in **Set** with inverse $(-)^{\sharp}_{\langle A,B\rangle}$. It remains to show naturality of $(-)^{\bullet}$ and $(-)^{\sharp}$.

To see this, we have to show that for each $\langle g^{op}, h\rangle \in \mathbf{A^{op}} \times \mathbf{B}, g : A' \to A \in \mathbf{A}, h : B \to B' \in \mathbf{B}$, the following diagram commutes:

$$
\begin{tikzcd}
\langle A, B\rangle \arrow[d, "\langle g^{op},h\rangle"'] & Hom_{\mathbf{B}}(FA, B) \arrow[r, "(-)^{\bullet}_{\langle A,B\rangle}"] \arrow[d, "Hom_{\mathbf{A}}(F(g^{op}),h)"'] & Hom_{\mathbf{A}}(A, GB) \arrow[d, "Hom_{\mathbf{A}}(g^{op},G(h))"] \\
\langle A', B'\rangle & Hom_{\mathbf{B}}(\langle FA', B'\rangle) \arrow[r, "(-)^{\bullet}_{\langle A',B'\rangle}"'] & Hom_{\mathbf{A}}(A, GB')
\end{tikzcd}
$$

Now, picking any $f : FA \to B$ in the upper left corner, and chasing the diagram counter-clockwise, we have:

$$
\begin{aligned}
(Hom_{\mathbf{B}}(F(g^{op}), h)(f))^{\bullet}_{\langle A',B'\rangle} &= (h \circ f \circ F(g))^{\bullet} \\
&\quad \text{(by definition of } Hom_{\mathbf{B}}(F(g^{op}), h)) \\
&= G(h \circ G(f) \circ GF(g)) \circ \eta_{A'} \\
&\quad \text{(by definition of } (-)^{\bullet}) \\
&= G(h) \circ G(f) \circ GF(g) \circ \eta_{A'} \\
&\quad (G \text{ is a functor}) \\
&= G(h) \circ G(f) \circ \eta_A \circ g \\
&\quad \text{(naturality of } \eta \text{ in (3) below)}
\end{aligned}
$$

$$
\begin{tikzcd}
A' \arrow[r, "\eta_{A'}"] \arrow[d, "g"'] & GF(A') \arrow[d, "GF(g)"'] \\
A \arrow[r, "\eta_A"'] & GF(A)
\end{tikzcd}
$$

while chasing $f : FA \to B$ clockwise, we have

$$
\begin{aligned}
Hom_{\mathbf{A}}(g^{op}, G(h)) \circ (f)^{\bullet}_{\langle A,B\rangle} &= Hom_{\mathbf{A}}(g^{op}, G(h))(G(f) \circ \eta_A) \\
&\quad \text{(by definition of } (-)^{\bullet}) \\
&= G(h) \circ G(f) \circ \eta_A \circ g \\
&\quad \text{(by definition of } Hom_{\mathbf{A}}(g^{op}, G(h)))
\end{aligned}
$$

This shows that $(-)^{\bullet}$ is natural at each component $\langle A, B\rangle$ of $\mathbf{A^{op}} \times \mathbf{B}$. Since the family $(-)^{\bullet}_{\langle A,B\rangle}$ is also a family of bijective functions (and hence a family of isomorphisms in **Set**) it follows from 6.2.3.3 that $(-)^{\sharp}$ is also a natural transformation.

"$\Leftarrow$" Assume we have a natural isomorphism $(-)^\bullet : Hom_{\mathbf{B}}(F-,-) \dot{\to} Hom_{\mathbf{A}}(-,G-)$. To be shown it is that, given $A \in \mathbf{A}, B \in \mathbf{B}$, the following diagram commutes for a unique $g : FA \to B \in Hom_{\mathbf{B}}(FA, B)$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\eta_A"'] & GB & & B \\ GFA \arrow[ur, "G(g)"'] & & & FA \arrow[u, "g"'] \end{tikzcd}$$

Let $A \in \mathbf{A}, FA \in \mathbf{B}$. Then by assumption we have that $Hom_{\mathbf{B}}(FA, FA) \cong Hom_{\mathbf{A}}(A, GFA)$. Let now $\eta_A = (id_{FA})^\bullet_{\langle A,FA\rangle}$. We first show that η_A is the component at A of a natural transformation $\eta : I_{\mathbf{A}} \dot{\to} G \circ F$. Therefore, we must show that for each $h : A' \to A \in \mathbf{A}$, the following diagram commutes:

$$\begin{tikzcd} A' \arrow[r, "\eta_{A'}"] \arrow[d, "h"'] & GFA' \arrow[d, "GF(h)"] \\ A \arrow[r, "\eta_A"'] & GFA \end{tikzcd}$$

To see this, note that

$$\begin{array}{rcllr} GF(h) \circ \eta_{A'} & = & GF(h) \circ (id_{FA'})^\bullet & & \text{(by definition of } \eta_{A'}) \\ & = & (F(h) \circ id_{FA'})^\bullet & & \text{(naturality of } (-)^\bullet \text{ in (4) below)} \\ & = & (id_{FA} \circ F(h))^\bullet & & \text{(identity)} \\ & = & (id_{FA})^\bullet \circ h & & \text{(naturality of } (-)^\bullet \text{ in (5) below)} \\ & = & \eta_A \circ h & & \text{(definition of } \eta_A) \end{array}$$

$$\begin{tikzcd} Hom_{\mathbf{B}}(FA', FA') \arrow[rrr, "(-)^\bullet_{\langle A',FA'\rangle}"] \arrow[ddd, "Hom_{\mathbf{B}}(F(id^{op}_{A'}){,}F(h))"'] & & & Hom_{\mathbf{A}}(A', GFA') \arrow[ddd, "Hom_{\mathbf{A}}(id^{op}_{A'}{,}GF(h))"] \\ & id_{FA'} \arrow[r, mapsto] \arrow[d, mapsto] \arrow[dr, phantom, "(4)"] & \eta_{A'} = (id_{FA'})^\bullet \arrow[d, mapsto] & \\ & F(h) \circ id_{FA'} \arrow[r, mapsto] & (F(h) \circ id_{FA'})^\bullet = GF(h) \circ \eta_{A'} & \\ Hom_{\mathbf{B}}(FA', FA) \arrow[rrr, "(-)^\bullet_{A',A}"'] & & & Hom_{\mathbf{A}}(A', GFA) \end{tikzcd}$$

$$
\begin{tikzcd}
Hom_{\mathbf{B}}(FA,FA) \arrow[rrr, "(-)^{\bullet}_{\langle A,FA\rangle}"] \arrow[ddd, "Hom_{\mathbf{B}}(F(h^{op}),id_{DA}"'] & & & Hom_{\mathbf{A}}(A,GFA) \arrow[ddd, "Hom_{\mathbf{A}}(h,G(id_{FA}))"] \\
& id_{FA} \arrow[r, mapsto] \arrow[d] \arrow[dr, phantom, "(5)"] & \eta_A = (id_{FA})^{\bullet} \arrow[d] & \\
& id_{FA} \circ F(h) \arrow[r, mapsto] & (id_{FA} \circ F(h))^{\bullet} = id_{GFA} \circ \eta_A \circ h & \\
Hom_{\mathbf{B}}(FA',FA) \arrow[rrr, "(-)^{\bullet}_{A',FA}"'] & & & Hom_{\mathbf{A}}(A',GFA)
\end{tikzcd}
$$

This shows that η_A is the component at A of a natural transformation $\eta : I \dot{\rightarrow} G \circ F$. It remains to show existence and uniqueness of $g : FA \to B$.

To existence, let us chase id_{FA} in the following diagram:

$$
\begin{tikzcd}
\langle A, FA\rangle \arrow[d, "\langle id_A^{op},g\rangle"'] & Hom_{\mathbf{B}}(FA,FA) \arrow[r, "(-)^{\bullet}_{\langle A,FA\rangle}"] \arrow[d, "Hom_{\mathbf{B}}(F(id_A^{op}),g)"'] & Hom_{\mathbf{A}}(A,GFA) \arrow[d, "Hom_{\mathbf{B}}(id_A^{op},G(g))"] \\
\langle A, B\rangle & Hom_{\mathbf{B}}(FA,B) \arrow[r, "(-)^{\bullet}_{\langle A,B\rangle}"'] & Hom_{\mathbf{A}}(A,GB)
\end{tikzcd}
$$

Chasing counterclockwise, we have that

$$
\begin{aligned}
(-)^{\bullet}_{\langle A,B\rangle} \circ Hom_B(F(id_A^{op}),g)(id_{FA}) &= (-)^{\bullet}_{\langle A,B\rangle}(g \circ id_{FA} \circ id_{FA}) \\
&\text{(definition of } Hom_B(F(id_A^{op}),g)) \\
&= (g)^{\bullet}_{\langle A,B\rangle} \\
&\text{(identity)}
\end{aligned}
$$

Chasing clockwise, we have that

$$
\begin{aligned}
Hom_{\mathbf{A}}(id_A^{op},G(g)) \circ (id_{FA})^{\bullet}_{\langle A,FA\rangle} &= Hom_{\mathbf{A}}(id_A^{op},G(g))(\eta_A) \\
&\text{(by definition of } \eta_A) \\
&= G(g) \circ \eta_A \circ id_A \\
&\text{(by definition of } Hom_{\mathbf{A}}(id_A^{op},G(g))) \\
&= G(g) \circ \eta_A \\
&\text{(identity)}
\end{aligned}
$$

Thus we have that $(g)^{\bullet} = G(g) \circ \eta_A$. Now, let f be the morphism in $Hom_{\mathbf{A}}(A,GB)$ corresponding to the bijection $(-)^{\bullet}_{\langle A,B\rangle} : Hom_{\mathbf{B}}(FA,B) \to Hom_{\mathbf{A}}(A,GB)$. This gives our desired equality $f = G(g) \circ \eta_A$.

It remains to show uniqueness of $g : FA \to B$. Therefore, suppose that there exists another $h : FA \to B$, such that $f = G(h) \circ \eta_A$. Now we have:

$$\begin{array}{rcll}
(h)^{\bullet}_{\langle A,B\rangle} & = & G(h) \circ \eta_A & \text{(by definition of } (-)^{\bullet}) \\
& = & f & \text{(assumption)} \\
& = & G(g) \circ \eta_A & \text{(by definition)} \\
& = & (g)^{\bullet}_{\langle A,B\rangle} & \text{(previous chasing)}
\end{array}$$

Since $(-)_{\langle A,B\rangle}$ is a bijective function in **Set** it is especially injective, and thus $(g)^{\bullet}_{\langle A,B\rangle} = (h)^{\bullet}_{\langle A,B\rangle} \Rightarrow g = h$, as required. □

Example 7.3.12 Let $\mathbf{C}$ be a category. Consider the product category $\mathbf{C} \times \mathbf{C}$ and the diagonal functor $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$. Let us see what a right adjoint to this functor is. Assuming there is a right adjoint $\times$ (see 6.1.1.19) to Δ, there should be a morphism we call

$$\langle \pi_A, \pi_B \rangle : \Delta \circ \times \langle A, B\rangle \to \langle A, B\rangle = \langle A \times B, A \times B\rangle \to \langle A, B\rangle$$

(this is the counit of the adjunction) with the following couniversal property.

For any object C of $\mathbf{C}$ and any morphism $\langle f, g\rangle : \Delta(C) \to \langle A, B\rangle = \langle C, C\rangle \to \langle A, B\rangle$ there is a unique morphism $\langle f, g\rangle^{\bullet} : C \to A \times B$ such that the following diagram commutes, i.e., such that $\langle \pi_A, \pi_B\rangle \circ \Delta(\langle f, g\rangle^{\bullet}) = \langle f, g\rangle$:

$$\begin{tikzcd}
\langle C, C\rangle \arrow[r, "\Delta(\langle f,g\rangle^{\bullet})"] \arrow[dr, "\langle f,g\rangle"'] & \langle A \times B, A \times B\rangle \arrow[d, "\langle \pi_A, \pi_B\rangle"] & C \arrow[d, "\langle f,g\rangle^{\bullet}"] \\
& \langle A, B\rangle & A \times B
\end{tikzcd}$$

If we separate the pairs into components, we have that given a pair of $\mathbf{C}$-morphisms $\pi_A : A \times B \to A$ and $\pi_B : A \times B \to B$ there is a unique morphism $\langle f, g\rangle^{\bullet} : C \to A \times B$ such that $\pi_A \circ \langle f, g\rangle^{\bullet} = f$ and $\pi_B \circ \langle f, g\rangle^{\bullet} = g$. This is exactly the couniversal property of products, i.e., $A \times B$ with projections π_A and π_B. Thus, a right adjoint to Δ is just a functor $\times : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ that chooses a product for each pair of objects of $\mathbf{C}$.

The unity of this adjunction is represented by the morphism $\langle id_C, id_C\rangle : C \to \times\Delta C$ in the following diagram:

$$
\begin{tikzcd}
C \arrow[r, "\langle id_C, id_C \rangle"] \arrow[dr, "\langle f,g \rangle"'] & \times\Delta(C) = C \times C \arrow[d, "f \times g"] & \langle C, C \rangle \arrow[d, "\langle f,g \rangle"] \\
 & \times\langle A, B \rangle = A \times B & \langle A, B \rangle
\end{tikzcd}
$$

Example 7.3.13 Consider now the functor $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$. If $+ : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ is a left adjoint to Δ then there should be a morphism

$$\langle in_A, in_B \rangle : \langle A, B \rangle \to \Delta + (\langle A, B \rangle) = \langle +\langle A, B \rangle, +\langle A, B \rangle \rangle$$

(this is the unit of the adjunction) such that for any C in $\mathbf{C}$ and any morphism $\langle f, g \rangle : \langle A, B \rangle \to \Delta(C) = \langle C, C \rangle$ there is a unique morphism $\langle f, g \rangle^\sharp : +\langle A, B \rangle \to C$ such that $\Delta(\langle f, g \rangle^\sharp) \circ \langle in_A, in_B \rangle = \langle \langle f, g \rangle^\sharp, \langle f, g \rangle^\sharp \rangle \circ \langle in_A, in_B \rangle = \langle f, g \rangle$. If we separate the pairs into components and write $A + B$ instead of $+\langle A, B \rangle$ we recover the categorical definition of the coproduct. Thus a left adjoint to Δ, if it exists, is just a functor $+ : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ that chooses a coproduct for each pair of objects of $\mathbf{C}$, i.e, an object $A + B$ with injections in_A and in_B.

Proposition 7.3.14 *Left adjoints preserve colimits.*

Proof: Let $F : \mathbf{C} \to \mathbf{D}$ be a free functor with respect to $G : \mathbf{D} \to \mathbf{C}$ with unit $\eta : I_{\mathbf{C}} \dot{\to} G \circ F$, $D : \mathbf{S} \to \mathbf{C}$ a diagram (see 6.1.5.1) with colimit $(\Delta(C), \tau : D \dot{\to} \Delta(C))$ (where $\Delta : \mathbf{C} \to \mathbf{C}^{\mathbf{S}}$), and let $F \circ D : S \to D$ be the translated diagram in $\mathbf{D}$:

$$
\begin{tikzcd}
I \arrow[dd, "e"] & D(I) \arrow[dd, "D(e)"'] \arrow[dr, "\tau_I"] & & F(D(I)) \arrow[dd, "F(D(e))"'] \arrow[dr, "F(\tau_I)"] & \\
 & & \Delta(C) & & F(\Delta(C)) \\
J & D(J) \arrow[ur, "\tau_J"'] & & F(D(J)) \arrow[ur, "F(\tau_J)"'] &
\end{tikzcd}
$$

Now let $(\Delta(A), \theta : F \circ D \to \Delta(A))$ be compatible with $F \circ D$, i.e., $\Delta(A)$ is another candidate for a colimit of $F \circ D : \mathbf{S} \to \mathbf{C}$. This means that the outer triangle (which we will call the non-limiting cocone) in the following diagram should commute for each $e : I \to J \in \mathbf{S}$.

$$\begin{tikzcd}
F(D(I)) \arrow[dd, "F(D(e))"'] \arrow[dr, "F(\tau_I)"] \arrow[drr, "\theta_I"] & & \\
& F(\Delta(C)) \arrow[r, "?"] & \Delta(A) \\
F(D(J)) \arrow[ur, "F(\tau_J)"] \arrow[urr, "\theta_J"'] & &
\end{tikzcd}$$

Now, after applying the functor $G : \mathbf{F} \to \mathbf{C}$ in the non-limiting cocone, we are able to obtain the following digram:

$$\begin{tikzcd}
D(I) \arrow[r, "\eta_{D_I}"] \arrow[dd, "D(e)"'] \arrow[ddr, phantom, "(1)"] & G(F(D(I))) \arrow[dr, "G(\theta_I)"] \arrow[dd, "G(F(D(e)))"] \arrow[dr, phantom, "(2)"'] & \\
& & G(\Delta(A)) \\
D(J) \arrow[r, "\eta_{D_J}"'] & G(F(D(J))) \arrow[ur, "G(\theta_J)"'] &
\end{tikzcd}$$

Now, observe that (1) commutes by naturality of η and (2) commutes because functors (in this case G) preserve commutative diagrams. Therefore, $\sigma : D \dot{\to} G \circ \Delta(A) =_{def} G(\theta) \circ \eta$ is another cocone for $D : \mathbf{S} \to \mathbf{C}$ in $\mathbf{C}$, i.e., $(G(\Delta(A)), \sigma : D \dot{\to} G \circ \Delta(A))$ is another candidate for a colimit of D in $\mathbf{C}$, according to the following diagram:

$$\begin{tikzcd}
D(I) \arrow[dd, "D(e)"'] \arrow[dr, "\tau_I"] \arrow[drr, "\sigma_I"] & & \\
& \Delta(C) \arrow[r, "f"] & G(\Delta(A)) \\
D(J) \arrow[ur, "\tau_J"] \arrow[urr, "\sigma_J"'] & &
\end{tikzcd}$$

However, by assumption, we know that the inner triangle is a colimit-cone, and therefore we know that there exists a unique $f : C \to G(\Delta(A))$ such that $f \circ \tau_I = \sigma_I \wedge f \circ \tau_J = \sigma_J$ for each $e : I \to J \in \mathbf{S}$.

Now, since $F(\Delta(C))$ is a free construction, we have a unique $g : F(\Delta(C)) \to \Delta(A)$ such that the following diagram commutes:

$$\begin{tikzcd}
\Delta(C) \arrow[r, "f"] \arrow[d, "\eta_{\Delta(C)}"'] & G(\Delta(A)) \\
G(F(\Delta(C))) \arrow[ur, "G(g)"'] &
\end{tikzcd}$$

Now consider the following diagram:

$$
\begin{tikzcd}
D(I) \arrow[r, "\tau_I"] \arrow[d, "\eta_{D(I)}"'] \arrow[dr, phantom, "(3)"] & \Delta(C) \arrow[d, "\eta_{\Delta(C)}"'] \arrow[r, "f"] \arrow[dr, phantom, "(4)"] & G(\Delta(A)) \\
G(F(D(I))) \arrow[r, "G(F(\tau_I))"'] & G(F(\Delta(C)))) \arrow[ur, "G(g)"'] &
\end{tikzcd}
$$

Note that (3) commutes by naturality of η, (4) is commutative by the previous diagram, and therefore (3)+(4) is commutative.

Now we get to the crucial part of our proof, i.e., we have to show that the following diagram commutes for a unique $g : F(\Delta(C)) \to \Delta(A)$.

$$
\begin{tikzcd}
F(D(I)) \arrow[r, "\theta_I"] \arrow[d, "F(\tau_I)"'] & \Delta(A) \\
F(\Delta(C) \arrow[ur, "g"'] &
\end{tikzcd}
$$

Thus, consider the following diagram:

$$
\begin{tikzcd}
D(I) \arrow[r, "\tau_I"] \arrow[d, "\eta_{D(I)}"'] \arrow[rr, bend left, "\sigma_I"] & \Delta(C) \arrow[r, "f"] & G(\Delta(A)) \\
G(F(D(I))) \arrow[urr, "G(?)"'] & &
\end{tikzcd}
$$

Since $F(D(I))$ is a free construction, we know that there is a unique morphism ? : $F(D(I)) \to \Delta(A)$ such that the above diagram commutes. We put ? $=_{def}$ $g \circ F(\tau_I)$ and we have:

$$
\begin{array}{rcll}
G(g \circ F(\tau_I)) \circ \eta_{D(I)} & = & G(g) \circ G(F(\tau_I)) \circ \eta_{D(I)} & (G \text{ is a functor}) \\
& = & f \circ \tau_I & (\text{by } (3)+(4))
\end{array}
$$

Now, if we set ? $=_{def}$ θ_I we have:

$$
\begin{array}{rcll}
G(\theta_I) \circ \eta_{D(I)} & = & \sigma_I & (\text{by definition of } \sigma) \\
& = & f \circ \tau_I & (C \text{ is a colimit of } \mathbf{D})
\end{array}
$$

This shows that both θ_I and $g \circ F(\tau_I)$ are solutions to the equation

$$G(?) \circ \eta_{D(I)} = f \circ \tau_I$$

Since $F(D(I))$ is a free construction, we must have $g \circ F(\tau_I) = \theta_I$

It remains to show uniqueness of $g : F(\Delta(C)) \to \Delta(A)$. Therefore, let $g' : F(\Delta(C)) \to \Delta(A)$ be such that the following diagram commutes:

$$
\begin{tikzcd}
F(D(I)) \arrow[r, "\theta_I"] \arrow[d, "F(\tau_I)"'] & \Delta(A) \\
F(\Delta(C) \arrow[ur, "G(g')"'] &
\end{tikzcd}
$$

Let $f' = G(g') \circ \eta_{\Delta(C)}$, i.e., such that the following diagram commutes:

$$
\begin{tikzcd}
\Delta(C) \arrow[r, "f'"] \arrow[d, "\eta_{\Delta(C)}"'] & G(\Delta(A)) \\
G(F(\Delta(C)) \arrow[ur, "g'"'] &
\end{tikzcd}
$$

Now, observe that

$$
\begin{array}{rcll}
f' \circ \tau_I & = & G(g') \circ \eta_{\Delta(C)} \circ \tau_I & \text{(by the above diagram)} \\
& = & G(g') \circ G(F(\tau_I)) \circ \eta_{D(I)} & \text{(by (3))} \\
& = & G(g' \circ F(\tau_I)) \circ \eta_{D(I)} & (G \text{ is a functor}) \\
& = & G(\theta_I) \circ \eta_{D(I)} & \text{(assumption)} \\
& = & f \circ \tau_I & \text{(previous deduction)} \\
& = & \sigma_I & ((\Delta(C), \tau) \text{ is a colimit})
\end{array}
$$

Since $(\Delta(C), \tau)$ is a colimit and we have $f \circ \tau_I = \sigma_I = f' \circ \tau_I$, we must have, by uniqueness, that $f = f'$. Now we have $f = f' = G(g') \circ \eta_{\Delta(C)}$. However, $F(\Delta(C))$ is a free construction, and therefore, by uniqueness, $g = g'$, as required. $\square$

Proposition 7.3.15 *Right adjoints preserve limits.*

Proof: $\boxed{\text{D}}$

7.4 Bibliographic notes

Free constructions and adjoint functors are treated in any textbook on category theory. Free constructions are carefully introduced in [Wal91], [AL91] and [BW90]. Theorem 7.3.1 is adapted from [EM85]. Detailed and comprehensible presentations of adjoint functors can be found in [BW90] and [Gol86].

Bibliography

[AHS90] Adamek, Jiri; Herrlich, Horst; Strecker, Gorge E. **Abstract and Concrete Categories:** The Joy of Cats. New York: John Wiley & Sons, 1990. 482p.

[AL91] Asperti, Andrea; Longo, Giuseppe. **Categories, Types, and Structures:** An Introduction to Category Theory for the Working Computer Scientist. Cambridge: Mit Press, 1991. 306p.

[AM75] Arbib, Michael; Manes, Ernest G. **Arrows, Structures, and Functors:** The Categorical Imperative. New York: Academic Press, 1975. 185p.

[AM86] Arbib, Michael A; Manes, Ernest G. **Algebraic Approaches to Program Semantics**. New York: Springer-Verlag, 1986. 351p.

[Bac81] Backus, J., The algebra of functional programs: function level reasoning, linear equations, and extended definitions. In: International Colloquium on Formalization of Program Concepts, 4., 1981, Peniscola. **Lecture Notes in Computer Science**, v.107. Berlin: Springer-Verlag, 1981, 478p. p.1-43.

[Bur86] Burmeister, P. **A Model theoretic Oriented Approach to Partial Algebras**, volume 32 of **Mathematical Research - Mathematische Forschung**. Akademie-Verlag, Berlin, 1986.

[EK76a] Ehrig, Hartmut; Kreowski, H. -J. Parallelism of manipulations in multidimensional information structures. In: Symposium on Mathematical Foundations of Computer Science, 9., 1976, Gdańsk. **Lecture Notes in Computer Science**, v.45. Berlin: Springer-Verlag, 1976, 606p. p. 284-293.

[EK76b] Ehrig, H.; Kreowski, H. -J. **Contributions to the Algebraic Theory of Graph Grammars**. Berlin: TU Berlin, FB Informatik, TR 76-22, 1976. 38p.

[EM45] Eilenberg, S.; MaC Lane, S. General Theory of natural equivalences. **Trans. Am. Math. Society**, v.58, p.231-294, 1945.

[EM85] Ehrig, Hartmut; Mahr; Bernd. **Fundamentals of Algebraic Specification I**: EATCS Monographs on Theoretical Computer Science, v.6. Berlin: Springer-Verlag, 1985. 321p.

[EP72] Ehrig, Hartmut; PFENDER, Michael. **Kategorien und Automaten**. Berlin: Walter de Gruyter, 1972. 170p.

[Gol86] Goldblatt, Robert. **TOPOI: The Categorial Analysis of Logic**: Studies in logic and the foundations of mathematics. New York: North-Holland, v.98, 1983. 551p.

[Gun92] Gunter, Carl A. **Semantics of Programming Languages:** Structures and Techniques. Cambridge: Mit Press, 1992. 419p.

[HS73] Herrlich H.; Strecker G. E. **Category Theory**. Boston: Allyn and Bacon, 1973. 400p.

[LS86] Lambek, J.; Scott, P. J. **Introduction to Higher Order Categorical Logic**. Cambridge: Cambridge University Press, 1986. 293p.

[MaC71] MaC Lane, Saunders. **Categories for the Working Mathematician**. New York: Springer-Verlag, 1971. 262p.

[Pad91] Padberg, Julia: **Kolimeskonstruktionen in Algebraischen Spezifikationssprachen**. Studient Arbeit, Technical University of Berlin,1991.

[Pie90] Pierce, Benjamin. **A Taste of Category Theory for Computer Scientists**. Pittsburgh: Carnegie Mellon University ,CMU-CS-90-113, 1990. 86p.

[Poi92] Poigne, A. Basic Category Theory. In: **Handbook of Logic in Computer Science**. Background: Mathematical Structures. Oxford: Clarendom Press, 1992, v.1, p.413-640.

[Wal91] Walters, R. F. C. **Categories and Computer Science**. Cambridge: Cambridge University Press, 1991. 166p.