

EQUIVALENCE V. ISOMORPHISMS IN CATEGORY THEORY

R. TRAYLOR*, PH. D. & V. FADEEV**

§1. INTRODUCTION

The biggest challenge we¹ have found in studying category theory is the lack of "concrete" examples to illustrate concepts that aren't themselves highly abstract. This may be due to our more "vulgar" nature as a primarily applied mathematician (from the viewpoint of many academic mathematicians). However, we firmly believe that if category theory wishes to deliver on the promises it makes to help bind many branches of mathematics (and even the sciences) together, it needs a little boring vulgarity sometimes when illustrating concepts as opposed to a never-ending march towards greater abstraction and generality.

Perhaps we are better poised than many to do the dirty work of illustrating some of these concepts in a more boring way, on topics so familiar many mathematicians ignore them. For this particular article, we will elaborate on the difference between isomorphic categories and equivalent categories. The difference between the two has a degree of subtlety, but exploring a tangible example involving matrices and finite-dimensional vector spaces really clarified these concepts for me.

§2. NECESSARY DEFINITIONS

We will not assume a familiarity with all the terms we will be using, so some definitions for reference are provided here. It is perfectly acceptable to skim these until they are invoked later, then return for further examination. For reference, please see *Abstract and Concrete Categories* by Adámek, Herrlich, and Strecker (1990).

Definition 2.1 (Category). A **category** is a quadruple $\mathbf{A} = (\mathcal{O}, \text{hom}, \text{id}, \circ)$ that consists of

- (1) a class \mathcal{O} whose members are called **A**—objects²
- (2) For each pair (A, B) of **A**—objects, a set $\text{hom}(A, B)$ whose elements are called *morphisms* from A to B . Denoted $A \xrightarrow{f} B$ or $f : A \rightarrow B$. The collection of all morphisms in **A** is denoted $\text{Mor}(\mathbf{A})$
- (3) For each **A**—object A , there is an *identity morphism* $A \xrightarrow{\text{id}_A} A$.³

Date: June 16, 2019.

^{1*}Marquette University/The Math Citadel, ^{**}The Open University, UK

²A class is a generalization of the notion of a set. This concept was created to sidestep issues with Russell's Paradox that sets generate. We couldn't talk about the "set of all sets", or the "set of all vector spaces" without running into a paradox, so we created classes which allow for the creation of a class of all sets with any particular property P . For the purposes of this article, accepting a class as a "bigger set" is sufficient.

³That means unlike group theory, where we just have a single identity element, we have here an identity morphism for each object in the category that simply takes the object to itself. Thus we have many identity morphisms.

(4) A *composition law* that associates with each \mathbf{A} -morphism $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ an \mathbf{A} -morphism $A \xrightarrow{g \circ f} C$ we call the *composite* of f and g .⁴ This composite map $g \circ f$ is subject to the following conditions:

- Composition is associative. That is, for $A \xrightarrow{f} B$, $B \xrightarrow{g} C$, and $C \xrightarrow{h} D$,

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

- For any morphism $A \xrightarrow{f} B$, $id_B \circ f = f$ and $f \circ id_A = f$.⁵
- the sets $\text{hom}(A, B)$ are pairwise-disjoint, meaning that a morphism f cannot belong to $\text{hom}(A, B)$ and $\text{hom}(A', B')$ at the same time. It can only be in one set of morphisms. This condition guarantees each morphism has a unique domain (where we're mapping from) and codomain (where we map to).

For those who have never seen any category theory at all, it is worth a pause to list a couple examples of categories. We shall only list the ones that of interest to this article. There are many more examples, and many more "types" of categories beyond these.

- (1) **Set** is the category of all sets. \mathcal{O} is the class of all sets (so a particular set is an object. \mathbb{N} is an object in this category, as is the set of all permutations of the numbers $\{1, 2, 3\}$. We don't care about individual elements of these sets. We move from one whole set to another whole set. $\text{hom}(A, B)$ is the set of all functions from set A to set B . id_A is just the identity function on a set A , and \circ is our standard composition of functions.
- (2) **Vect** is the category whose objects are all real vector spaces. Morphisms in this category are linear transformations from one vector space to another, so $\text{hom}(A, B)$ is the set of all linear transformations from vector space A to vector space B .⁶ The identity morphisms are identity transformations, and composition in this category is the standard composition of linear maps.

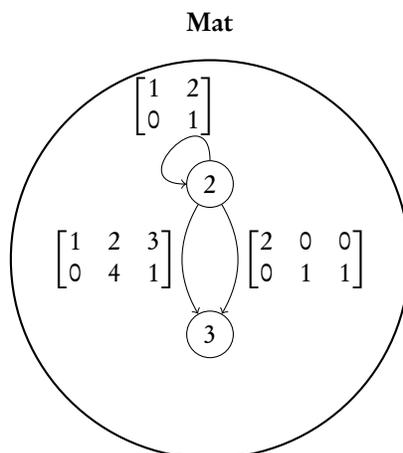


FIGURE 1. The category **Mat**

⁴We first apply f to get into B , then apply g to move into C . Thus we perform compositions right to left.

⁵We just want to make sure that composing with identity morphisms doesn't change anything

⁶Later we'll restrict our attention to only finite dimensional vector spaces.

- (3) **Mat** is an interesting category worth spending some time on. Objects here are *not* matrices, as one might guess. Objects in **Mat** are natural numbers. Morphisms are $m \times n$ real-valued matrices.⁷ Put differently, morphisms take a natural number m "to" a natural number n by assigning it a matrix of dimensions $m \times n$ with real-valued entries. $\text{hom}(m, n)$ is the set of all $m \times n$ real-valued matrices. Structurally, there is only one morphism from m to n . While we can have different $m \times n$ matrices that perform the task of mapping m to n , they are structurally equivalent in that they map two natural numbers to the same dimensions. In this category, the identity morphism id_m are $m \times m$ identity matrices. Composition in this category is defined by $A \circ B = BA$, where BA is just the usual matrix multiplication. It's worth noting here that a category doesn't require that all morphisms be composable. **Mat** provides a nice example of this. We can only multiply matrices when the dimensions are appropriate to do so, which means only certain morphisms are composable. This doesn't violate the definition of a category or associativity, etc. Composition must be associative when composition can be done.

Morphisms allow us to move between objects in a category. If we want to move from one category to another, we'll need the concept of a functor.

Definition 2.2 (Functor). If **A** and **B** are categories, then a **functor** F from **A** to **B** is a mapping that assigns to each **A**-object A and **B**-object FA . It also assigns to each **A**-morphism $A \xrightarrow{f} A'$ a **B**-morphism $FA \xrightarrow{Ff} FA'$ such that the following conditions hold:

- $F(f \circ g) = Ff \circ Fg$ (Composition is preserved whenever $f \circ g$ is defined.)
- For every **A**-object A , $F(id_A) = id_{FA}$. This makes sure the functor maps the identity morphism on A to the identity morphism on the object that A maps to in the category **B**.

Notationally, we'll see functors written $F : \mathbf{A} \rightarrow \mathbf{B}$, or $F(A \xrightarrow{f} B)$, which shows more clearly that functors move objects *and* morphisms.

Functors themselves can have different properties, that are a bit similar to concepts from studying regular functions (e.g. injectivity, surjectivity). We can also compose functors, and that composite functor is indeed a functor.

Definition 2.3 (Isomorphism (Functor)). A functor $F : \mathbf{A} \rightarrow \mathbf{B}$ is called an **isomorphism** if there is a functor $G : \mathbf{B} \rightarrow \mathbf{A}$ such that $G \circ F = id_{\mathbf{A}}$ and $F \circ G = id_{\mathbf{B}}$.⁸

Definition 2.4 (Faithful Functor). A functor is **faithful** if all the hom-set restrictions $F : \text{hom}_{\mathbf{A}}(A, A') \rightarrow \text{hom}_{\mathbf{B}}(FA, FA')$ are injective.

Here, this means that if for a morphism $f \in \text{hom}_{\mathbf{A}}(A, A')$ maps to $Ff \in \text{hom}_{\mathbf{B}}(FA, FA')$, and another morphism $g \in \text{hom}_{\mathbf{A}}(A, A')$ maps to $Fg \in \text{hom}_{\mathbf{B}}(FA, FA')$ we have that $Ff = Fg$ (the two "target" morphisms are the same), then $f = g$. This needs to hold for all sets of morphisms between any two objects.

⁷We can generalize this concept to \mathbf{Mat}_R , where the entries of the matrices are the elements of a particular specified ring R , but let's just stick with real-valued matrices here.

⁸Note id is the identity functor, not the identity morphism on an object. The identity functor will just take all objects to themselves, and all morphisms to themselves, but the identity functor acts on a category, whereas the identity morphism acts on an object within a category.

Definition 2.5 (Full Functor). A functor F is **full** if all the hom-set restrictions $F : \text{hom}_{\mathbf{A}}(A, A') \rightarrow \text{hom}_{\mathbf{B}}(FA, FA')$ are surjective.

In other words, for a functor to be full, we need a morphism g in $\text{hom}_{\mathbf{B}}(FA, FA')$ to have a morphism f in $\text{hom}_{\mathbf{A}}(A, A')$ such that $Ff = g$. g should have a "preimage" under the functor.

Definition 2.6 (Isomorphism-dense). A functor is **isomorphism-dense** if for any \mathbf{B} -object B there exists some \mathbf{A} -object A such that $FA \cong B$.

Two objects FA and B in a category \mathbf{B} are *isomorphic* if there exists a morphism between the two that is an isomorphism. It is worth clearly noting the definition of an isomorphism at it pertains to morphisms in a category:

Definition 2.7 (Isomorphism (Morphism)). A morphism $f : A \rightarrow B$ in a category \mathbf{A} is an isomorphism if there exists a morphism $g : B \rightarrow A$ in $\text{Mor}(\mathbf{A})$ such that $f \circ g = id_B$ and $g \circ f = id_A$ where id_A and id_B are the identity morphisms for the objects A and B .

Finally, we can combine a few of these to get the notion of an *equivalence*:

Definition 2.8 (Equivalence). A functor is an **equivalence** if it is full, faithful, and isomorphism-dense.

If we re-write the definition of an isomorphism (functor), then an isomorphism is a functor that is full, faithful, and bijective on objects, whereas an equivalence is a functor that is full, faithful, and isomorphism-dense. The only difference between these two is the notion of being bijective on objects v. being isomorphism-dense. An isomorphism is more restrictive than an equivalence in the sense that all isomorphisms are equivalences, but we can exhibit equivalences that are not isomorphisms. When we talk about categories being either equivalent or isomorphic, these two words are not interchangeable. In the next sections, we'll discuss this difference, and illustrate with an explicit example.

§3. LINEAR ALGEBRA AND MATRICES

At this point, we'll revisit briefly some standard results from linear algebra, focusing on real vector spaces. In most undergraduate linear algebra courses, the notion that a linear transformation can be represented in matrix form is simply stated (sometimes proven, but not often). In particular, those linear algebra courses designed for engineers may only focus on linear systems of equations (which are linear transformations) and omit much talk of linear transformations at all. I then consider it reasonable to revisit some of these ideas. For more details, please consult your favorite linear algebra text.

3.1. LINEAR SPACES (VECTOR SPACES). *Remark: The terms linear space and vector space are interchangeable. Different texts will use different terms.*

Definition 3.1 (Linear Space). A **linear space** over a field \mathbb{F} is a set of elements (vectors) V with two operations $+, \cdot$. The operation $+$ is called **vector addition** takes two elements $v, w \in V$ and returns another element $v + w \in V$. The operation \cdot is scalar multiplication, and takes a scalar element α from the field \mathbb{F} and multiplies it by an element $v \in V$ and returns a "scaled" element $\alpha v \in V$. We require *closure* under linear combinations. For any $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}$ and $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in V$,

$$\sum_{i=1}^n \alpha_i \mathbf{v}_i \in V$$

. This is a consolidation of the ten axioms a linear space must satisfy. The expanded axioms are relegated to the appendix.

We're going to focus on real vector spaces rather than this abstract definition. Real vector spaces have \mathbb{R} as our field (so we multiply vectors by real numbers).

The vector spaces most often encountered in engineering problems are \mathbb{R}^n —the standard n —dimensional real vector spaces on real-valued vectors. (Also known as a standard Cartesian space.)

Now suppose we wish to move from one vector space V to another W . There are many ways we can define such a move, but the one we're interested in are **linear transformations**.

Definition 3.2 (Linear Transformations). If V, W are linear spaces, then a function $T : V \rightarrow W$ is a linear transformation if

- $T(x + y) = T(x) + T(y)$ for all $x, y \in V$. If we add two vectors, and then transform the result into the space W , it should yield the same vector in W as if we transformed each of x and y into the space W , then added them together.
- $T(\alpha x) = \alpha T(x)$ for all $\alpha \in \mathbb{F}, x \in V$. (We can "factor out" scalars.)

We can take linear transformations from $\mathbb{R}^m \rightarrow \mathbb{R}^n$ by transforming coordinates as in the following example:

Example 3.1. $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is given by $T(x) = x + 2y$ and $T(y) = 3y$.

From linear algebra, we know we may rewrite the transformation in vector form, since we live in "vector space" (in 2 dimensions here):

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + 2y \\ 3y \end{bmatrix}$$

Typically the next step in linear algebra is to say that we may represent the transformation T in matrix form such that multiplication of our "source vector" by the matrix yields the result of the linear transformation. For our example, the matrix A corresponding to the linear transformation T is $A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$, so that for $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

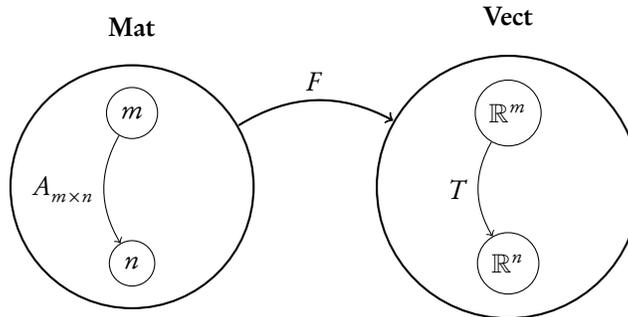
$$A\mathbf{x} = \begin{bmatrix} x + 2y \\ 3y \end{bmatrix}$$

At this point, this equivalence between linear transformations and matrix representation (with operation matrix/vector multiplication) is simply stated and expected to be accepted. However, we're going to use the notion of equivalences in category theory to establish this formally.

§4. MOVING FROM MAT TO VECT

As mentioned before, we're going to focus only on finite dimensional vector spaces (\mathbb{R}^n for various values of n or other spaces of finite dimension) and real-valued matrices to keep things simple. These concepts generalize much further, though. Since we're operating on real-valued matrices, we'll be using $\mathbf{Mat}_{\mathbb{R}}$ ⁹

⁹If I changed the values of the matrices to, say, complex numbers, we would be using a totally separate category. Changing the values of the matrices changes the ring and thus denotes a completely separate category of matrices

FIGURE 2. The functor F between \mathbf{Mat} and \mathbf{Vect}

To show the equivalence between \mathbf{Vect} and \mathbf{Mat} , we must establish an equivalence functor between the two that allows us to map \mathbf{Mat} to \mathbf{Vect} . We define the functor as follows:

$F : \mathbf{Mat} \rightarrow \mathbf{Vect}$ assigns objects n in \mathbf{Mat} the vector space \mathbb{R}^n . Thus, we simply assign a natural number n in \mathbf{Mat} to the corresponding n -dimensional real vector space in \mathbf{Vect} , which is an object in \mathbf{Vect} .

Functors also map morphisms in the source category (\mathbf{Mat} in our case) to morphism in the destination category (\mathbf{Vect} here). Morphisms in \mathbf{Mat} are $m \times n$ matrices with real values. Morphisms in \mathbf{Vect} are linear transformations. Thus, it's natural to have our functor take an $m \times n$ matrix A in \mathbf{Mat} and assign to it the linear map T in $\text{Mor}(\mathbf{Vect})$

from $\mathbb{R}^m \rightarrow \mathbb{R}^n$ that takes an m -dimensional vector $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \in \mathbb{R}^m$ maps it to the

n -dimensional vector $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}^T \in \mathbb{R}^n$ given by $\mathbf{x}^T A$.¹⁰

We need to check that this functor satisfies the definition of both a functor and an equivalence. That F is a functor follows from matrix multiplication on one hand, and from composition of linear maps on the other. To ensure it satisfies the definition of an equivalence, we must check to see that the functor is full, faithful, and isomorphism-dense.

To show the functor is faithful, take two linear transformations T and T' in \mathbf{Vect} such that $T = T'$. In other words, these two transformations do the exact same thing to input vectors. It's clear there can't be if we had two different matrices A and A' such that multiplication by \mathbf{x} yields the exact same vector \mathbf{y} , then $A = A'$. Thus the functor is indeed faithful.

To show the functor is full, we need to show that for any linear transformation $T \in \text{Mor}(\mathbf{Vect})$, there is a matrix $A \in \text{Mor}(\mathbf{Mat})$ such that $F(A) = T$. A linear transformation by definition (see above) results in a component $x_i \in \mathbb{R}^m$ being taken to a linear combination of components $x_j \in \mathbb{R}^n$. Refer back to the example above, where we mapped from \mathbb{R}^2 to \mathbb{R}^2 .

¹⁰T is transpose here, and we only needed to shift some things (like multiplying on the right) to keep our "direction" matching the $m \times n$ matrix dimension and appropriateness of matrix multiplication

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + 2y \\ 3y \end{bmatrix}$$

Since multiplication of matrices results in components which are linear combinations of the source vector by definition of matrix multiplication, there is certainly a matrix $A \in \text{Mor}(\mathbf{Mat})$ that will map to $T \in \text{Mor}(\mathbf{Vect})$ for any linear transformation T . Thus, F is certainly a full functor.

It remains to show that F is isomorphism-dense. To do so, we're going to attempt to create its "inverse" functor $G : \mathbf{Vect} \rightarrow \mathbf{Mat}$. It is at this point we'll really understand the difference between being isomorphism-dense and being a true isomorphism.

§5. CREATING THE INVERSE FUNCTOR

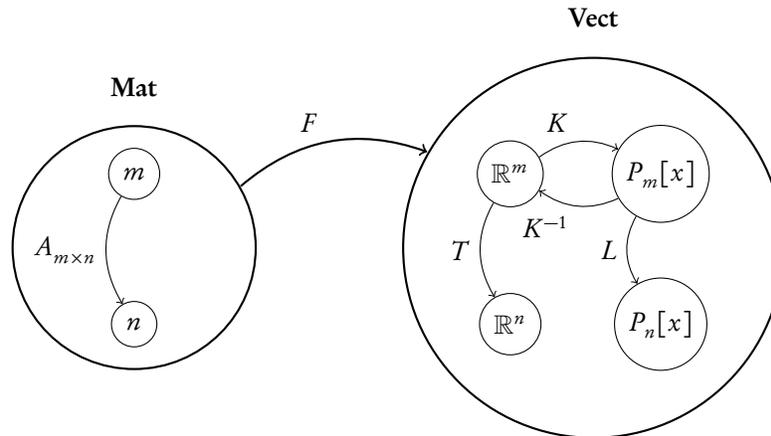


FIGURE 3. The functor F between \mathbf{Mat} and \mathbf{Vect}

We're going to create a functor G that maps \mathbf{Vect} to \mathbf{Mat} . For this, we'll define it as the "natural" inverse. G will take vector spaces in \mathbf{Vect} to their dimension $n \in \mathbb{N}$, which is an object in \mathbf{Mat} . We'll also take $T \in \text{Mor}(\mathbf{Vect})$ to its corresponding matrix $A \in \text{Mor}(\mathbf{Mat})$, simply "reversing" the direction.

We can show in an identical manner to what we did for F that the functor G is full. Isomorphisms in \mathbf{Vect} are invertible linear maps between vector spaces. Since the objects of \mathbf{Mat} are natural numbers, we cannot have isomorphisms between different objects. That is, m cannot be isomorphic to n inside \mathbf{Mat} unless $m = n$. However, we can certainly have isomorphisms between linear spaces of different dimensions.

We now illustrate the key difference between isomorphism and isomorphism-dense functors. Take the space of polynomials of order m with real coefficients. This is indeed a real vector space, and belongs to \mathbf{Vect} . This polynomial space is isomorphic to \mathbb{R}^m inside of \mathbf{Vect} , meaning that there is an invertible linear map K from the polynomial space to \mathbb{R}^m . Recall that functors not only map objects to objects, but they also map morphisms to morphisms. Since all the functor F originating \mathbf{Mat} cares about is the dimension of the space, the isomorphism K between the polynomial space and \mathbb{R}^m is "invisible" in the category \mathbf{Mat} , because \mathbf{Mat} sees the polynomial space and \mathbb{R}^m just as

"some" m -dimensional spaces. Thus the morphism $K \in \text{Mor}(\mathbf{Vect})$ has no morphism in $\text{Mor}(\mathbf{Mat})$ that will map to it.

The functor G is not faithful because T and L will map to the same matrix $A_{m \times n}$ in $\text{Mor}(\mathbf{Mat})$.

Therefore, the two categories \mathbf{Vect} and \mathbf{Mat} are equivalent, but not isomorphic, because F is only isomorphism-dense, not fully isomorphic.

§6. CONCLUSION

We exhibited an explicit example involving the category \mathbf{Mat} of matrices over \mathbb{R} and the category \mathbf{Vect} of real vector spaces to illustrate the difference between functors that are isomorphisms and functors that are equivalences. The key difference is the notion of *isomorphism-dense*, which means that an object in the target category of a functor may not have a "preimage" in the source space, but is isomorphic inside the target category to an object which does have a "preimage" in the source space.

§7. APPENDIX

For a linear space V over a field \mathbb{F} , the following axioms regarding the two operations must be satisfied:

- (1) For $x, y \in V$, $x + y \in V$. That is, if we perform the addition operation on two elements of V , the result can't be outside the set V .
- (2) For each $\alpha \in \mathbb{F}$ and $v \in V$, $\alpha v \in V$. Multiplying an element by a scalar shouldn't "knock" us outside the set V .
- (3) For $x, y \in V$, $x + y = y + x$. (Commutativity)
- (4) For $x, y, z \in V$, $x + (y + z) = (x + y) + z$ (Associativity)
- (5) There must be some element in V , denoted 0_V that is an *additive identity*. That is, for any $v \in V$, $v + 0_V = 0_V + v = v$.
- (6) For every $x \in V$, the element $(-1)x$ has the property that $x + (-1)x = 0_V$. (Existence of "negatives", or additive inverses)
- (7) For $\alpha, \beta \in \mathbb{F}$, $v \in V$, $\alpha(\beta v) = (\alpha\beta)v$. (Associativity of multiplication)
- (8) For all $x, y \in V$, and for all $\alpha \in \mathbb{F}$, $\alpha(x + y) = \alpha x + \alpha y$ (Distributivity of vector addition)
- (9) For all $\alpha, \beta \in \mathbb{F}$, $v \in V$, $(\alpha + \beta)v = \alpha v + \beta v$.¹¹
- (10) For every $v \in V$, we have that $1v = v$, where 1 is the multiplicative identity in \mathbb{F}

§8. ACKNOWLEDGEMENTS

We'd like to gratefully acknowledge Matt Kukla and Jason Hathcock for their insights, review, and meaningful discussion.

¹¹Addition of α and β is performed in the field, and may not be the same as vector addition. Vector addition gives a vector. Scalar addition give scalars