Lecture Notes on Category Theory

Paul Ossientis

January 19, 2020

Contents

Chapter 1

Category

1.1 Small category

Before we define a category in full generality, we shall focus our attention on the notion of *small category*. This notion is interesting to us because while it essentially describes the notion of *category* itself, it remains simple enough to be compared with various other algebraic structures. For example, consider the case of a monoid: a monoid is essentially a set M together with a binary relation $\circ$ defined on M which is associative, and an element e of M which acts as an identity element for $\circ$. In short a monoid is a tuple $(M, \circ, e)$ containing some data, and which satisfy certain properties. The same is true of a *small category*: it is also a tuple containing some data, and which satisfy certain properties:

Definition 1 *We call* small category *any tuple* $(\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ *with:*

(1)	Ob *is a set*
(2)	Arr *is a set*
(3)	$\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$ *is a function*
(4)	$\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$ *is a function*
(5)	$\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ *is a function*
(6)	$\circ : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ *is a partial function*
(7)	$g \circ f$ *is defined* $\Leftrightarrow\ \mathrm{cod}(f) = \mathrm{dom}(g)$
(8)	$\mathrm{cod}(f) = \mathrm{dom}(g) \ \Rightarrow\ \mathrm{dom}(g \circ f) = \mathrm{dom}(f)$
(9)	$\mathrm{cod}(f) = \mathrm{dom}(g) \ \Rightarrow\ \mathrm{cod}(g \circ f) = \mathrm{cod}(g)$
(10)	$\mathrm{cod}(f) = \mathrm{dom}(g) \wedge \mathrm{cod}(g) = \mathrm{dom}(h) \ \Rightarrow\ (h \circ g) \circ f = h \circ (g \circ f)$
(11)	$\mathrm{dom}\,(\,\mathrm{id}(a)\,) = a = \mathrm{cod}\,(\,\mathrm{id}(a)\,)$
(12)	$\mathrm{dom}(f) = a \ \Rightarrow\ f \circ \mathrm{id}(a) = f$
(13)	$\mathrm{cod}(f) = a \ \Rightarrow\ \mathrm{id}(a) \circ f = f$

where $(7) - (13)$ *hold for all* $f, g, h \in \mathrm{Arr}$ *and* $a \in \mathrm{Ob}$*:*

So if $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ is a small category, we have two sets Ob and Arr together with some structure defined on these sets. This feels very much like a monoid, except that we have two sets instead of one and it all looks more complicated. The set Ob is called the *set of objects* of the small category $\mathcal{C}$ and is denoted $\mathrm{Ob}\ \mathcal{C}$, while the set Arr is called the *set of arrows* of the small category $\mathcal{C}$ and is denoted $\mathrm{Arr}\ \mathcal{C}$. An element $x \in \mathrm{Ob}\ \mathcal{C}$ is called an *object* of $\mathcal{C}$, while an element $f \in \mathrm{Arr}\ \mathcal{C}$ is called an *arrow* of $\mathcal{C}$.

As part of the structure defined on the small category $\mathcal{C}$, we have two functions $\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$ and $\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$. Hence, given an arrow f of the small category $\mathcal{C}$, we have two objects $\mathrm{dom}(f)$ and $\mathrm{cod}(f)$ of the small category $\mathcal{C}$. The object $\mathrm{dom}(f)$ is called the *domain* of f. The object $\mathrm{cod}(f)$ is called the *codomain* of f. Note that an arrow f of the small category $\mathcal{C}$ is simply an element of the set $\mathrm{Arr}\ \mathcal{C}$. So it is itself a set but it may not be a function. The words *domain* and *codomain* are therefore overloaded as we are using them in relation to a set f which is possibly not a function. Whenever f is an arrow of the small category $\mathcal{C}$ and a, b are objets, it is common to use the notation $f : a \to b$ as a notational shortcut for the equations $\mathrm{dom}(f) = a$ and $\mathrm{cod}(f) = b$. Once again, it is important to guard against the possible confusion induced by the notation $f : a \to b$ which does not mean that f is function. It simply means that f is an arrow with domain a and codomain b in the small category $\mathcal{C}$.

One of the main ingredients of the structure defining a small category $\mathcal{C}$ is the partial function $\circ : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$, called the *composition operator* in the small category $\mathcal{C}$. The domain of this partial function is made of all ordered pairs (g, f) of arrows in $\mathcal{C}$ for which $\mathrm{cod}(f) = \mathrm{dom}(g)$. As already indicated in definition (1), we use the infix notation $g \circ f$ rather than $\circ(g, f)$ and the arrow $g \circ f$ is called the *composition* of g and f. Once again, we should remember that the notation $g \circ f$ does not mean that g or f are functions. They are simply arrows in the small category $\mathcal{C}$. One key property of the composition operator $\circ$ is the associativity postulated by (10) of definition (1). Note that if $f : a \to b$ and $g : b \to c$, then from properties (8) and (9) of definition (1) we obtain $g \circ f : a \to c$. Furthermore, if $h : c \to d$ we have $h \circ g : b \to d$ and the arrows $(h \circ g) \circ f$ and $h \circ (g \circ f)$ are therefore well-defined arrows with domain a and codomain d. This shows that the expression involved in the associativity condition (10) of definition (1) is always meaningful, involving terms which are well-defined provided $g \circ f$ and $h \circ g$ are themselves well-defined, i.e. provided $\mathrm{cod}(f) = \mathrm{dom}(g)$ and $\mathrm{cod}(g) = \mathrm{dom}(h)$.

Finally, as part of the structure defining the small category $\mathcal{C}$, we have a function $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ called the *identity operator* on the small category $\mathcal{C}$. Hence, for every object a of $\mathcal{C}$ we have an arrow $\mathrm{id}(a)$ called the *identity at a*. Looking at property (11) of definition (1) we see that $\mathrm{id}(a) : a \to a$. In other words, the arrow $\mathrm{id}(a)$ has domain a and codomain a. Furthermore, looking at properties (12) and (13) of definition (1), for every arrow $f : a \to b$, the composition arrows $\mathrm{id}(b) \circ f$ and $f \circ \mathrm{id}(a)$ are well-defined and both equal to f.

1.2 Category

The notion of *small category* defined in definition (1) is similar to that of any other algebraic structure the reader may be familiar with. It can safely be encoded in set theory as a tuple (which is a set) containing data (which are other sets) which satisfies certain properties. In set theory, everything is a set. A small category $\mathcal{C}$ is a set, its collection of objects $\mathrm{Ob}\ \mathcal{C}$ is a set, its arrows $\mathrm{Arr}\ \mathcal{C}$ form a set, the functions dom, cod, id and the composition operator $\circ$ are all sets (functions are typically encoded as sets of ordered pairs).

Category theory falls outside of set theory. While the definition of a *category* we provide below is formally identical to that of a small category, the object we are defining can no longer be encoded in general as an object of set theory. For example, say we want to speak about the *universe of all sets* or the *universe of all monoids*. These *universes* which are known as *classes* cannot be represented as sets. They are not objects of set theory. Or say we are working within the formal framework of a proof assistant such as Coq, Agda or Lean. These tools are based on type theory and do not fall within the scope of set theory. When defining a *category*, we assume some form of meta-theoretic context, some form of logic, some way of reasoning about objects which may not be sets, where some meaning is attached to the words *tuple*, *collection*, *equality* and *map*. This may sound all very fuzzy, yet we cannot be more formal at this stage.

Definition 2 *We call* category *any tuple* $(\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ *such that:*

(1) Ob *is a collection with equality*

(2) Arr *is a collection with equality*

(3) $\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$ *is a map*

(4) $\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$ *is a map*

(5) $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ *is a map*

(6) $\circ : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ *is a partial map*

(7) $g \circ f$ *is defined* $\Leftrightarrow\ \mathrm{cod}(f) = \mathrm{dom}(g)$

(8) $\mathrm{cod}(f) = \mathrm{dom}(g)\ \Rightarrow\ \mathrm{dom}(g \circ f) = \mathrm{dom}(f)$

(9) $\mathrm{cod}(f) = \mathrm{dom}(g)\ \Rightarrow\ \mathrm{cod}(g \circ f) = \mathrm{cod}(g)$

(10) $\mathrm{cod}(f) = \mathrm{dom}(g) \wedge \mathrm{cod}(g) = \mathrm{dom}(h)\ \Rightarrow\ (h \circ g) \circ f = h \circ (g \circ f)$

(11) $\mathrm{dom}\,(\,\mathrm{id}(a)\,) = a = \mathrm{cod}\,(\,\mathrm{id}(a)\,)$

(12) $\mathrm{dom}(f) = a\ \Rightarrow\ f \circ \mathrm{id}(a) = f$

(13) $\mathrm{cod}(f) = a\ \Rightarrow\ \mathrm{id}(a) \circ f = f$

where $(7) - (13)$ *hold for all* $f, g, h \in \mathrm{Arr}$ *and* $a \in \mathrm{Ob}$*:*

So let $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ be a category: then $\mathcal{C}$ is a *tuple* but it is no longer a tuple in a set-theoretic sense. We assume given some logical framework where the notion of *tuple* is clear, even if not formally defined. Furthermore, We are no longer imposing that Ob should be a set, but are instead using the phrase *collection with equality*, whatever this may mean in our given logical context. So we shall still make use of the notation Ob $\mathcal{C}$ but this will now refer to the *collection* of all *objects* of the category $\mathcal{C}$. In fact, if a is an object of the category $\mathcal{C}$, we shall abuse notations somewhat by writing '$a \in \mathrm{Ob}\ \mathcal{C}$' or even simply '$a \in \mathcal{C}$' to express the fact that a is an object of $\mathcal{C}$, being understood that this use of the set membership symbol '$\in$' does not mean anything is a set. Since we are stepping out of set theory, the objects of the category $\mathcal{C}$ may not be sets themselves. They are simply members of the *collection* Ob $\mathcal{C}$. However, properties (7) $-$ (13) of definition (2) are all referring to equalities between objects such that $\mathrm{cod}(f) = \mathrm{dom}(g)$. So it must be the case that the notion of *equality* be meaningful on the collection Ob $\mathcal{C}$. This explains our use of the phrase *collection with equality*: given $a, b \in \mathcal{C}$, the statement $a = b$ while not a set-theoretic equality is nonetheless assumed to be defined.

Similarly, the *collection* of *arrows* of the category $\mathcal{C}$ shall still be denoted Arr $\mathcal{C}$, but is no longer required to be a set. If f is an arrow of the category $\mathcal{C}$ then f itself may not be a set and we may still write '$f \in \mathrm{Arr}\ \mathcal{C}$' simply to indicate that f is a *member* of the *collection* Arr $\mathcal{C}$. Properties (10), (12) and (13) of definition (2) are all referring to equalities between arrows so the *collection* Arr $\mathcal{C}$ must have some notion of *equality* defined on it.

Since Ob and Arr are no longer sets in general, the *maps* $\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$, $\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$, $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ and the partial map $\circ : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ cannot possibly be *functions* in the set-theoretic sense. So there must be some meaning to the word *map* (from one *collection* to another) in whatever logical framework we are working in. The *collection* $\mathrm{Arr} \times \mathrm{Arr}$ is not a set, and is simply the *collection* of all 2-dimensional *tuples* made from Arr. Our using the word *map* rather than *function* in definition (2) is simply an attempt at reminding ourselves of the fact these are not set-theoretic functions, eventhough the words *map* and *function* are perfectly interchangeable in standard (set-theoretic) mathematics.

Given $f \in \mathrm{Arr}\ \mathcal{C}$, we shall still call the object $\mathrm{dom}(f)$ the *domain* of f and the object $\mathrm{cod}(f)$ the *codomain* of f. Given $a, b \in \mathcal{C}$, we shall still use the notation $f : a \to b$ as a notational shortcut for $\mathrm{dom}(f) = a$ and $\mathrm{cod}(f) = b$. The partial map $\circ$ is still the *composition operator* and the arrow $g \circ f$ shall still be called the *composition* of g and f, provided it is defined. The map $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ is still the *identity operator* on the category $\mathcal{C}$, and for all $a \in \mathcal{C}$, the arrow $id(a) : a \to a$ is known as the *identity at a*. For all arrows $f : a \to b$, it is still the case that the arrows $\mathrm{id}(b) \circ f$ and $f \circ \mathrm{id}(a)$ are well-defined and both equal to f. Just as in the case of a small category, whenever $f : a \to b$, $g : b \to c$ and $h : c \to d$, all the terms involved in the associativity condition $(h \circ g) \circ f = h \circ (g \circ f)$ of definition (2) are well defined.

1.3 Equality of categories

Whichever logical framework we are working from, we saw that when defining a category $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$, some notion of equality had to be defined on the collections Ob and Arr. Now if $\mathcal{C}' = (\mathrm{Ob}', \mathrm{Arr}', \mathrm{dom}', \mathrm{cod}', \mathrm{id}', \circ')$ is another category, the question may arise as to whether $\mathcal{C} = \mathcal{C}'$. Or indeed, we may simply be asking whether the collections Ob and Ob' are the same, or whether $\mathrm{dom} = \mathrm{dom}'$ etc. It is very difficult for us to carry out any sort of formal reasoning on things without equality. So having equality defined on Ob and Arr is neccessary for definition (2) to even make sense, but it is not enough for us to formally prove anything about categories. Hence we shall assume:

Axiom 1 *A notion of equality exists for* collections*.*

It is implicit in the statement of axiom (1) that the notion of equality between *collections* should be reflexive, symmetric and transitive. Furthermore:

Axiom 2 *Two collections with identical members are equal.*

In particular, if $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ is a category and we have another partial map $\circ' : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ such that $g \circ' f$ is defined if and only if $g \circ f$ is defined, then axiom (2) allows us to argue that the domain of $\circ'$ is the same collection as the domain of $\circ$.

Axiom 3 *Let A be a collection and B be a collection with equality. Then two maps $F, G : A \to B$ are equal if and only if $F(x) = G(x)$ for all x in A.*

In particular, if $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ is a category and $\mathrm{dom}' : \mathrm{Arr} \to \mathrm{Ob}$ is another map such that $\mathrm{dom}'(x) = \mathrm{dom}(x)$ for every object $x \in \mathcal{C}$, then $\mathrm{dom}' = \mathrm{dom}$. Or if $\circ' : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ is another partial map with the same domain as that of $\circ$ and such that $g \circ' f = g \circ f$ when defined, then $\circ' = \circ$.

Axiom 4 *Two tuples with identical entries are equal.*

So if $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ and $\mathcal{C}' = (\mathrm{Ob}', \mathrm{Arr}', \mathrm{dom}', \mathrm{cod}', \mathrm{id}', \circ')$ are two catgeories such that $\mathrm{Ob} = \mathrm{Ob}'$, $\mathrm{Arr} = \mathrm{Arr}'$, $\mathrm{dom} = \mathrm{dom}'$, $\mathrm{cod} = \mathrm{cod}'$, $\mathrm{id} = \mathrm{id}'$ and $\circ = \circ'$ then we have the equality $\mathcal{C} = \mathcal{C}'$.

1.4 Category of sets

Definition 3 *We call* **Set** *the category* **Set** $= (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ *where*

$$\begin{array}{ll}
(1) & \mathrm{Ob} = \{\ x \mid x \textit{ is a set }\} \\
(2) & \mathrm{Arr} = \{\ (a,b,f) \mid f \textit{ is a function } f : a \to b\ \} \\
(3) & \mathrm{dom}\,(a,b,f) = a \\
(4) & \mathrm{cod}\,(a,b,f) = b \\
(5) & \mathrm{id}(a) = (a,a,i(a)) \\
(6) & (b,c,g) \circ (a,b,f) = (a,c,g \circ f)
\end{array}$$

where $(3) - (6)$ *hold for all sets* a, b, c *and functions* $f : a \to b$, $g : b \to c$, $i(a) : a \to a$ *denotes the usual identity function on* a, *and* $g \circ f$ *denotes the usual function composition defined by* $(g \circ f)(x) = g(f(x))$, *for all* $x \in a$.

The collection of objects of the category **Set** is defined to be the class of all sets. We are using the set comprehension notation $\{\ x \mid x \text{ is a set }\}$ to denote this class, but this is an abuse of notation as Ob is not a set but a proper class. One could think of a class as a precicate $P(x)$ of first order logic with one free variable. From this point of view Ob becomes the predicate $\mathrm{Ob}(x) = \top$, i.e. the predicate which returns true for all x. Every set satisfies the predicate Ob, so every set is a member of the class Ob. The class Ob is not a set because the set-theoretic statement $\exists y, \forall z, z \in y \ \Leftrightarrow\ \mathrm{Ob}(z)$ can be proven false. In other words, there exists no set y whose elements z are exactly the sets which satisfy the predicate Ob. There exists no set which contains all sets.

The collection of arrows of the category **Set** is defined to be the class of triples (a, b, f) where a, b are sets and f is a function $f : a \to b$. This last notation is a common set-theoretic shortcut to express the fact that f is a *function* with *domain* a and *range* **which is a subset of** b. A *function* is any set f whose elements are ordered pairs (x, y) and which is functional, i.e. for which the following implication holds for all sets x, y, y':

$$(x,y) \in f \ \wedge\ (x,y') \in f \ \Rightarrow\ y = y'$$

The *domain* of a function f is the set of all sets x for which there exists a set y with $(x, y) \in f$. The *range* of a function f is the set of all sets y for which there exists a set x with $(x, y) \in f$. If x belongs to the domain of a function f, the notation '$f(x)$' commonly refers to the unique set y with $(x, y) \in f$.

Now, as already pointed out the notation $f : a \to b$ only requires that the range of f should be a subset of b. There is no requirement that the range of f should be equal to b. So if $f : a \to b$ and $b \subseteq c$ then $f : a \to c$. This explains why the collection of arrows Arr is defined as a class of triples (a, b, f) rather than a class of functions f. Knowing the function f does not tell you which *codomain* it should have. Any set b which is a superset of its range is a possible codomain. So we keep the set b together with the function f in the triple (a, b, f) so as to remember which codomain is intended for this particular arrow of the category **Set**. Incidentally, we also keep the range a of the function f in the triple (a, b, f) but this is not necessary, as the knowledge of f does allow us to recover its domain a. However, the triple (a, b, f) is convenient, allowing us to treat *domain* and *codomain* uniformly. Once again, it should be remembered that the collection of arrows Arr is not a set but a proper class, corresponding to the predicate $\mathrm{Arr}(x)$ defined as follows:

$$\mathrm{Arr}(x) = \exists a\, \exists b\, \exists f\ ,\ x = (a, b, f)\ \wedge\ f : a \to b$$

The maps $\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$ and $\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$ for the category **Set** are defined respectively by $\mathrm{dom}(a, b, f) = a$ and $\mathrm{cod}(a, b, f) = b$. This looks simple enough, but for those who worry about foundational issues, we should just note that these are also proper classes which can be encoded as predicates. For example:

$$\mathrm{dom}(x) = \exists u\, \exists v\ ,\ x = (u, v)\ \wedge\ \mathrm{Arr}(u)\ \wedge\ (\ \exists a\, \exists b\, \exists f\ ,\ u = (a, b, f)\ \wedge\ v = a\ )$$

In other words, any set x satisfies the predicate $\mathrm{dom}(x)$ if and only if it is an ordered pair (u, v) where u satisfies the predicate $\mathrm{Arr}(u)$ and for which there exist sets a, b, f with $u = (a, b, f)$ and $v = a$. In short, (u, v) satisfies the predicate dom if and only if u is an arrow $u = (a, b, f)$ and $v = a$.

We defined the identity operator id by $\mathrm{id}(a) = (a, a, i(a))$ and the composition operator $\circ$ by $(b, c, g) \circ (a, b, f) = (a, c, g \circ f)$ where $g \circ f$ is the usual function composition and $i(a) : a \to a$ is the usual identity function. As before, these defined maps are not functional sets of ordered pairs but rather proper classes which we could also encode as precicates of first order logic. One important point to note is the fact that (6) of definition (3) only defines the composition arrow $(b, c, g) \circ (a, b, f)$ where $f : a \to b$ and $g : b \to c$. In other words, the composition $(d, c, g) \circ (a, b, f)$ with $f : a \to b$ and $g : d \to c$ is only defined when $b = d$. Furthermore, the usual function composition $g \circ f$ is a function $g \circ f : a \to c$ which from (2) of definition (3) means that the composed arrow $(b, c, g) \circ (a, b, f) = (a, c, g \circ f)$ is indeed a member of the collection Arr, and the partial map $\circ$ thus defined is indeed a partial map $\circ : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$.

Proposition 1 *The category* **Set** *of definition (3) is a category.*

Proof
Now that we have defined the data $(\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ of the category **Set**, it is time to check this data actually forms a category. We need to check that conditions $(7) - (13)$ of definition (2) are satisfied.

(7): suppose f^* and g^* are two members of the collection Arr. We need to check that $g^* \circ f^*$ is defined if and only if $\mathrm{cod}(f^*) = \mathrm{dom}(g^*)$. Using (2) of definition (3), f^* can be written $f^* = (a, b, f)$ for some function $f : a \to b$ and g^* can be written $g^* = (d, c, g)$ for some function $g : d \to c$. However, from (6) of definition (3), the arrrow $(d, c, g) \circ (a, b, f)$ is only defined in the case when $b = d$. Furthermore, from (4) of definition (3) we have $\mathrm{cod}(f^*) = b$ and from (3) of definition (3) we have $\mathrm{dom}(g^*) = d$. We conclude that $g^* \circ f^*$ is defined if and only if $\mathrm{cod}(f^*) = \mathrm{dom}(g^*)$ as required.

(8): Let $f^*, g^* \in \text{Arr}$ such that $\text{cod}(f^*) = \text{dom}(g^*)$. We need to show that $\text{dom}(g^* \circ f^*) = \text{dom}(f^*)$. As before, f^* and g^* can be written as $f^* = (a, b, f)$ and $g^* = (b, c, g)$ where $f : a \to b$ and $g : b \to c$. We have $g^* \circ f^* = (a, c, g \circ f)$. Using (3) of definition (3) we obtain $\text{dom}(g^* \circ f^*) = a = \text{dom}(f^*)$.

(9): Let $f^*, g^* \in \text{Arr}$ such that $\text{cod}(f^*) = \text{dom}(g^*)$. We need to show that $\text{cod}(g^* \circ f^*) = \text{cod}(g^*)$. As before, we have $g^* \circ f^* = (a, c, g \circ f)$ and $g^* = (b, c, g)$. Using (4) of definition (3) we obtain $\text{cod}(g^* \circ f^*) = c = \text{cod}(g^*)$.

(10): Let $f^*, g^*, h^* \in \text{Arr}$ with $\text{cod}(f^*) = \text{dom}(g^*)$ and $\text{cod}(g^*) = \text{dom}(h^*)$. We need to show the equality: $(h^* \circ g^*) \circ f^* = h^* \circ (g^* \circ f^*)$. However, f^*, g^*, h^* can be decomposed as $f^* = (a, b, f)$, $g^* = (b, c, g)$ and $h^* = (c, d, h)$ with $f : a \to b$, $g : b \to c$, and $h : c \to d$. We have:

$$\begin{array}{rcl}
(h^* \circ g^*) \circ f^* & = & ((c, d, h) \circ (b, c, g)) \circ (a, b, f) \\
\text{(6) of Def (3)} \to & = & (b, d, h \circ g) \circ (a, b, f) \\
\text{(6) of Def (3)} \to & = & (a, d, (h \circ g) \circ f) \\
\text{assoc of usual composition} \to & = & (a, d, h \circ (g \circ f)) \\
\text{(6) of Def (3)} \to & = & (c, d, h) \circ (a, c, g \circ f) \\
\text{(6) of Def (3)} \to & = & (c, d, h) \circ ((b, c, g) \circ (a, b, f)) \\
 & = & h^* \circ (g^* \circ f^*)
\end{array}$$

(11): Let a be a set. We need to show that $\text{dom}(\text{id}(a)) = a = \text{cod}(\text{id}(a))$. This follows immediately from $\text{id}(a) = (a, a, i(a))$ which is (5) of definition (3).

(12): Let $f^* = (a, b, f)$ be an arrow with $\text{dom}(f^*) = a$. We need to show that $f^* \circ \text{id}(a) = f^*$, which follows from:

$$\begin{array}{rcl}
f^* \circ \text{id}(a) & = & (a, b, f) \circ \text{id}(a) \\
\text{(5) of Def (3)} \to & = & (a, b, f) \circ (a, a, i(a)) \\
\text{(6) of Def (3)} \to & = & (a, b, f \circ i(a)) \\
\text{usual right-identity} \to & = & (a, b, f) \\
 & = & f^*
\end{array}$$

(13): Let $f^* = (b, a, f)$ be an arrow with $\text{cod}(f^*) = a$. We need to show that $\text{id}(a) \circ f^* = f^*$, which follows from:

$$\begin{array}{rcl}
\text{id}(a) \circ f^* & = & \text{id}(a) \circ (b, a, f) \\
\text{(5) of Def (3)} \to & = & (a, a, i(a)) \circ (b, a, f) \\
\text{(6) of Def (3)} \to & = & (b, a, i(a) \circ f) \\
\text{usual left-identity} \to & = & (b, a, f) \\
 & = & f^*
\end{array}$$

This completes our proof of properties (7) − (13). ⋄

Notational convention: We shall denote an arrow (a, b, f) of the category **Set** simply by 'f'. So on top of its usual meaning, '$f : a \to b$' may also express the fact that f is an arrow of the category **Set** with domain a and codomain b.

1.5 Opposite category

Definition 4 *Let* $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ *be a category. We call* opposite category of $\mathcal{C}$*, the category denoted* $\mathcal{C}^{op}$ *and defined by:*

$$\mathcal{C}^{op} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{cod}, \mathrm{dom}, \mathrm{id}, \circ')$$

where the composition operator $\circ'$ *is defined by* $f \circ' g = g \circ f$*, for all* $f, g \in \mathrm{Arr}$.

So if $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ is a category, the opposite category $\mathcal{C}^{op}$ is almost identical, except for the composition operator $\circ'$ which is a flipped version of $\circ$, and for dom and cod which have been swapped with each other. The collection of objects of $\mathcal{C}^{op}$ is the same as that of $\mathcal{C}$, giving us the equality $\mathrm{Ob}\,\mathcal{C}^{op} = \mathrm{Ob}\,\mathcal{C}$. Likewise, the collection of arrows of $\mathcal{C}^{op}$ is the same as that of $\mathcal{C}$, giving us this other equality $\mathrm{Arr}\,\mathcal{C}^{op} = \mathrm{Arr}\,\mathcal{C}$. If we denote $\mathrm{dom}' : \mathrm{Arr} \to \mathrm{Ob}$ and $\mathrm{cod}' : \mathrm{Arr} \to \mathrm{Ob}$ the domain and codomain maps on $\mathcal{C}^{op}$, then $\mathrm{dom}' = \mathrm{cod}$ and $\mathrm{cod}' = \mathrm{dom}$. The identity operator $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ is the same for both $\mathcal{C}$ and $\mathcal{C}^{op}$, and the composition arrow $f \circ' g$ in $\mathcal{C}^{op}$ is defined whenever the composition arrow $g \circ f$ in $\mathcal{C}$ is defined, and we have $f \circ' g = g \circ f$.

Proposition 2 *Let* $\mathcal{C}$ *be a category. Then* $\mathcal{C}^{op}$ *of definition (4) is a category.*

Proof
We need to check that the data $\mathcal{C}^{op} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{cod}, \mathrm{dom}, \mathrm{id}, \circ')$ of definition (4) forms a category, having assumed that the underlying data for $\mathcal{C}$ does. We have indeed two collections Ob and Arr with maps between them $\mathrm{cod} : \mathrm{Arr} \to \mathrm{Ob}$, $\mathrm{dom} : \mathrm{Arr} \to \mathrm{Ob}$, $\mathrm{id} : \mathrm{Ob} \to \mathrm{Arr}$ and partial map $\circ' : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$. So it remains to show that conditions (7) $-$ (13) of definition (2) are satisfied. For the purpose of this proof, we shall denote $\mathrm{dom}' = \mathrm{cod}$ and $\mathrm{cod}' = \mathrm{dom}$.

(7): We need to check that $f \circ' g$ is defined if and only if $\mathrm{cod}'(g) = \mathrm{dom}'(f)$ which is $\mathrm{dom}(g) = \mathrm{cod}(f)$. However by definition, we have set $f \circ' g$ to be defined whenever $g \circ f$ is itself defined, and since $\mathcal{C}$ is a category, this is in turn equivalent to $\mathrm{cod}(f) = \mathrm{dom}(g)$. Hence, we are done.

(8): We need to check that $\mathrm{dom}'(f \circ' g) = \mathrm{dom}'(g)$ which can be written as $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$ and which is true since $\mathcal{C}$ is a category.

(9): We need to check that $\mathrm{cod}'(f \circ' g) = \mathrm{cod}'(f)$ which can be written as $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ and which is true since $\mathcal{C}$ is a category.

(10): Given arrows h, g, f with $\mathrm{cod}'(h) = \mathrm{dom}'(g)$ and $\mathrm{cod}'(g) = \mathrm{dom}'(f)$, we need to check that $(f \circ' g) \circ' h = f \circ' (g \circ' h)$. However, our assumption can be written as $\mathrm{dom}(h) = \mathrm{cod}(g)$ and $\mathrm{dom}(g) = \mathrm{cod}(f)$ and having assumed that $\mathcal{C}$ is a category, by property (10) of definition (2) we have:

$$\begin{array}{rcl} (f \circ' g) \circ' h & = & h \circ (f \circ' g) \\ & = & h \circ (g \circ f) \\ \mathcal{C} \text{ is a category} \rightarrow \quad & = & (h \circ g) \circ f \\ & = & f \circ' (h \circ g) \\ & = & f \circ' (g \circ' h) \end{array}$$

(11): We need to check that $\mathrm{dom}'(\,\mathrm{id}(a)\,) = a = \mathrm{cod}'(\,\mathrm{id}(a)\,)$ for all $a \in \mathcal{C}$, which follows from $\mathrm{dom}' = \mathrm{cod}$, $\mathrm{cod}' = \mathrm{dom}$ and the fact that $\mathcal{C}$ is a category.

(12): We need to check that $f \circ' \mathrm{id}(a) = f$ whenever $\mathrm{dom}'(f) = a$, that is $\mathrm{id}(a) \circ f = f$ whenever $\mathrm{cod}(f) = a$, which follows from $\mathcal{C}$ being a category.

(13): We need to check that $\mathrm{id}(a) \circ' f = f$ whenever $\mathrm{cod}'(f) = a$, that is $f \circ \mathrm{id}(a) = f$ whenever $\mathrm{dom}(f) = a$, which follows from $\mathcal{C}$ being a category. This completes our proof of properties $(7) - (13)$. $\diamond$

Proposition 3 *Let $\mathcal{C}$ be a category. Then the opposite category of $\mathcal{C}^{op}$ is $\mathcal{C}$, i.e.*

$$(\mathcal{C}^{op})^{op} = \mathcal{C}$$

Proof
Let $\mathcal{C} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ be a category. From definition (4), we have $\mathcal{C}^{op} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{cod}, \mathrm{dom}, \mathrm{id}, \circ')$ and consequently:

$$(\mathcal{C}^{op})^{op} = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ'')$$

In order to show that $(\mathcal{C}^{op})^{op} = \mathcal{C}$, by virtue of axiom (4) we simply need to show that the partial maps $\circ, \circ'' : \mathrm{Arr} \times \mathrm{Arr} \to \mathrm{Arr}$ are equal. Given two arrows f and g, the composition arrow $g \circ'' f$ is defined if and only if $f \circ' g$ is defined, which is itself equivalent to $g \circ f$ being defined. By virtue of axiom (2), both $\circ$ and $\circ''$ are therefore defined on the same collection of arrow tuples (g, f). Furthemore, whenever $g \circ f$ is defined, we have $g \circ'' f = f \circ' g = g \circ f$. Using axiom (3) we conclude that $\circ'' = \circ$ as requested. $\diamond$

1.6 Canonical product of categories

Definition 5 *We call* canonical product *of categories $\mathcal{C}_1$ and $\mathcal{C}_2$ the category denoted $\mathcal{C}_1 \times \mathcal{C}_2$ and defined by $\mathcal{C}_1 \times \mathcal{C}_2 = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ where:*

$$\begin{array}{ll}
(1) & \mathrm{Ob} = \{\ (x_1, x_2) \mid x_1 \in \mathrm{Ob}\ \mathcal{C}_1\ ,\ x_2 \in \mathrm{Ob}\ \mathcal{C}_2\ \} \\
(2) & \mathrm{Arr} = \{\ (f_1, f_2) \mid f_1 \in \mathrm{Arr}\ \mathcal{C}_1\ ,\ f_2 \in \mathrm{Arr}\ \mathcal{C}_2\ \} \\
(3) & \mathrm{dom}(f_1, f_2) = (\,\mathrm{dom}(f_1)\,,\,\mathrm{dom}(f_2)\,) \\
(4) & \mathrm{cod}(f_1, f_2) = (\,\mathrm{cod}(f_1)\,,\,\mathrm{cod}(f_2)\,) \\
(5) & \mathrm{id}(x_1, x_2) = (\,\mathrm{id}(x_1)\,,\,\mathrm{id}(x_2)\,) \\
(6) & (g_1, g_2) \circ (f_1, f_2) = (\,g_1 \circ f_1\,,\,g_2 \circ f_2\,)
\end{array}$$

where (3) *and* (4) *hold for all $f_1 \in \mathrm{Arr}\ \mathcal{C}_1$ and $f_2 \in \mathrm{Arr}\ \mathcal{C}_2$,* (5) *holds for all $x_1 \in \mathrm{Ob}\ \mathcal{C}_1$ and $x_2 \in \mathrm{Ob}\ \mathcal{C}_2$, and* (6) *holds for all $f_1, g_1 \in \mathrm{Arr}\ \mathcal{C}_1$ and $f_2, g_2 \in \mathrm{Arr}\ \mathcal{C}_2$ for which $g_1 \circ f_1$ and $g_2 \circ f_2$ are defined.*

So if $\mathcal{C}_1$ and $\mathcal{C}_2$ are two categories, the objects of $\mathcal{C}_1 \times \mathcal{C}_2$ are the collection of all tuples (x_1, x_2) where x_1 is an object of $\mathcal{C}_1$ and x_2 is an object of $\mathcal{C}_2$. The set comprehension notation $\{\ (x_1, x_2) \mid x_1 \in \mathrm{Ob}\ \mathcal{C}_1\ ,\ x_2 \in \mathrm{Ob}\ \mathcal{C}_2\ \}$ is of course an abuse of notation as it does not in general represent a set but a collection. We could also have denoted this collection Ob $\mathcal{C}_1 \times$ Ob $\mathcal{C}_2$ using a cartesian product notation, keeping in mind that this is a product of two collections.

Similarly, the arrows of $\mathcal{C}_1 \times \mathcal{C}_2$ are the collection of all tuples (f_1, f_2) where f_1 is an arrow of $\mathcal{C}_1$ and f_2 is an arrow of $\mathcal{C}_2$, a collection which could reasonably be denoted Arr $\mathcal{C}_1 \times$ Arr $\mathcal{C}_2$ instead of the set-comprehension notation.

It should be clear from definition (5) that the notations 'dom', 'cod', 'id' and '$\circ$' are overloaded, referring either to $\mathcal{C}_1$, $\mathcal{C}_2$ or $\mathcal{C}_1 \times \mathcal{C}_2$. Given our definitions of Ob $(\mathcal{C}_1 \times \mathcal{C}_2)$ and Arr $(\mathcal{C}_1 \times \mathcal{C}_2)$, given that we have dom : Arr $\mathcal{C}_1 \to$ Ob $\mathcal{C}_1$ and dom : Arr $\mathcal{C}_2 \to$ Ob $\mathcal{C}_2$ it should be clear that (3) of definition (5) defines a map dom : Arr $(\mathcal{C}_1 \times \mathcal{C}_2) \to$ Ob $(\mathcal{C}_1 \times \mathcal{C}_2)$, and cod : Arr $(\mathcal{C}_1 \times \mathcal{C}_2) \to$ Ob $(\mathcal{C}_1 \times \mathcal{C}_2)$ follows from (4). Furthermore from id : Ob $\mathcal{C}_1 \to$ Arr $\mathcal{C}_1$ and id : Ob $\mathcal{C}_2 \to$ Arr $\mathcal{C}_2$ we obtain id : Ob $(\mathcal{C}_1 \times \mathcal{C}_2) \to$ Arr $(\mathcal{C}_1 \times \mathcal{C}_2)$ using (5). Finally using (6), given the partial maps $\circ$: Arr $\mathcal{C}_1 \times$ Arr $\mathcal{C}_1 \to$ Arr $\mathcal{C}_1$ and $\circ$: Arr $\mathcal{C}_2 \times$ Arr $\mathcal{C}_2 \to$ Arr $\mathcal{C}_2$ we obtain a partial map $\circ$: Arr $(\mathcal{C}_1 \times \mathcal{C}_2) \times$ Arr $(\mathcal{C}_1 \times \mathcal{C}_2) \to$ Arr $\mathcal{C}_1 \times \mathcal{C}_2$.

Proposition 4 *The canonical product $\mathcal{C}_1 \times \mathcal{C}_2$ of definition (5) is a category.*

Proof
We need to check that the data $\mathcal{C}_1 \times \mathcal{C}_2 = (\mathrm{Ob}, \mathrm{Arr}, \mathrm{dom}, \mathrm{cod}, \mathrm{id}, \circ)$ of definition (5) forms a category, having assumed $\mathcal{C}_1$ and $\mathcal{C}_2$ are categories. We have established that Ob and Arr are collections, that dom, cod, id are maps with the appropriate signatures and $\circ$ is a partial map with the appropriate signature. It remains to check properties (7) $-$ (13) of definition (2).

(7): Let $f, g \in$ Arr $(\mathcal{C}_1 \times \mathcal{C}_2)$. We need to show that $g \circ f$ is defined if and only if $\mathrm{cod}(f) = \mathrm{dom}(g)$. Let $f_1, g_1 \in$ Arr $\mathcal{C}_1$ and $f_2, g_2 \in$ Arr $\mathcal{C}_2$ such that $f = (f_1, f_2)$ and $g = (g_1, g_2)$. Then $\mathrm{cod}(f) = (\mathrm{cod}(f_1), \mathrm{cod}(f_2))$ and $\mathrm{dom}(g) = (\mathrm{dom}(g_1), \mathrm{dom}(g_2))$. So we need to show that $g \circ f$ is defined if and only if $\mathrm{cod}(f_1) = \mathrm{dom}(g_1)$ and $\mathrm{cod}(f_2) = \mathrm{dom}(g_2)$. However since $\mathcal{C}_1$ and $\mathcal{C}_2$ are categories, $\mathrm{cod}(f_1) = \mathrm{dom}(g_1)$ is equivalent to $g_1 \circ f_1$ being defined, and $\mathrm{cod}(f_2) = \mathrm{dom}(g_2)$ is equivalent to $g_2 \circ f_2$ being defined. So we need to show that $g \circ f$ is defined if and only if both $g_1 \circ f_1$ and $g_2 \circ f_2$ are defined which follows exactly from definition (5).

(8): Let $f = (f_1, f_2)$ and $g = (g_1, g_2)$ be arrows in $\mathcal{C}_1 \times \mathcal{C}_2$ with the equality $\mathrm{cod}(f) = \mathrm{dom}(g)$, i.e. for which $g_1 \circ f_1$ and $g_2 \circ f_2$ are defined. We need to check the equality $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ which goes as follows:

$$\begin{array}{rcl}
\mathrm{dom}(g \circ f) & = & \mathrm{dom}(\,(g_1, g_2) \circ (f_1, f_2)\,) \\
\text{(6) of def. (5)} \rightarrow & = & \mathrm{dom}(\,(g_1 \circ f_1\,,\, g_2 \circ f_2)\,) \\
\text{(3) of def. (5)} \rightarrow & = & (\,\mathrm{dom}(g_1 \circ f_1)\,,\, \mathrm{dom}(g_2 \circ f_2)\,) \\
\mathcal{C}_1, \mathcal{C}_2 \text{ categories, (8) of def. (2)} \rightarrow & = & (\,\mathrm{dom}(f_1)\,,\, \mathrm{dom}(f_2)\,) \\
\text{(3) of def. (5)} \rightarrow & = & \mathrm{dom}(f_1, f_2) \\
& = & \mathrm{dom}(f)
\end{array}$$

(9): Let $f = (f_1, f_2)$ and $g = (g_1, g_2)$ be arrows in $\mathcal{C}_1 \times \mathcal{C}_2$ with the equality $\mathrm{cod}(f) = \mathrm{dom}(g)$, i.e. for which $g_1 \circ f_1$ and $g_2 \circ f_2$ are defined. We need to check the equality $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$ which goes as follows:

$$
\begin{array}{rcl}
\mathrm{cod}(g \circ f) & = & \mathrm{cod}(\,(g_1, g_2) \,\circ\, (f_1, f_2)\,) \\
\text{(6) of def. (5)} \rightarrow & = & \mathrm{cod}(\,(g_1 \circ f_1 \,,\, g_2 \circ f_2)\,) \\
\text{(4) of def. (5)} \rightarrow & = & (\,\mathrm{cod}(g_1 \circ f_1)\,,\, \mathrm{cod}(g_2 \circ f_2)\,) \\
\mathcal{C}_1,\, \mathcal{C}_2 \text{ categories, (9) of def. (2)} \rightarrow & = & (\,\mathrm{cod}(g_1)\,,\, \mathrm{cod}(g_2)\,) \\
\text{(4) of def. (5)} \rightarrow & = & \mathrm{cod}(g_1, g_2) \\
 & = & \mathrm{cod}(g)
\end{array}
$$

(10): Let $f = (f_1, f_2)$, $g = (g_1, g_2)$ and $h = (h_1, h_2)$ be arrows in $\mathcal{C}_1 \times \mathcal{C}_2$ with the equalities $\mathrm{cod}(f) = \mathrm{dom}(g)$ and $\mathrm{cod}(g) = \mathrm{dom}(h)$, i.e. for which the composition arrows $g_1 \circ f_1$, $g_2 \circ f_2$, $h_1 \circ g_1$ and $h_2 \circ g_2$ are defined. We need to check the equality $(h \circ g) \circ f = h \circ (g \circ f)$ which goes as follows:

$$
\begin{array}{rcl}
(h \circ g) \circ f & = & (\,(h_1, h_2) \,\circ\, (g_1, g_2)\,) \,\circ\, (f_1, f_2) \\
\text{(6) of def. (5)} \rightarrow & = & (\,h_1 \circ g_1 \,,\, h_2 \circ g_2\,) \,\circ\, (f_1, f_2) \\
\text{(6) of def. (5)} \rightarrow & = & (\,(h_1 \circ g_1) \circ f_1 \,,\, (h_2 \circ g_2) \circ f_2\,) \\
\mathcal{C}_1,\, \mathcal{C}_2 \text{ categories, (10) of def. (2)} \rightarrow & = & (\,h_1 \circ (g_1 \circ f_1)\,,\, h_2 \circ (g_2 \circ f_2)\,) \\
\text{(6) of def. (5)} \rightarrow & = & (h_1, h_2) \circ (\,(g_1 \circ f_1)\,,\, (g_2 \circ f_2)\,) \\
\text{(6) of def. (5)} \rightarrow & = & (h_1, h_2) \circ (\,(g_1, g_2) \,\circ (f_1, f_2)\,) \\
 & = & h \circ (g \circ f)
\end{array}
$$

(11): Let $a = (a_1, a_2)$ be an object in $\mathcal{C}_1 \times \mathcal{C}_2$. We need to check that $\mathrm{dom}\,(\,\mathrm{id}(a)\,) = a = \mathrm{cod}\,(\,\mathrm{id}(a)\,)$ which goes as follows:

$$
\begin{array}{rcl}
\mathrm{dom}\,(\,\mathrm{id}(a)\,) & = & \mathrm{dom}\,(\,\mathrm{id}(a_1, a_2)\,) \\
\text{(5) of def. (5)} \rightarrow & = & \mathrm{dom}\,(\,\mathrm{id}(a_1)\,,\, \mathrm{id}(a_2)\,) \\
\text{(3) of def. (5)} \rightarrow & = & (\,\mathrm{dom}\,(\,\mathrm{id}(a_1)\,)\,,\, \mathrm{dom}\,(\,\mathrm{id}(a_2)\,)\,) \\
\mathcal{C}_1,\, \mathcal{C}_2 \text{ categories, (11) of def. (2)} \rightarrow & = & (a_1, a_2) \\
 & = & a
\end{array}
$$

$$
\begin{array}{rcl}
\mathrm{cod}\,(\,\mathrm{id}(a)\,) & = & \mathrm{cod}\,(\,\mathrm{id}(a_1, a_2)\,) \\
\text{(5) of def. (5)} \rightarrow & = & \mathrm{cod}\,(\,\mathrm{id}(a_1)\,,\, \mathrm{id}(a_2)\,) \\
\text{(4) of def. (5)} \rightarrow & = & (\,\mathrm{cod}\,(\,\mathrm{id}(a_1)\,)\,,\, \mathrm{cod}\,(\,\mathrm{id}(a_2)\,)\,) \\
\mathcal{C}_1,\, \mathcal{C}_2 \text{ categories, (11) of def. (2)} \rightarrow & = & (a_1, a_2) \\
 & = & a
\end{array}
$$

(12): Let $f = (f_1, f_2)$ be an arrow and $a = (a_1, a_2)$ be an object in $\mathcal{C}_1 \times \mathcal{C}_2$ such that $\mathrm{dom}(f) = a$. We need to show that $f \circ \mathrm{id}(a) = f$ which goes as follows: Using (3) of definition (5) and the condition $\mathrm{dom}(f) = a$ we obtain the equation $(\,\mathrm{dom}(f_1)\,,\, \mathrm{dom}(f_2)\,) = (a_1, a_2)$. Hence, we have:

$$
f \circ \mathrm{id}(a) \quad = \quad (f_1, f_2) \circ \mathrm{id}\,(a_1, a_2)
$$

$$
\begin{array}{rcl}
(5) \text{ of def. } (5) \rightarrow & = & (f_1, f_2) \circ (\,\mathrm{id}(a_1)\,,\,\mathrm{id}(a_2)\,) \\
(6) \text{ of def. } (5) \rightarrow & = & (\,f_1 \circ \mathrm{id}(a_1)\,,\, f_2 \circ \mathrm{id}(a_2)\,) \\
\mathcal{C}_1 \text{ category, } \mathrm{dom}(f_1) = a_1 \rightarrow & = & (f_1,\, f_2 \circ \mathrm{id}(a_2)\,) \\
\mathcal{C}_2 \text{ category, } \mathrm{dom}(f_2) = a_2 \rightarrow & = & (f_1, f_2) \\
& = & f
\end{array}
$$

(13): Let $f = (f_1, f_2)$ be an arrow and $a = (a_1, a_2)$ be an object in $\mathcal{C}_1 \times \mathcal{C}_2$ such that $\mathrm{cod}(f) = a$. We need to show that $\mathrm{id}(a) \circ f = f$ which goes as follows: Using (4) of definition (5) and the condition $\mathrm{cod}(f) = a$ we obtain the equation $(\,\mathrm{cod}(f_1)\,,\,\mathrm{cod}(f_2)\,) = (a_1, a_2)$. Hence, we have:

$$
\begin{array}{rcl}
\mathrm{id}(a) \circ f & = & \mathrm{id}\,(a_1, a_2) \circ (f_1, f_2) \\
(5) \text{ of def. } (5) \rightarrow & = & (\,\mathrm{id}(a_1)\,,\,\mathrm{id}(a_2)\,) \circ (f_1, f_2) \\
(6) \text{ of def. } (5) \rightarrow & = & (\,\mathrm{id}(a_1) \circ f_1\,,\,\mathrm{id}(a_2) \circ f_2\,) \\
\mathcal{C}_1 \text{ category, } \mathrm{cod}(f_1) = a_1 \rightarrow & = & (f_1,\, \mathrm{id}(a_2) \circ f_2\,) \\
\mathcal{C}_2 \text{ category, } \mathrm{cod}(f_2) = a_2 \rightarrow & = & (f_1, f_2) \\
& = & f
\end{array}
$$

This completes our proof of properties (7) – (13). $\diamond$

1.7 Hom-sets of a category

Definition 6 *Let $\mathcal{C}$ be a category and $a, b \in \mathcal{C}$. We call* hom-set of $\mathcal{C}$ associated with the ordered pair (a, b) *the collection denoted $\mathcal{C}(a, b)$ and defined as:*

$$
\mathcal{C}(a, b) = \{\ f \in \mathrm{Arr}\ \mathcal{C} \ \mid\ f : a \rightarrow b\ \}
$$

In other words the collection $\mathcal{C}(a, b)$ is the collection of all arrows f in $\mathcal{C}$ such that $\mathrm{dom}(f) = a$ and $\mathrm{cod}(f) = b$. Note that despite being called a 'hom-set', the collection $\mathcal{C}(a, b)$ is generally not a set but an arbitary collection.

Proposition 5 *Let $\mathcal{C}$ be a category and $a, b \in \mathcal{C}$. Then $\mathcal{C}^{op}(a, b) = \mathcal{C}(b, a)$.*

Proof
When working in the context of two categories $\mathcal{C}$ and $\mathcal{C}^{op}$, the notation $f : a \rightarrow b$ is ambiguous as the underlying category is unclear. Let us write $f : a \rightarrow b$ @ $\mathcal{C}$ and $f : a \rightarrow b$ @ $\mathcal{C}^{op}$ to disambiguate. Then if $\mathrm{dom}' = \mathrm{cod}$ and $\mathrm{cod}' = \mathrm{dom}$:

$$
\begin{array}{rcl}
\mathcal{C}^{op}(a, b) & = & \{\ f \in \mathrm{Arr}\ \mathcal{C}^{op} \mid f : a \rightarrow b \text{ @ } \mathcal{C}^{op}\ \} \\
\text{def. } (4) \rightarrow & = & \{\ f \in \mathrm{Arr}\ \mathcal{C} \mid f : a \rightarrow b \text{ @ } \mathcal{C}^{op}\ \} \\
\text{def. } (4) \rightarrow & = & \{\ f \in \mathrm{Arr}\ \mathcal{C} \mid \mathrm{dom}'(f) = a\ ,\ \mathrm{cod}'(f) = b\ \} \\
& = & \{\ f \in \mathrm{Arr}\ \mathcal{C} \mid \mathrm{cod}(f) = a\ ,\ \mathrm{dom}(f) = b\ \} \\
& = & \{\ f \in \mathrm{Arr}\ \mathcal{C} \mid f : b \rightarrow a \text{ @ } \mathcal{C}\ \} \\
& = & \mathcal{C}(b, a)
\end{array}
$$

$\diamond$

Proposition 6 *Let $\mathcal{C}_1$, $\mathcal{C}_2$ be two categories and $a, b \in \mathcal{C}_1 \times \mathcal{C}_2$. Then:*

$$\mathcal{C}_1 \times \mathcal{C}_2\,(a, b) = \mathcal{C}_1(a_1, b_1) \times \mathcal{C}_2(a_2, b_2)$$

where it is understood that $a = (a_1, a_2)$ and $b = (b_1, b_2)$.

Proof
Let $a = (a_1, a_2)$ and $b = (b_1, b_2)$ be objects in the product category $\mathcal{C}_1 \times \mathcal{C}_2$:

$$\begin{array}{rcl}
\mathcal{C}_1 \times \mathcal{C}_2\,(a, b) & = & \mathcal{C}_1 \times \mathcal{C}_2\,[\,(a_1, a_2)\,,\,(b_1, b_2)\,] \\
\text{def. (6)} \rightarrow & = & \{\ f \in \text{Arr}\ (\mathcal{C}_1 \times \mathcal{C}_2)\ \ |\ \ f : (a_1, a_2) \rightarrow (b_1, b_2)\ \} \\
\text{def. (5)} \rightarrow & = & \{\ (f_1, f_2)\ |\ f_1 \in \text{Arr}\ \mathcal{C}_1\ ,\ \ f_2 \in \text{Arr}\ \mathcal{C}_2 \\
 & & ,\ (f_1, f_2) : (a_1, a_2) \rightarrow (b_1, b_2)\ \} \\
 & = & \{\ (f_1, f_2)\ |\ f_1 \in \text{Arr}\ \mathcal{C}_1\ ,\ \ f_2 \in \text{Arr}\ \mathcal{C}_2 \\
 & & ,\ \text{dom}(f_1, f_2) = (a_1, a_2) \\
 & & ,\ \text{cod}(f_1, f_2) = (b_1, b_2)\ \} \\
\text{(3) and (4) of def. (5)} \rightarrow & = & \{\ (f_1, f_2)\ |\ f_1 \in \text{Arr}\ \mathcal{C}_1\ ,\ \ f_2 \in \text{Arr}\ \mathcal{C}_2 \\
 & & ,\ (\,\text{dom}(f_1)\,,\,\text{dom}(f_2)\,) = (a_1, a_2) \\
 & & ,\ (\,\text{cod}(f_1)\,,\,\text{cod}(f_2)\,) = (b_1, b_2)\ \} \\
 & = & \{\ (f_1, f_2)\ |\ f_1 \in \text{Arr}\ \mathcal{C}_1\ ,\ \ f_2 \in \text{Arr}\ \mathcal{C}_2 \\
 & & ,\ f_1 : a_1 \rightarrow b_1\ ,\ \ f_2 : a_2 \rightarrow b_2\ \} \\
\text{product of collections} \rightarrow & = & \{\ f_1 \in \text{Arr}\ \mathcal{C}_1\ |\ f_1 : a_1 \rightarrow b_1\ \} \\
 & \times & \{\ f_2 \in \text{Arr}\ \mathcal{C}_2\ |\ f_2 : a_2 \rightarrow b_2\ \} \\
\text{def. (6)} \rightarrow & = & \mathcal{C}_1(a_1, b_1) \times \mathcal{C}_2(a_2, b_2)
\end{array}$$

$\diamond$

1.8 Locally small category

Definition 7 *A category $\mathcal{C}$ is said to be* locally small *if and only if the hom-set $\mathcal{C}(a, b)$ associated with every ordered pair of objects (a, b) is actually a set.*

Proposition 7 *A category $\mathcal{C}$ is locally small if and only if $\mathcal{C}^{op}$ is locally small.*

Proof
The category $\mathcal{C}$ being locally small is equivalent to $\mathcal{C}(a, b)$ being a set for all $a, b \in \text{Ob}\ \mathcal{C}$. Since $\text{Ob}\ \mathcal{C} = \text{Ob}\ \mathcal{C}^{op}$ and $\mathcal{C}^{op}(a, b) = \mathcal{C}(b, a)$ from proposition (5), this is in turn equivalent to $\mathcal{C}^{op}(a, b)$ being a set for all $a, b \in \text{Ob}\ \mathcal{C}^{op}$. Hence, it is equivalent to $\mathcal{C}^{op}$ being locally small. $\diamond$

Proposition 8 *The product $\mathcal{C}_1 \times \mathcal{C}_2$ of locally small categories is locally small.*

Proof
Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two locally small categories. We need to show that the canonical product $\mathcal{C}_1 \times \mathcal{C}_2$ is itself locally small. In other words, given $a, b \in \mathcal{C}_1 \times \mathcal{C}_2$

we need to show that the collection $\mathcal{C}_1 \times \mathcal{C}_2(a,b)$ is actually a set. However from proposition (6) we have $\mathcal{C}_1 \times \mathcal{C}_2\,(a,b) = \mathcal{C}_1(a_1,b_1) \times \mathcal{C}_2(a_2,b_2)$ where $a = (a_1,a_2)$ and $b = (b_1,b_2)$. So the proposition follows from the fact that both $\mathcal{C}_1(a_1,b_1)$ and $\mathcal{C}_2(a_2,b_2)$ are sets, $\mathcal{C}_1$ and $\mathcal{C}_2$ being locally small. $\diamond$

Chapter 2

Functor

2.1 Functor

Definition 8 *We call* functor *from categories* $\mathcal{C}$ *to* $\mathcal{D}$ *any tuple* (F_0, F_1) *with:*

$$
\begin{array}{ll}
(1) & F_0 : \mathrm{Ob}\ \mathcal{C} \to \mathrm{Ob}\ \mathcal{D} \textit{ is a map} \\
(2) & F_1 : \mathrm{Arr}\ \mathcal{C} \to \mathrm{Arr}\ \mathcal{D} \textit{ is a map} \\
(3) & F_1(f) : F_0(a) \to F_0(b) \\
(4) & F_1(\,\mathrm{id}(a)\,) = \mathrm{id}(\,F_0(a)\,) \\
(5) & F_1(g \circ f) = F_1(g) \circ F_1(f)
\end{array}
$$

where $(3) - (5)$ *hold for all* $a, b, c \in \mathcal{C}$*,* $f : a \to b$ *and* $g : b \to c$*.*

Notational convention: We shall use $F : \mathcal{C} \to \mathcal{D}$ as a notational shortcut for the statement that F *is a functor from the category* $\mathcal{C}$ *to the category* $\mathcal{D}$. If $F = (F_0, F_1)$ we shall also commonly denote F_0 and F_1 simply by 'F'.

So if F is a functor $F : \mathcal{C} \to \mathcal{D}$ we effectively have a map $F : \mathrm{Ob}\ \mathcal{C} \to \mathrm{Ob}\ \mathcal{D}$ acting on objects, and a map $F : \mathrm{Arr}\ \mathcal{C} \to \mathrm{Arr}\ \mathcal{D}$ acting on arrows. These two maps satisfy the consistency condition (3) of definition (8) i.e. that if f is an arrow $f : a \to b$ in $\mathcal{C}$, then $F(f)$ must be an arrow $F(f) : F(a) \to F(b)$ in $\mathcal{D}$. Furthermore, the functor F must preserve the identity operators on $\mathcal{C}$ and $\mathcal{D}$ which is condition (4) of definition (8): for all objects $a \in \mathcal{C}$, we must have $F(\,\mathrm{id}(a)\,) = \mathrm{id}(\,F(a)\,)$. Note that since $\mathrm{id}(a) : a \to a$, by consistency we have $F(\,\mathrm{id}(a)\,) : F(a) \to F(a)$, and since $\mathrm{id}(\,F(a)\,) : F(a) \to F(a)$ the equality makes sense. Another way to express the preservation of identity operators by F is simply $F \circ \mathrm{id} = \mathrm{id} \circ F$ or $F_1 \circ \mathrm{id} = \mathrm{id} \circ F_0$ to be more explicit. However, we should remember that the notation '$\circ$' in these equality does not refer to the composition operator $\circ$ of either $\mathcal{C}$ or $\mathcal{D}$, nor does it in general refer to the usual function composition since id, F_0 and F_1 are maps between collections and not functions between sets. Now going back to our functor F, it must also preserve the composition operators on $\mathcal{C}$ and $\mathcal{D}$, which is condition (5) of definition (8):

For all objects $a, b, c \in \mathcal{C}$ and arrows $f : a \to b$ and $g : b \to c$, we must have $F(g \circ f) = F(g) \circ F(f)$. Note that given these assumptions, the composition arrow $g \circ f$ is well-defined, and by consistency we have $F(f) : F(a) \to F(b)$ and $F(g) : F(b) \to F(c)$, so $F(g) \circ F(f)$ is also well-defined. Furthermore, since $g \circ f : a \to c$ by consistency we have $F(g \circ f) : F(a) \to F(c)$ and since $F(g) \circ F(f) : F(a) \to F(c)$, the equality $F(g \circ f) = F(g) \circ F(f)$ makes sense.

2.2 Hom-functor of a locally small category

Definition 9 *Let $\mathcal{C}$ be a locally small category. We call* hom-functor associated with $\mathcal{C}$ *the functor $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathbf{Set}$ defined by $F = (F_0, F_1)$ with:*

$$\begin{array}{ll} (1) & F_0(a_1, a_2) = \mathcal{C}(a_1, a_2) \\ (2) & F_1(f_1, f_2)(h) = f_2 \circ h \circ f_1 \end{array}$$

where (2) *hold for $a_1, a_2, b_1, b_2 \in \mathcal{C}$, $f_1 : b_1 \to a_1$, $h : a_1 \to a_2$ and $f_2 : a_2 \to b_2$.*

Given a locally small category $\mathcal{C}$, definition (9) defines a tuple $F = (F_0, F_1)$ where F_0 appears to be a map defined on $\mathrm{Ob}\, \mathcal{C} \times \mathrm{Ob}\, \mathcal{C}$ with values in $\mathbf{Set}$, and F_1 appears to be a map defined on $\mathrm{Arr}\, \mathcal{C} \times \mathrm{Arr}\, \mathcal{C}$ with values in some functional space (since it takes an h as argument). Looking at this, it is far from obvious that definition (9) defines a functor $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathbf{Set}$. Hence we state:

Proposition 9 *Let $\mathcal{C}$ be a locally-small category. Then the hom-functor F associated with $\mathcal{C}$ is indeed a functor $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathbf{Set}$.*

Chapter 3

Natural Transformation

Chapter 4

Adjunction

4.1 Definition

Definition 10 *We call* adjunction *an ordered pair* (F, G) *where* F *is a functor* $F : \mathcal{C} \to \mathcal{D}$ *and* G *is a functor* $G : \mathcal{D} \to \mathcal{C}$ *while* $\mathcal{C}$ *and* $\mathcal{D}$ *are two locally-small categories for which there exists a natural isomorphism:*

$$\alpha \;:\; \mathcal{D} \circ (F \times I_{\mathcal{D}}) \;\Rightarrow\; \mathcal{C} \circ (I_{\mathcal{C}^{op}} \times G)$$

in the functor category $[\,\mathcal{C}^{op} \times \mathcal{D}\,,\, \mathbf{Set}\,]$, *where* F *also denotes* $F : \mathcal{C}^{op} \to \mathcal{D}^{op}$.