Representations of a concrete category as objects in the functor category

Tibor Grün

July 8, 2020

Contents

1 Introduction 1

2 A short overview of the tools used 1

3 Quivers, categories and k-algebroids 1

4 Limits and colimits 3
- 4.1 Monomorphisms and epimorphisms . . . 3
- 4.2 Kernel and cokernel; image and coimage . . . 3
- 4.3 Direct sum and direct product . . . 3

5 Functors and natural transformations 3
- 5.1 Functors map one category to another . . . 3
- 5.2 Natural transformations are morphisms between functors . . . 4

6 Adjunctions 4
- 6.1 Universal objects . . . 4
- 6.2 Forgetting the forgetful functor: Free constructions . . . 4

7 Yoneda's Lemma: Completion and cocompletion of a category 4
- 7.1 Embedding categories . . . 4
- 7.2 Taking what a category lacks . . . 4

8 Functors and natural transformations 4
- 8.1 Functors act on objects and morphisms of a category . . . 4
- 8.2 Natural transformations are morphisms between functors . . . 4
- 8.3 Representations are Functors into a matrix category . . . 4

9 Relations of the Algebroid 5
- 9.1 Relations of endomorphisms . . . 5

10 Category 5

11 $\mathbb{K}$-linear Category (Algebroid) 6

12 Additive Category 7

13 Abelian Category 7

14 The Category of Categories 7

15 The Categories of Functors 7

16 The Representation of a Category 7

17 Representation 7

18 Algorithms 7

1 Introduction

$$\mathbf{Quiv} \to^{CatClosure} \leftarrow_U \mathbf{Cats} \to^{k-Algebroid} \leftarrow_U \mathbf{k}-\mathbf{Cats} \to^{AdditiveClosure} \leftarrow_U \mathbf{k}-\mathbf{Cats}^{\oplus}$$

2 A short overview of the tools used

GAP, QPA / QPA2, Catreps, CAP, homalg_project

3 Quivers, categories and k-algebroids

Definition 3.1. (Quiver)
A directed graph or quiver q consists of a class of objects (or vertices) $q_0 = \operatorname{Obj} q$ and a class of morphisms (or arrows) $q_1 = \operatorname{Mor} q$ together with two defining maps

$$s,t\colon q_1 \rightrightarrows q_0$$

s called source and t called target.

In `QPA`, the objects are coded by natural numbers, so the first object is 1, the second 2 and so on. The arrows are denoted by small letters a,b,c and so on. There is a difference between `RightQuiver` and `LeftQuiver` in that the right quiver is right-oriented (that is, the convention for order in multiplication of paths is the opposite of that used for left-oriented quivers).

Definition 3.2. (Hom-set of a (locally) small quiver)

(1) Given two objects $M,N \in q_0$ we write $\operatorname{Hom}_q(M,N)$ or $q(M,N)$ for the fiber $(s,t)^{-1}(\{(M,N)\})$ of the product map $(s,t) : q_1 \longrightarrow q_0 \times q_0$ over the pair $(M,N) \in q_0 \times q_0$. This is the class of all morphisms with source $= M$ and target $= N$. We indicate this by writing $\varphi : M \longrightarrow N$ or $M \xrightarrow{\varphi} N$. Hence q_1 is the disjoint union $\dot{\bigcup}_{M,N \in q_0} \operatorname{Hom}_q(M,N) = q_1$. As usual we define $\operatorname{End}_q(M) := \operatorname{Hom}_q(M,M)$.

(2) If the class $\operatorname{Hom}_q(M,N)$ is a set for all pairs (M,N) then we call the quiver locally small. We therefore talk about Hom-sets. If additionally, q_0 is a set, then the quiver is called small.

Example 3.3. (Quiver with 2 objects and 3 morphisms)

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

The objects of this quiver q are $q_0 = \{1,2\}$, and the morphisms are $q_1 = \{a,b,c\}$ with
$s(a) = 1 = t(a)$, $s(c) = 2 = t(c)$ and $s(b) = 1, t(b) = 2$.
Thus $\operatorname{End}_q(1) = \{a\}, \operatorname{End}_q(2) = \{c\}$ and $\operatorname{Hom}_q(1,2) = \{b\}$ whereas $\operatorname{Hom}_q(2,1) = \varnothing$.

In `QPA` this quiver is encoded as `q(2)[a:1->1,b:1->2,c:2->2]` where the first `(2)` in parentheses stands for the total number of objects.

Definition 3.4. (Composable arrows; path in a quiver)
Since we already have the source and target maps, we say two arrows $a,b \in q_1$ are composable if $t(a) = s(b)$ or $t(b) = s(a)$. In this case we can write a sequence of composable arrows $p = a_1 a_2 \cdots a_n$ where $t(a_i) = s(a_{i+1})$ for $i = 1,\ldots,n-1$. We call this sequence a path from $s(a_1)$ to $t(a_n)$ and the integer $n \in \mathbb{Z}_{\geq 0}$ the length $l(p)$ of the path p. Although it's not an arrow, we can define the source and target of a path $p = a_1 \cdots a_n$ as $s(p) := s(a_1)$ and $t(p) := t(a_n)$. A path $p = a_1 \cdots a_n$ with $s(a_1) = t(a_n)$, i.e. $s(p) = t(p)$, is called cyclic.
For an endomorphism $a \in \operatorname{End}_q(M)$ we write a^n for $aa\cdots a$ (n times). In the case of $n = 0$ an empty path whose source and target are the vertex $i \in q_0$ is called the trivial path at i and is denoted e_i. Note that the composition of paths $e_i e_i$ has length zero starting at i therefore $e_i^2 = e_i$.

Lemma 3.5. *Let Q be a quiver. If there is a path of length at least $|Q_0|$, then there are cyclic paths, and thus infinitely many paths.*

Proof. Assume that there exists a path of length greater or equal to $|Q_0|$. Then there exists a path of length $n = |Q_0|$, say $\alpha_1 \cdots \alpha_n$. Consider the vertices $x_i = s(\alpha_i)$ for $1 \leq i \leq n$ and $x_{n+1} = t(\alpha_n)$. Then these are $n+1$ vertices, thus there has to exist $i < j$ with $x_i = x_j$. Let $\omega = \alpha_i \cdots \alpha_{j-1}$, this is a path with source and target $x_i = x_j$, thus a cyclic path. But then ω^m is a path for any natural number m. The path ω has length $j - i \geq 1$, thus ω^m has length $m(j-i)$. This shows that these paths are pairwise different. □

Example 3.6. (A quiver with no cycles)

$$\begin{tikzcd}
2 \arrow[r, "\psi"] \arrow[dr, "\psi\rho"] & 3 \arrow[d, "\rho"] \\
1 \arrow[u, "\varphi"] \arrow[ur, "\varphi\psi"] \arrow[r, "\varphi\psi\rho"'] & 4
\end{tikzcd}$$

Definition 3.7. (Category)

A <u>category</u> $\mathcal{C}$ is a quiver with two further maps:

(id) The <u>identity map</u> $1_{()}$ mapping every object $X \in \mathcal{C}_0$ to its <u>identity morphism</u> 1_X:

$$\mathcal{C}_0 \xrightarrow{1} \mathcal{C}_1$$

(μ) And for any two <u>composable</u> morphisms φ and $\psi \in \mathcal{C}_1$, i.e. with $t(\varphi) = s(\psi)$, the <u>composition map</u> μ, which maps $\varphi, \psi \in \mathcal{C}_1 \times \mathcal{C}_1$ to $\mu(\varphi, \psi) \in \mathcal{C}_1$ which we also write as $\varphi\psi$.

$$\mathcal{C}_1 \times \mathcal{C}_1 \xrightarrow{\mu} \mathcal{C}_1$$

The defining properties for 1 and μ are:

(1) $s(1_M) = M = t(1_M)$, i.e.
$1_M \in \mathrm{End}_{\mathcal{C}} \,\forall M \in \mathcal{C}$.

(2) $s(\varphi\psi) = s(\varphi)$ and
$t(\varphi\psi) = t(\psi)$
for all composable morphisms $\varphi, \psi \in \mathcal{C}$.

$$\mu : \mathrm{Hom}_{\mathcal{C}}(M, L) \times \mathrm{Hom}_{\mathcal{C}}(L, N) \longrightarrow \mathrm{Hom}_{\mathcal{C}}(M, N)$$

(3) $(\varphi\psi)\rho = \varphi(\psi\rho)$ [associativity of composition]

(4) $1_{s(\varphi)}\varphi = \varphi = \varphi 1_{t(\varphi)}$ [unit property]
The identity is a left and right <u>unit</u> of the composition.

As we have seen, every category is a quiver, but in general, to become a category, a quiver is lacking identity morphisms and the composition of morphisms. To be more precise, there is a <u>functor</u> U from the <u>category of categories</u> CAT to the <u>category of quivers</u> Quiv, called the <u>underlying quiver</u> or <u>forgetful functor</u>.

$$\mathrm{CAT} \xrightarrow{U} \mathrm{Quiv}$$

mapping every object $M \in \mathcal{C}_0$ to the same objects in q_0, mapping every arrow $\varphi \in \mathcal{C}_1$ to an arrow $a \in q_1$, respecting source and target, but forgetting the special role of the identity morphisms and of the composition morphisms.

Example 3.8. (Underlying quiver)

$$
\begin{tikzcd}
B \arrow[loop above, "1_B"] \arrow[rr, "\psi"] \arrow[rrd, "\psi\rho" near start] & & C \arrow[loop above, "1_C"] \arrow[d, "\rho"] \\
A \arrow[loop below, "1_A"] \arrow[u, "\varphi"] \arrow[rru, "\varphi\psi"' near start] \arrow[rr, bend left, "(\varphi\psi)\rho"] \arrow[rr, bend right, "\varphi(\psi\rho)"'] & & D \arrow[loop below, "1_D"]
\end{tikzcd}
\quad \xRightarrow{U} \quad
\begin{tikzcd}
2 \arrow[loop above, "h"] \arrow[rr, "b"] \arrow[rrd, "e" near start] & & 3 \arrow[loop above, "i"] \arrow[d, "c"] \\
1 \arrow[loop below, "g"] \arrow[u, "a"] \arrow[rru, "d"' near start] \arrow[rr, bend left, "f"] \arrow[rr, bend right, "f"'] & & 4 \arrow[loop below, "j"]
\end{tikzcd}
$$

In the category on the left, associativity of composition guaranteed that $(\varphi\psi)\rho = \varphi(\psi\rho)$, so those two arrows were already the same, so they are mapped to the same arrow $f = U((\varphi\psi)\rho) = U(\varphi(\psi\rho))$ in the quiver on the right. We didn't have to draw both arrows for f, but since they are equal, there is still only one arrow in the hom-set $\mathrm{Hom}_q(1,4) = \{f, f\} = \{f\}$.
All the other identities are not preserved under the forgetful functor, e.g. d doesn't know what it has to do with a and b apart from $s(d) = s(a)$ and $t(d) = t(b)$. Especially the former identity arrows are now just endomorphisms with no defining property.
The paths $g^2 f, gf$ and fj^3 are all different, while in the category, they all simplify to $1_A 1_A(\varphi\psi)\rho = 1_A(\varphi\psi)\rho = (\varphi\psi)\rho 1_D 1_D 1_D = (\varphi\psi)\rho$ due to the unit property and associativity.

Definition 3.9. (Ab-category) An Ab-category is a category in which all homomorphism sets are abelian groups, and composition distributes over addition.
In other words, A category $\mathcal{C}$ is an Ab-category if for every pair of objects $M, N \in \mathcal{C}_0$, $(\mathrm{Hom}_{\mathcal{C}}(M,N), +)$ is an abelian group (with the neutral element called zero morphism), and for all morphisms $\gamma, \delta \in \mathrm{Hom}_{\mathcal{C}}(M,N), \alpha, \beta \in \mathrm{Hom}_{\mathcal{C}}(N,L)$

$$(\gamma + \delta)\alpha = \gamma\alpha + \delta\alpha \text{ and}$$
$$\gamma(\alpha + \beta) = \gamma\alpha + \gamma\beta.$$

Note that every hom-set has its own unique zero morphism. E.g. in $\mathrm{Mat}_{\mathbb{Q}}$ the 2-by-3 zero-matrix $0 \in \mathrm{Hom}(2,3)$ is different from the 4-by-4 zero-matrix $0 \in \mathrm{Hom}(4,4)$.

Definition 3.10. (Initial object, terminal object, zero object)

Example 3.11.

Definition 3.12. (Kernel of a morphism

Definition 3.13. (Abelian category)

Definition 3.14. (k-linear category)

Quiver -¿ CAT: U: forget 1, forget composition
search U^{-1}
Beispiel für Adjunktion
Path Algebra:

4 Limits and colimits

4.1 Monomorphisms and epimorphisms

4.2 Kernel and cokernel; image and coimage

4.3 Direct sum and direct product

5 Functors and natural transformations

5.1 Functors map one category to another

Example 5.1. (Identity Functor)

Example 5.2. (Forgetful functor)

Definition 5.3. (full functor; faithful functor)

5.2 Natural transformations are morphisms between functors

6 Adjunctions

6.1 Universal objects

6.2 Forgetting the forgetful functor: Free constructions

7 Yoneda's Lemma: Completion and cocompletion of a category

7.1 Embedding categories

7.2 Taking what a category lacks

Example 7.1. (Probability theory: The σ algebra of a random variable)

$$\begin{tikzcd}
(\Omega_1, \sigma_1) \arrow[r, "X"] \arrow[d] & (\Omega_2, \sigma_2, P) \\
(\Omega_1, \sigma_1, P_X) \arrow[ur] &
\end{tikzcd}$$

8 Functors and natural transformations

8.1 Functors act on objects and morphisms of a category

8.2 Natural transformations are morphisms between functors

8.3 Representations are Functors into a matrix category

Yonedas Einbettungs-Lemma: Fehlende Limiten bzw. Kolimiten exitieren nach der Einbettung.

Einbettung in Kategorien, die mehr Limiten haben als die Zielkategorie.

"(Ko-)Vervollständigung" der Kategorie (Completion / Cocompletion)

Quiver = unvollständige Struktur einer Kategorie Erzeugendensystem einer Kategorie.

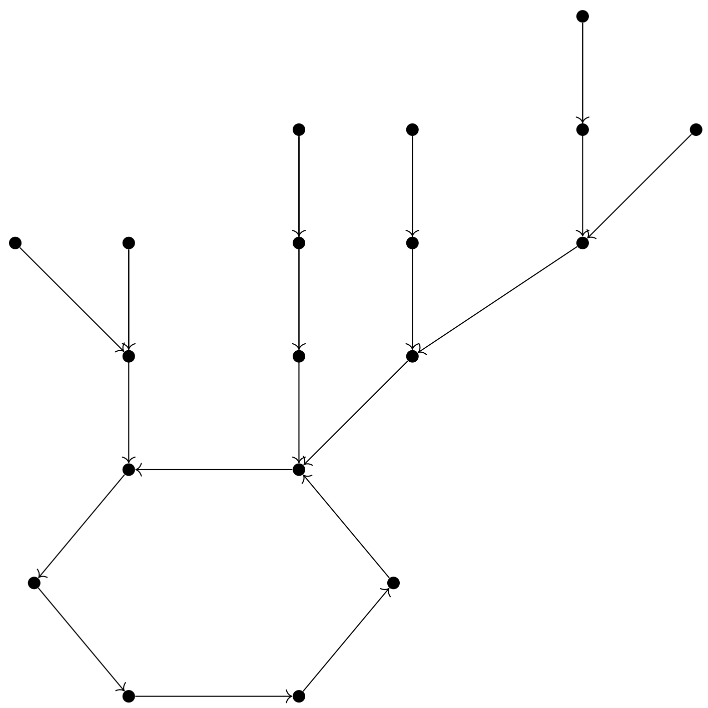
K-linearer Abschluss einer Kategorie

Pfadalgebra = Kategorien-Algebra path algebra = 1 Object, welches eine Algebra ist. Dabei verliert man wieder die Informationen über die mehreren Objekte.

So wie Menge ein Erz-system eines Monoid.

9 Relations of the Algebroid

9.1 Relations of endomorphisms



Lemma 9.1 (σ-Lemma). *For each endomorphism f in a finite concrete category $\mathcal{C}$ there exist $m, n \in \mathbb{N}$ such that $f^{(m+n)} = f^m$.*

Beschreibung der Algorithmen
WeakDirectSumDecomposition ¡– Tiefensuche. Objekte (Funktoren) in indecomposable Functors.

10 Category

Definition 10.1. (Quiver)
A quiver A consists of a class of objects (or vertices) $A_0 = \mathrm{Obj}A$ and a class of morphisms (or arrows) $A_1 = \mathrm{Mor}A$ together with two defining maps

$$s, t \colon A_1 \rightrightarrows A_0$$

s called source and t called target.[1]
We write $\mathrm{Hom}_A(M, N)$ (sometimes also $A(M, N)$) for the fiber $(s, t)^{-1}(\{(M, N)\})$ of the product map $(s, t) : A_1 \to A_0 \times A_0$ over the pair $(M, N) \in A_0 \times A_0$.
This is the class of all morphisms with source = M and target = N.
For a morphism $\varphi \in \mathrm{Hom}_A(M, N)$ we write

$$\varphi : M \longrightarrow N \text{ or } M \xrightarrow{\varphi} N$$

Clearly A_1 is the disjoint union $\dot{\bigcup}_{M,N \in A_0} \mathrm{Hom}_A(M, N) = A_1$. As usual we define $\mathrm{End}_A(M) := \mathrm{Hom}_A(M, M)$.

Definition 10.2. (Category)
A category $\mathfrak{A}$ is a quiver with two further defining maps

$$A_0 \xrightarrow{1} A_1 \xleftarrow[\mu]{} A_1 \times_{s, A_0, t} A_1$$

Example 10.3. (Representation of a concrete category)

$$
\begin{tikzcd}
5 \arrow[r, "\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}"] \arrow[loop below, "\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}"] & 4 \arrow[loop below, "\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}"]
\end{tikzcd}
$$

$$nine \Big\Uparrow$$

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

$$
\begin{tikzcd}
\{1,2,3\} \arrow[r, "{(4,5,6)}"] \arrow[loop below, "{(2,1,3)}"] & \{4,5,6\} \arrow[loop below, "{(5,6,4)}"]
\end{tikzcd}
$$

$$F(a)\eta_1 = \eta_1 G(a) F(b)\eta_2 = \eta_1 G(b)$$

11 $\mathbb{K}$-linear Category (Algebroid)

Group: Category with one object.

Groupoid: A small category in which every morphism is an isomorphism.

Algebroid

EmbeddingOfSumOfImages

What is an Algebroid? Bialgebroid?

12 Additive Category

13 Abelian Category

14 The Category of Categories

15 The Categories of Functors

16 The Representation of a Category

17 Representation

Grundidee von FunctorCategory
Standard-Monoidale Struktur von der Zielkategorie z.B. TensorUnit(C)

18 Algorithms

```
    AddInverse( C,
      function( alpha )
        return Inverse( UnderlyingCell( alpha ) ) / CapCategory( alpha );
    end );

    c := ConcreteCategory( L );

    C!.ConcreteCategoryRecord := c;

    objects := List( c.objects, FinSet );

    SetSetOfObjects( C, List( objects, o -> o / C ) );

    SetSetOfGeneratingMorphisms( C, List( c.generators, g -> ConvertToMapOfFinSets( obje

    Finalize( C );

    return C;

end );

##
InstallMethod( Algebroid,
        "for a homalg ring and a finite category",
        [ IsHomalgRing and IsCommutative, IsFiniteConcreteCategory ],

  function( k, C )
    local objects, gmorphisms, q, kq, relEndo, A, F, vertices, rel,
          func, st, s, t, homST, list, p, pos;

    objects := SetOfObjects( C );
    gmorphisms := SetOfGeneratingMorphisms( C );
    q := RightQuiverFromConcreteCategory( C );
    kq := PathAlgebra( k, q );
    relEndo := RelationsOfEndomorphisms( k, C );
    A := Algebroid( kq, relEndo );
    kq := UnderlyingQuiverAlgebra( A );
    F := CapFunctor( A, objects, gmorphisms, C );
```

```

    vertices := List( SetOfObjects(A), UnderlyingVertex );

    rel := [];
    func :=
      function( p, l )
        return ForAny( l, p1->
                       IsCongruentForMorphisms(
                               ApplyToQuiverAlgebraElement( F, p ),
                               ApplyToQuiverAlgebraElement( F, p1 ) )
                       );
    end;

    for st in Cartesian(vertices,vertices) do
        s := st[1];
        t := st[2];
        if s = t then
            continue;
        fi;
        homST := BasisPathsBetweenVertices( kq, s, t );
        homST := List( homST, p -> PathAsAlgebraElement( kq, p ) );

        list := [];

        for p in homST do
            pos := PositionProperty( list, l->func(p,l) );
            if IsInt(pos) then
                Add( list[pos], p );
            else
                Add( list, [p] );
            fi;
        od;
        list := List( list, l-> List( l, p -> p!.representative ) );
        Append( rel, list );
    od;

    rel := Filtered( rel, l -> Length(l)>1 );
    rel := List( rel, l -> List( l{[ 2 .. Length(l) ]}, p -> l[1]-p ) );
    rel := Flat( rel );
    rel := Concatenation( relEndo, rel );

    kq := PathAlgebra( kq ) / rel;

    kq := PathAlgebra( kq ) / GroebnerBasis( IdealOfQuotient( kq ) );
```

We want the endomorphism relations so that the path algebra is finite-dimensional and we get a finite Gröbner basis.

Proof that algorithm is correct Proof that it terminates.

Wir haben BasisOfExternalHom benutzt um Decompose in CAP umzusetzen um EmbeddingOfSubRepresentation umzusetzen um WeakDirectSumDecomposition umzusetzen.

Notes

[1]Some authors use maps t, h for *tail* and *head* instead of source and target, defining the arrows to go from the tail to the head. This use of t as the starting point instead of the end target as in our definition can lead to some confusion.

Algorithm 1: `RightQuiverFromConcreteCategory`

Input : a finite concrete category C with n objects
Output : the right quiver $q(n)$

let Obj be the set of objects of C;
let $n := Length(Obj)$;
let $gMor$ be the set of generating morphisms of C;
let A be the empty set and let $i := 1$;
foreach *morphism mor in gMor* **do**
    let $A_{i,1}$ be the position of $Source(mor)$ in Obj;
    let $A_{i,2}$ be the position of $Range(mor)$ in Obj;
    let $i := i + 1$;
end
let q be the right quiver with vertices $\{1, \dots, n\}$ and arrows A.
return q;

Algorithm 2: `RelationsOfEndomorphisms`

Input: a commutative ring k and a finite concrete category C
Output: the endomorphism relations of the category C

let $q :=$ `RightQuiverFromConcreteCategory`(C);
let kq be the path algebra generated by k and q;
let $gMor$ be the set of generating morphisms of C;
let $A := Arrows(q)$;
let $relsEndo$ be the empty set;
foreach $i = 1, \dots, Length(gMor)$ **do**
    let $mor := gMor_i$ **if** *mor is not an endomorphism* **then**
        continue;
    end
    let $m := 0$ and let $powers$ be the empty set;
    let $foundEqual$ be false;
    while $mor^m \notin powers$ **do**
        let $n := 1$;
        while $\neg foundEqual$ **do**
            if $mor^{(m+n)} = mor^m$ **then**
                Add the relation $kq.(A_i)^{(m+n)} - kq.(A_i)^m$ to relsEndo;
                foundEqual := true;
            end
            n := n+1;
        end
        Add mor^m to powers;
        m := m+1;
    end
end
return *relsEndo*;

References

[1] https://web.northeastern.edu/martsinkovsky/p/Parnu2019/slides-facchini.pdf