Willoughby Seago

Notes from

Hopf Algebras

January 16th, 2024

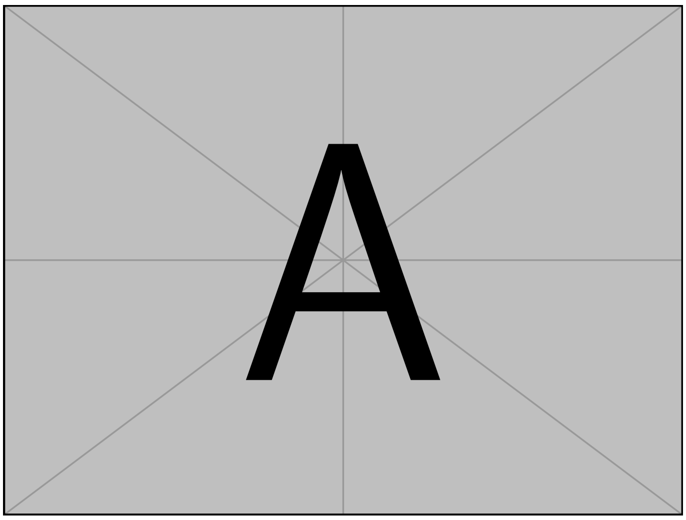
UNIVERSITY OF GLASGOW

Hopf Algebras

Willoughby Seago

January 16th, 2024

These are my notes from the SMSTC course *Hopf Algebras* taught by Dr Andrew Baker. For category theory details I have been referring to Riehl's *Category Theory in Context*. These notes were last updated at 17:50 on January 25, 2025.



Chapters

Page

Contents

Page

One

Category Theory

We will make use of a lot of category theory throughout the course. It is expected that people are familiar with the basic notions of category theory, some of which we recap here.

1.1 Initial and Terminal Objects

Definition 1.1.1 Let C be a category.

- An object, $t \in \mathsf{C}$, is **terminal** if for all $a \in \mathsf{C}$ the set $\mathsf{C}(a, t)$ contains exactly one element.
- An object, $i \in \mathsf{C}$, is **initial** if for all $a \in \mathsf{C}$ the set $\mathsf{C}(i, a)$ contains exactly one element.

An object that is both initial and terminal is called a **null** or **zero object**.

Lemma 1.1.2 Initial and terminal objects are unique up to unique isomorhpism.

Notation 1.1.3 We denote the initial object in a category by $\mathbf{0}$ and the terminal object by $\mathbf{1}$.

Example 1.1.4

- The category of sets, Set, has $\mathbf{0} = \varnothing$ and $\mathbf{1} = \{\bullet\}$ (that is, any singleton set is terminal).
- The category of pointe sets, $\mathsf{Set}_{\bullet}$, has $\mathbf{0} = \mathbf{1} = \{\bullet\}$ where $\{\bullet\}$ is the pointed set with $\bullet$ as its distinguished element.
- The category of groups, Grp, or abelian groups, Ab, has $\mathbf{0} = \mathbf{1} = \{e\}$, that is the trivial group is a null object.

- The category of rings[a], Ring, has $\mathbf{0} = \mathbb{Z}$, since a ring homomorhpism[b] $\varphi : \mathbb{Z} \to \mathbb{R}$ is uniquely determined by $\varphi(1) = 1_R$ since then $\varphi(n) = \varphi(n1) = n\varphi(1) = n1_R$. If we allow $0 = 1$ then the terminal object is the trivial ring, $\mathbf{1} = \{0\}$. If we do not allow $0 = 1$ then Ring has no terminal object[c].
- The category of non-unital rings, Rng, has $\mathbf{1} = \{0\}$, and no initial object.
- **Abelian categories** are a special class of categories with certain properties. One of these properties is that they have a null object. Examples of abelian categories include Ab, $\mathsf{Vect}_{\mathbb{k}}$, and more generally R-Mod.

[a] which we assume are unital
[b] which we assume preserves the unit
[c] In this course we generally assume $0 \neq 1$. I disagree with this choice.

1.2 Products and Coproducts

Definition 1.2.1 — Product Let C be a category. A set of morphisms $\{p_i : c \to c_i \mid i \in I\}$, for some indexing set, I, is a **product** of the p_i (although usually we refer to it as a product of the c_i) if given any set of morphisms $\{f_i : d \to c_i \mid i \in I\}$ there exists a unique morphism $f : d \to c$ such that $f_i = p_i f$ for all $i \in I$.
If $I = \varnothing$ then we define the product to be the terminal object, if it exists.

Definition 1.2.2 Let C be a category. A set of morphisms $\{j_i : c_i \to c \mid i \in I\}$, for some indexing set, I, is a **coproduct** of the j_i if given any set of morphisms $\{g_i : c_i \to d \mid i \in I\}$ there exists a unique morphism $g : c \to d$ such that $g_i = g j_i$ for all $i \in I$.
If $I = \varnothing$ then we define the product to be the initial object, if it exists.

Lemma 1.2.3 Products and coproducts are unique up to unique isomorphism.

We can express the definition of the (co)product as the existence of a morphism making a certain I-indexed family of diagrams commute:

$$\begin{tikzcd} d \arrow[d, dashed, "\exists! f"'] \arrow[dr, "f_i"] & \\ c \arrow[r, "p_i"'] & c_i \end{tikzcd} \qquad \left(\begin{tikzcd} c \arrow[r, "j_i"] \arrow[d, dashed, "g"'] & c_i \arrow[dl, "g_i"] \\ d & \end{tikzcd}\right). \tag{1.2.4}$$

When we're dealing with binary (co)products ($I = \{1, 2\}$) we usually combine the two triangles into the commuting diagram

$$
\begin{tikzcd}
 & d \arrow[dl, "f_1"'] \arrow[d, dashed, "\exists! f"] \arrow[dr, "f_2"] & \\
c_1 & c \arrow[l, "p_1"] \arrow[r, "p_2"'] & c_2
\end{tikzcd}
\qquad
\left(
\begin{tikzcd}
c_1 \arrow[r, "j_1"] \arrow[dr, "g_1"'] & c \arrow[d, dashed, "\exists! g"] & c_2 \arrow[l, "j_2"'] \arrow[dl, "g_2"] \\
 & d &
\end{tikzcd}
\right).
\tag{1.2.5}
$$

Notation 1.2.6 The (co)product is typically denoted in terms of the objects, with the projections (inclusions) left implicit. We will denote binary products by $\times$, other notations include Π and $\otimes$, We will denote coproducts by $+$, other notations include $\amalg$, $\oplus$ and $\sqcup$.
We'll also use $\prod$ and $\coprod$ to denote products over arbitrary families.

Note that there is ambiguity in the above notation in that I is not assumed to be ordered, so $c_1 \times c_2$ and $c_2 \times c_1$ are both valid notations for the product of c_1 and c_2. Formally these objects may be different, but they will be isomorphic with a unique isomorphism between them so it doesn't really matter.

Products and coproducts are functorial in their variables. That is, if we have two products $\{p_i : c \to c_i \mid i \in I\}$ and $q_i : d \to d_i \mid i \in I$ and morphisms $f_i : c_i \to d_i$ then there is a unique morphism $h : c \to d$ such that for all $i \in I$ the diagram

$$
\begin{tikzcd}
c \arrow[r, "h"] \arrow[d, "p_i"'] & d \arrow[d, "q_i"] \\
c_i \arrow[r, "f_i"'] & d_i
\end{tikzcd}
\tag{1.2.7}
$$

commutes. We call h the product of the f_i, and denote it $h = \prod_{i\in I} f_i$. A dual result holds for the coproduct.

Consider the special case of the above where $d_i = c_{\sigma^{-1}(i)}$ for some bijection $\sigma : I \to I$. This uniquely defines an isomorphism $\mathrm{T}_\sigma : \prod_I c_i \to \prod_I c_{\sigma^{-1}(i)}$. The most important case of this is when $I = \{1, 2\}$ and $\sigma(0) = 1$ and $\sigma(1) = 0$, in which case we have the isomorphism

$$
\mathrm{T} = \mathrm{T}_{(1\,2)} : c_1 \times c_2 \to c_2 \times c_1. \tag{1.2.8}
$$

Note that T actually depends on c_1 and c_2, so we should probably call it T_{c_1,c_2}. However, for any given objects c_1 and c_2 for which the product $c_1 \times c_2$ exists there is such a morphism, so we typically drop the objects from the notation letting context inform us of which T we're using. The more formal justification for this is that the functor $(c_1, c_2) \mapsto c_1 \times c_2$ is naturally isomorphic to the functor $(c_1, c_2) \mapsto c_2 \times c_1$, and T_{c_1,c_2} is the component of this natural isomorphism at $(c_1, c_2) \in \mathsf{C} \times \mathsf{C}$.

Note that the composite

$$
c_1 \times c_2 \xrightarrow{\mathrm{T}_{c_1,c_2}} c_2 \times c_1 \xrightarrow{\mathrm{T}_{c_2,c_1}} c_1 \times c_2 \tag{1.2.9}
$$

is actually the identity. That is,

$$
\mathrm{T}_{c_2,c_1} \circ \mathrm{T}_{c_1,c_2} = \mathrm{id}_{\mathsf{C}}, \tag{1.2.10}
$$

or more snappily, $\mathrm{T}^2 = \mathrm{id}_{\mathsf{C}}$.

Example 1.2.11

- In Set the product is the Cartesian product, $\times$, and the coproduct is the disjoin union, $\sqcup$.
- In $\mathsf{Set}_\bullet$ the product $(X,x)\times(Y,y)$ is the pointed set $(X\times Y,(x,y))$ where $X\times Y$ is the Cartesian product. The coproduct $(X,x)+(Y,y)$ is the pointed set $((X\sqcap Y)/\sim,[x])$ where $\sim$ is the equivalence relation identifying x and y in the disjoint union with each other, and $[x]$ is the equivalence class that x and y end up in under this relation. More intuitively, the coproduct is just the disjoint union where we identify the base points of the original sets with each other.
- In Grp the product is the Cartesian product (also called the direct product) of the underlying sets with the operation defined pointwise. The coproduct is the free product, $G * H$, which intuitively consists of words of the form $g_1h_1g_2h_2\cdots g_nh_n$ with $g_i\in G$ and $h_i\in H$ (and only g_1 or h_n allowed to be identities).
- In Ring the product is the Cartesian product and the coproduct is a free product defined similarly to the free product of groups, but also allowing us to add elements together as well as forming products.
- In Ab any finite product or coproduct is given by the direct product (also called the direct sum).
- In Top the product and coproduct are the Cartesian product and disjoint union of the underlying sets equipped with the appropriate topologies which may be characterised as being the coarsest topologies such that the projections and inclusions are continuous.
- In an **abelian category** finite products and coproducts coincide by definition. Thus, in $\mathsf{Vect}_{\Bbbk}$ or R-Mod all finite products and coproducts are given by the direct sum.

Proposition 1.2.12 Let C be a category in which all binary products exist and there is a terminal object, $\mathbf{1}$. Then all finite products exist and for all $c\in\mathsf{C}$ we have

$$\mathbf{1}\times c\cong c\cong c\times\mathbf{1}. \tag{1.2.13}$$

Let C be a category in which all binary coproducts exist and there is an initial object, $\mathbf{0}$. Then all finite coproducts exist and for all $c\in\mathsf{C}$ we have

$$\mathbf{0}+c\cong c\cong c+\mathbf{0}. \tag{1.2.14}$$

1.3 Monoids and Comonoids

I'm going to take a slightly different approach here and not define monoids in an arbitrary category with products and terminal objects, but instead move straight to monoidal categories, which subsume these cases.

Definition 1.3.1 — Monoidal Category A **monoidal category**, $(\mathsf{C}, \otimes, I, \alpha, \lambda, \rho)$, is a category, C, with a functor

$$- \otimes - \colon \mathsf{C} \times \mathsf{C} \to \mathsf{C}, \tag{1.3.2}$$

object $I \in \mathsf{C}$, and natural transformations

$$\alpha \colon (- \otimes -) \otimes - \Rightarrow - \otimes (- \otimes -); \tag{1.3.3}$$
$$\lambda \colon I \otimes - \Rightarrow -; \tag{1.3.4}$$
$$\rho \colon -\otimes I \Rightarrow -. \tag{1.3.5}$$

This data is subject to the following:

- For all $a, b, c, d \in \mathsf{C}$ the diagram

$$\begin{tikzcd} & (ab)(cd) \arrow[r, "\alpha_{a,b,cd}"] & a(b(cd)) \\ ((ab)c)d \arrow[ur, "\alpha_{ab,c,d}"] \arrow[dr, "\alpha_{a,b,c}\otimes \mathrm{id}_d"'] & & \\ & (a(bc))d \arrow[r, "\alpha_{a,bc,d}"'] & a((bc)d) \arrow[uu, "\mathrm{id}_a \otimes \alpha_{b,c,d}"'] \end{tikzcd} \tag{1.3.6}$$

  commutes where we use the shorthand $ab = a \otimes b$.

- For all $a, b \in \mathsf{C}$ the diagram

$$\begin{tikzcd} (a \otimes I) \otimes b \arrow[rr, "\alpha_{a,I,b}"] \arrow[dr, "\rho_a \otimes \mathrm{id}_b"'] & & a \otimes (I \otimes b) \arrow[dl, "\mathrm{id}_a \otimes \lambda_b"] \\ & a \otimes b & \end{tikzcd} \tag{1.3.7}$$

  commutes.

Definition 1.3.8 — Braided Monoidal Category A **braided monoidal category**, $(\mathsf{C}, \otimes, I, \alpha, \lambda, \rho, \gamma)$, is a monoidal category $(\mathsf{C}, \otimes, I, \alpha, \lambda, \rho)$ equipped with a natural transformation, γ, with components

$$\gamma_{a,b} \colon a \otimes b \Rightarrow b \otimes a. \tag{1.3.9}$$

This is subject to the condition that the diagrams

$$
\begin{tikzcd}
 & a(bc) \arrow[r, "\gamma_{a,bc}"] & (bc)a \arrow[dr, "\alpha_{b,c,a}"] & \\
(ab)c \arrow[ur, "\alpha_{a,b,c}"] \arrow[dr, "\gamma_{a,b}\otimes\mathrm{id}_c"'] & & & b(ca) \\
 & (ba)c \arrow[r, "\alpha_{b,a,c}"'] & b(ac) \arrow[ur, "\mathrm{id}_b\otimes\gamma_{a,c}"'] &
\end{tikzcd}
\tag{1.3.10}
$$

and

$$
\begin{tikzcd}
 & (ab)c \arrow[r, "\gamma_{ab,c}"] & c(ab) \arrow[dr, "\alpha^{-1}_{c,a,b}"] & \\
a(bc) \arrow[ur, "\alpha^{-1}_{a,b,c}"] \arrow[dr, "\mathrm{id}_a\otimes\gamma_{b,c}"'] & & & (ca)b \\
 & a(cb) \arrow[r, "\alpha^{-1}_{a,c,b}"'] & (ac)b \arrow[ur, "\gamma_{a,c}\otimes\mathrm{id}_b"'] &
\end{tikzcd}
\tag{1.3.11}
$$

commute. Again, writing $ab = a \otimes b$ as shorthand.

Definition 1.3.12 — Symmetric Monoidal Category A **symmetric monoidal category** is a braided monoidal category for which

$$\gamma_{b,a} \circ \gamma_{a,b} = \mathrm{id}_{a\otimes b} \tag{1.3.13}$$

for all $a, b \in \mathsf{C}$.

The natural isomorphisms in these definitions are called the coherence morphisms. They all model a particular property mirroring a property of a monoid:

- α is the associator, and it means that the product (which is not necessarily a categorical product) $\otimes$ is associative up to natural isomorphism.
- λ and ρ are the left and right unitors, and they mean that I acts as an identity element for the product $\otimes$, again up to natural isomorphism.
- γ (when it exists) is the braiding or symmetry of the category, and it means that the product $\otimes$ is commutative up to natural isomorphism.

Example 1.3.14 The following are all monoidal categories, the coherence morphisms are left out of the notation (as is standard) and they can usually be worked out from context.

- $(\mathsf{Set}, \times, \{\bullet\})$: The coherence maps are
  - $\alpha_{a,b,c}((x, y), z) = (x, (y, z))$;

  - $\lambda_a(\bullet, x) = x$;
  - $\rho_a(x, \bullet) = x$;
  - $\gamma_{a,b}(x, y) = (y, x)$;

- $(\mathsf{Set}, \sqcup, \varnothing)$:
- $(\mathsf{Vect}_{\Bbbk}, \otimes, \Bbbk)$;
- $(R\text{-}\mathsf{Mod}, \otimes_R, R)$;
- $(\mathsf{C}, \times, \mathbf{1})$ where C is any category with all binary products and a terminal object.

Note that multiple monoidal structures can exist on the same underlying category. All of these are symmetric, as most examples in common practice are. An example of a non-symmetric monoidal category is $[\mathsf{C}, \mathsf{C}]$, the category of endofunctors, $\mathsf{C} \to \mathsf{C}$, with the monoidal product given by composition and the unit object being the identity functor, id_{C}.

We are now ready to give the definition of a monoid in a monoidal category in its full glory.

Definition 1.3.15 — Monoid Let C be a monoidal category. A **monoid**, (m, μ, η), is an object, $m \in \mathsf{C}$, equipped with morphisms

$$\mu\colon\ m \otimes m \to m, \qquad \text{and} \qquad \iota\colon\ I \to m, \tag{1.3.16}$$

called **multiplication** and **unit** respectively, such that the diagrams

$$\begin{tikzcd}
 & m \otimes (m \otimes m) \arrow[r, "\mathrm{id}_m \otimes \mu"] & m \otimes m \arrow[dd, "\mu"] \\
(m \otimes m) \otimes m \arrow[ur, "\alpha_{m,m,m}"] \arrow[dr, "\mu \otimes \mathrm{id}_m"'] & & \\
 & m \otimes m \arrow[r, "\mu"'] & m
\end{tikzcd} \tag{1.3.17}$$

and

$$\begin{tikzcd}
I \otimes m \arrow[r, "\eta \otimes \mathrm{id}_m"] \arrow[dr, "\lambda_m"'] & m \otimes m \arrow[d, "\mu"] & m \otimes I \arrow[l, "\mathrm{id}_m \otimes \eta"'] \arrow[dl, "\rho_m"] \\
 & m &
\end{tikzcd} \tag{1.3.18}$$

commute.

If C is symmetric monoidal category with braiding γ then (m, μ, η) is a **commutative monoid** if

$$\begin{tikzcd}
m \otimes m \arrow[rr, "\gamma_{m,m}"] \arrow[dr, "\mu"'] & & m \otimes m \arrow[dl, "\mu"] \\
 & m &
\end{tikzcd} \tag{1.3.19}$$

commutes, that is, $\mu \circ \gamma_{m,m} = \mu$.

Example 1.3.20

- A monoid, (M, μ, η), in the monoidal category Set (under the Cartesian product, which we assume to be the monoidal structure of Set unless stated otherwise) is just a monoid in the usual sense. The map $\mu : M \times M \to M$ is just the multiplication map on M, and $\eta : \{\bullet\} \to M$ picks out an element of M, namely $\eta(\bullet)$.

  The first diagram states that the multiplication in M is associative, which we can see by starting at $(M \times M) \times M$ and tracing both directions around the diagram, taking $x, y, z \in M$ going the top path gives

$$\mu \circ (\mathrm{id}_m \times \mu) \circ \alpha_{m,m,m}((x,y),z) = \mu \circ (\mathrm{id}_m \times \mu)(x,(y,z)) \tag{1.3.21}$$
$$= \mu(\mathrm{id}_m(x), \mu(y,z)) \tag{1.3.22}$$
$$= \mu(x, \mu(y,z)) \tag{1.3.23}$$

  and going along the bottom path gives

$$\mu \circ (\mu \times \mathrm{id}_m)((x,y),z) = \mu(\mu(x,y), \mathrm{id}_m(z)) \tag{1.3.24}$$
$$= \mu(\mu(x,y),z). \tag{1.3.25}$$

  Writing $\mu(a,b) = ab$ these two results become $x(yz)$ and $(xy)z$. Demanding that they are equal is thus associativity of multiplication.

  The second diagram asserts that the distinguished element, $e = \eta(\bullet) \in M$, is the identity for this multiplication map. The left triangle gives us

$$\mu \circ (\eta \times \mathrm{id}_m)(\bullet, x) = \mu(e,x) = ex \quad \text{and} \quad \lambda_m(\bullet, x) = x, \tag{1.3.26}$$

  so imposing equality gives us that $ex = x$, so e is a left identity. Similarly, the right triangle gives us that e is a right identity.

  The commutativity condition, assuming it holds, states that $\mu \circ \gamma_{m,m}(x,y) = \mu(y,x) = yx$ and $\mu(x,y) = xy$ agree, so it's just stating that the multiplication in this monoid is commutative.

- A monoid in the category of monoids, $(\mathsf{Mon}, \times, \{\bullet\})$, is a commutative monoid, this follows from the Eckmann–Hilton argument[a], which demonstrates that if there are two compatible monoidal structures then they actually agree and must be commutative. Similarly, a monoid in the category of groups, Grp, is an abelian group.

- A monoid in the category of topological spaces, $(\mathsf{Top}, \times, \{\bullet\})$, is a topological monoid, that is a monoid for which the multiplication and unit maps are continuous. A monoid in the category of smooth manifolds, $(\mathsf{Man}, \times, \{\bullet\})$, is a Lie monoid (a Lie group without the condition that inverses exist).

- A monoid in the category of abelian groups, $(\mathsf{Ab}, \otimes_{\mathbb{Z}}, \mathbb{Z})$, is a ring, the group operation providing the addition and the monoid multiplication providing the multiplication. Distributivity of multiplication over addition is witnessed by the requirement that μ is a group homomorphism, so

$$\mu(x, y + z) = \mu(x, y) + \mu(x, z) \tag{1.3.27}$$

  which gives

$$x(y + z) = xy + xz \tag{1.3.28}$$

  and similar for distributivity on the right.

- For a commutative[b] ring, R, a monoid in the category of R-modules, $(R\text{-}\mathsf{Mod}, \otimes_R, R)$, is an R-algebra. The monoid multiplication provides the multiplication of the algebra, which is commutative exactly when this is a commutative monoid. Note that $R = \Bbbk$ is a special case of this, in which case the monoidal category is $\mathsf{Vect}_{\Bbbk}$.

- A monoid in the monoidal category of endofunctors, $[\mathsf{C}, \mathsf{C}]$, is a monad.

[a] Suppose that we have a set, M, equipped with two binary operations, $\circ$ and $\diamond$ with associated units $1_\circ$ and $1_\diamond$. These are compatible if $(a \circ b) \diamond (c \circ d) = (a \diamond c) \circ (b \diamond d)$ (a result known as the interchange law, note that this is true of vertical and horizontal composition of natural transformations). Then using these properties we have $1_\circ = 1_\circ \circ 1_\circ = (1_\diamond \diamond 1_\circ) \circ (1_\circ \diamond 1_\diamond) = (1_\diamond \circ 1_\circ) \diamond (1_\circ \circ 1_\diamond) = 1_\diamond \diamond 1_\diamond = 1_\diamond$, so the units coincide and we write $1 = 1_\circ = 1_\diamond$. Then for $x, y \in M$ we have $a \circ b = (1 \diamond a) \circ (b \diamond 1) = (1 \circ b) \diamond (a \circ 1) = b \diamond a = (b \circ 1) \diamond (1 \circ a) = (b \diamond 1) \circ (1 \diamond a) = b \circ a$, thus $\circ$ is commutative, and from this we see that $b \circ a = b \diamond a$, so $\diamond$ and $\circ$ coincide. Note that one can also prove associativity of these operations: $(ab)c = (ab)(1c) = (a1)(bc) = a(bc)$, where now that we know that the two multiplications are the same we just use juxtaposition.

[b] commutativity is required for the tensor product of R-modules to be an R-module, we could instead look at a category of bimodules for a slightly more general result.

So far all we've achieved is a somewhat complex unification of many similar algebraic structures. The nice thing is that now we've defined monoids in terms of commutative diagrams as well as picking them up and placing them in other categories we can also dualise the definition.

Definition 1.3.29 — Comonoid Let C be a monoidal category. A comonoid, (c, Δ, ε), is an object, $c \in \mathsf{C}$, equipped with morphisms

$$\Delta : c \to c \otimes c, \qquad \text{and} \qquad \varepsilon : c \to I, \tag{1.3.30}$$

called **comultiplication** and **counit** respectively, such that the diagrams

$$\begin{tikzcd} & c \otimes (c \otimes c) \arrow[dl, "\alpha^{-1}_{c,c,c}"'] & c \otimes c \arrow[l, "\mathrm{id}_c \otimes \Delta"'] \\ (c \otimes c) \otimes c & & \\ & c \otimes c \arrow[ul, "\Delta \otimes \mathrm{id}_c"] & c \arrow[l, "\Delta"] \arrow[uu, "\Delta"'] \end{tikzcd} \tag{1.3.31}$$

and

$$\begin{tikzcd} I \otimes c & c \otimes c \arrow[l, "\varepsilon \otimes \mathrm{id}_c"'] \arrow[r, "\mathrm{id}_c \otimes \varepsilon"] & c \otimes I \\ & c \arrow[ul, "\lambda_c^{-1}"] \arrow[u, "\Delta"] \arrow[ur, "\rho_c^{-1}"'] & \end{tikzcd} \tag{1.3.32}$$

commute.
If C is a symmetric monoidal category with braiding γ then (c, Δ, ε) is a **cocommutative comonoid** if

$$\begin{tikzcd} c \otimes c & & c \otimes c \arrow[ll, "\gamma_{c,c}"'] \\ & c \arrow[ul, "\Delta"] \arrow[ur, "\Delta"'] & \end{tikzcd} \tag{1.3.33}$$

commutes, that is, $\gamma_{c,c} \circ \Delta = \Delta$.

Comonoids typically aren't as familiar as monoids, but nonetheless they're important, in particular Hopf algebras are both monoids and comonoids (in a compatible way with one other piece of data).

Example 1.3.34

- In $(\mathsf{Set}, \times, \{\bullet\})$ any set, X, has a unique comonoid structure given by declaring that $\Delta(x) = (x, x) \in X \times X$ and $\varepsilon(x) = \bullet$ is the unique element of the singleton set, $I = \{\bullet\}$. This makes comonoids particularly boring in Set, which explains why they tend not to be so familiar. Taking $X = \mathbb{R}$ and picturing $\mathbb{R} \times \mathbb{R}$ as the plane the image of Δ is thus the diagonal line $y = x$, which is why $x \mapsto (x, x)$ is sometimes called the diagonal map, and explains the choice of Δ (D for Diagonal) as the notation for the comultiplication.

- In $(\mathsf{Set}, \sqcup, \varnothing)$ the only comonoid is $\varnothing$, since to define a comonoid structure on some non-empty set, X, we'd have to have a map $\varepsilon : X \to \varnothing$, and no such maps exist.

- In $\mathsf{Vect}_{\Bbbk}$ any $\Bbbk$-vector space, V, may be equipped with a comonoid structure as follows. First, pick a basis, $\{e_i\}$. Then define comultiplication by $\Delta(e_i) = \sum_i e_i \otimes e_i$, and the counit by $e_i \mapsto 1 \in \Bbbk$, both extended by linearity to the whole space.

- For a commutative ring, R, a comonoid in $(R\text{-}\mathsf{Mod}, \otimes_R, R)$ is called a coalgebra (more on these later). A specific example is the polynomial ring $R[X]$, which is a comonoid when we define the comultiplication to be $\Delta(X) = X \otimes X$ and the counit to be $\varepsilon(X) = 1 \in R$.
- The monoidal category $(\mathsf{Top}_\bullet, \vee, \{\bullet\})$ of based topological spaces under the wedge product (union and then identify the base points) has the sphere, S^n, as an *almost* comonoid. There's a map $S^n \vee S^n \to S^n$ that almost works as the comultiplication. The only problem is that coassociativity only holds up to homotopy. We can get a proper comonoid by instead considering $(\mathsf{hTop}_\bullet, \vee, \{\bullet\})$, the monoidal category of based topological spaces up to homotopy.
- A comonoid in the monoidal category of endofunctors, $[\mathsf{C}, \mathsf{C}]$, is a comonad.

1.3.1 String Diagrams

When working with monoidal categories a common notation is **string diagrams**. These are used to represent expressions like $f \otimes (g \circ h)$ with $h\colon a \to b$, $g\colon d \to e$ and $h\colon c \to d$. The idea is that an object is represented by its identity morphism, which we draw as a string, labelled with its object if necessary:

$\mathrm{id}_a =$ 

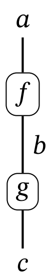
(1.3.35)

An arbitrary morphism, $f\colon a \to b$, is then denoted with a box, we read the diagram from top to bottom[1], with the incoming wire being the domain, a, and the outgoing wire the codomain, b:

$f =$ 

(1.3.36)

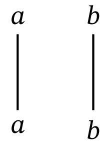
Composition of morphisms is then denoted by chaining together strings, so if $f\colon a \to b$ and $g\colon b \to c$ we write $g \circ f$ as

$g \circ f =$ 

(1.3.37)

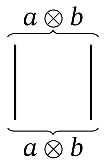
[1] conventions vary here, some read bottom to top, some left to right

The monoidal product is written by just placing the two wires next to each other, so $a \otimes b$ is represented by the monoidal product of the corresponding morphisms $\mathrm{id}_a \otimes \mathrm{id}_b$:

$\mathrm{id}_a \otimes \mathrm{id}_b =$ 

(1.3.38)

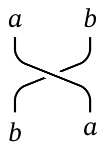
Note that since $\otimes$ is functorial we have $\mathrm{id}_a \otimes \mathrm{id}_b = \mathrm{id}_{a \otimes b}$, and so we can interpret this diagram as

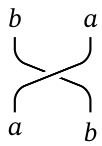
$\mathrm{id}_{a \otimes b} =$ 

(1.3.39)

The unit object of the monoidal structure is usually not drawn, since we can always apply an appropriate unitor isomorphism to remove it. In fact, none of the structure morphisms of the monoidal structure are drawn. Instead these morphisms are implicit whenever they're needed to make the morphism that the diagram represents type check. This is basically like not writing brackets and unit objects when we compute products in a group, even though we could (and maybe formally should) in certain places.

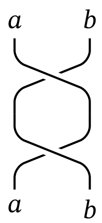
If we have a braided monoidal category then we draw the braiding, a morphism $\sigma_{a,b} : a \otimes b \to b \otimes a$, and its inverse, $\sigma_{a,b}^{-1} : b \otimes a \to a \otimes b$, as crossing wires:

$\sigma_{a,b} =$ 

$\sigma_{a,b}^{-1} =$ 

(1.3.40)

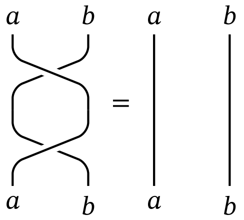
The fact that in a *braided* monoidal category we don't, in general, have $\sigma_{b,a} \circ \sigma_{a,b} = \mathrm{id}_{a \otimes b}$ is expressed by the fact that disentangle the following diagram to get Equation (1.3.38) without passing one wire through the other, note that we assume the end points of wires are fixed, and we don't allow moves like passing over the end of the wires, imagine everything is constrained to a box with the wires attached to opposite sides of the box:

$\sigma_{b,a} \circ \sigma_{a,b} =$ 

(1.3.41)

Conversely, we can disentangle the following to get the identity:

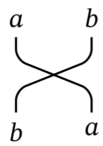
$$\sigma_{a,b}^{-1} \circ \sigma_{a,b} =$$



(1.3.42)

For a symmetric monoidal category we do have that $\sigma_{a,b}^{-1} = \sigma_{b,a}$. This is represented in the string diagrams by allowing strings to pass through each other, and because of this we don't need to draw one string passing over the other, so we draw the symmetry as

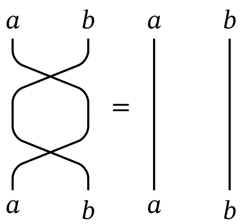
$$\sigma_{a,b} =$$



$$= \sigma_{b,a}^{-1}. \tag{1.3.43}$$

Then the fact that $\sigma_{b,a} \circ \sigma_{a,b} = \mathrm{id}_{a \otimes b}$ is expressed by

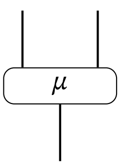
$$\sigma_{b,a} \circ \sigma_{a,b} =$$



(1.3.44)

Given a monoid object, (m, μ, η), we have morphisms $\mu : m \otimes m \to m$ and $\eta : I \to m$. We represent these as

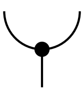
$$\mu =$$



(1.3.45)

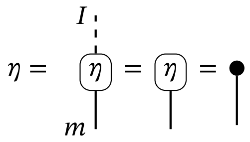
Note that all the wires here are m, as we can deduce from the domain and codomain of μ, so we'll stop labelling wires when it's easy to work out what they are. The interpretation of this diagram is that two copies of m enter from the top, then we apply the monoid multiplication to combine them into one. It's common to replace the μ box with just a dot, just to make the diagrams slightly less busy:

$$\mu =$$



(1.3.46)

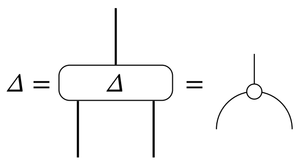
The unit map, $\eta : I \to m$ starts at the monoidal unit and ends at m, so we can draw this as a map



(1.3.47)

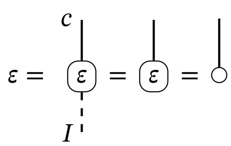
Here we use a dashed line to denote the unit object, or most the time we just don't draw it. Again, most of the time we just replace η with a dot as well.

If instead we have a comonoid object, (c, Δ, ε), then we simply flip all of the diagrams upside down, this is just what happens to diagrams when we take duals. This gives us the comonoid maps, $\Delta : c \to c \otimes c$ and $\varepsilon : c \to I$, expressed as the string diagrams



(1.3.48)

and



(1.3.49)

1.4 Adjunctions and Monads

We now (briefly) introduce two important concepts of category theory. The first is adjunction, the idea that often functors come in opposing pairs. It's usually too restrictive to require that a pair of functors are inverses[2], and instead the natural requirement is that they compose to something that is naturally isomorphic to the identity. This gives rise to the notion of an adjunction. There are several equivalent definitions of adjunctions. The one we give here is perhaps the most standard.

[2] As a general rule imposing equalities on functors is usually not a good idea.

Definition 1.4.1 — Adjoint Functors Let C and D be categories. Let $L : \mathsf{D} \to \mathsf{C}$ and $R : \mathsf{C} \to \mathsf{D}$ be functors. We say that L is **left adjoint** to R, or R is **right adjoint** to L, or write $L \dashv R$ ($R \vdash L$) if there is a natural isomorphism

$$\mathsf{C}(L-, -) \cong \mathsf{D}(-, R-). \tag{1.4.2}$$

Note that the functors $\mathsf{C}(L-, -)$ and $\mathsf{D}(-, R-)$ in this definition have domain $\mathsf{D} \times \mathsf{C}$, and codomain[3] Set.

[3] assuming locally small categories

Lemma 1.4.3 Left and right adjoints are unique up to unique natural isomorphism.

Example 1.4.4

- A **forgetful functor** is a functor $\mathsf{C} \to \mathsf{D}$ which "forgets" some structure in C to produce an object of D. For example, there is a functor $U\colon \mathsf{Grp} \to \mathsf{Set}$ that sends a group to its underlying set, or a functor $U\colon \mathsf{Vect}_{\Bbbk} \to \mathsf{Ab}$ which forgets about scalar multiplication and sends a vector space to its underlying set of vectors equipped with addition as the group operation. Forgetful functors tend to have left adjoints, $\mathsf{D} \to \mathsf{C}$, which are then termed **free functors**. The idea of these functors is that they produce an object of C from an object of D in the most general way possible. For example, the left adjoint to the forgetful functor $U\colon \mathsf{Grp} \to \mathsf{Set}$ is the free group functor, $F\colon \mathsf{Set} \to \mathsf{Grp}$ which constructs the free group on a given set. Similarly, the forgetful functor $U\colon \mathsf{Vect}_{\Bbbk} \to \mathsf{Set}$, which sends a vector space to its underlying set of vectors, is right adjoint to the free vector space functor, $F\colon \mathsf{Set} \to \mathsf{Vect}_{\Bbbk}$, which sends a set to the vector space for which that set is a basis.

  Take this last example, the adjunction means we have an isomorphism

  $$\mathsf{Vect}_{\Bbbk}(F-,-) \cong \mathsf{Set}(-,U-). \tag{1.4.5}$$

  Taking a vector space, V, and set, X, this becomes a bijection

  $$\mathsf{Vect}_{\Bbbk}(FX,V) \cong \mathsf{Set}(X,UV). \tag{1.4.6}$$

  Given a linear map $FX \to V$ we can restrict to the basis X and this gives us a function $X \to UV$. Conversely, given a function $X \to UV$ we can extend this by linearity to a linear map $FX \to V$. This is the bijection in question.

- Consider the functor $\Pi\colon \mathsf{Set} \times \mathsf{Set} \to \mathsf{Set}$ which sends a pair of sets, (X,Y), to its Cartesian product, $X \times Y$. This is right adjoint to the diagonal functor $\Delta\colon \mathsf{Set} \to \mathsf{Set} \times \mathsf{Set}$ which sends X to (X,X). The adjunction here corresponds to an isomorphism

  $$(\mathsf{Set} \times \mathsf{Set})(\Delta-,-) \cong \mathsf{Set}(-,\Pi-) \tag{1.4.7}$$

  which, given a set, X, and a pair of sets, (Y,Z), gives rise to an isomorphism

  $$(\mathsf{Set} \times \mathsf{Set})((X,X),(Y,Z)) \cong \mathsf{Set}(X, Y \times Z). \tag{1.4.8}$$

  Morphisms in $\mathsf{Set} \times \mathsf{Set}$ are just pairs of morphisms in Set, so this is expressing the fact that a pair of function $(f\colon X \to Y, g\colon X \to Z)$ is really the same as a single function $(f \times g)\colon X \to Y \times Z$. Given (f,g) we can construct $f \times g$ by defining $(f \times g)(x) = (f(x), g(x))$, and given $f \times g$ we can define f and g by defining $f = \pi_1 \circ (f \times g)$ and $g = \pi_2 \circ (f \times g)$, where π_i is projection onto the ith factor. This adjunction is then related to the fact that $\times$ is the product in Set.

- Consider a pair of rings, R and S, and some fixed ring homomorphism $f : R \to S$. We can view S as a right R-module by $s \,.\, r = sf(s)$. We may consider the following three functors between categories of right modules:
  - $f_! : \mathsf{Mod}\text{-}R \to \mathsf{Mod}\text{-}S$ given by $f_! M = M \otimes_R S$ (extension of scalars);
  - $f_* : \mathsf{Mod}\text{-}R \to \mathsf{Mod}\text{-}S$ given by $f_* M = \mathrm{Hom}_R(S, M)$;
  - $f^* : \mathsf{Mod}\text{-}S \to \mathsf{Mod}\text{-}R$ given by $f^* N = N_R$ where N_R in which $n \,.\, r = n \,.\, f(r)$ with the action on the left being that of N_R and on the right the action of N.

  These three constructions are actually related by a pair of adjunctions, namely, $f_!$ is left adjoint to restriction of scalars, f^*, and f^* is left adjoint to f_*.
- The inclusion $\mathsf{Ab} \hookrightarrow \mathsf{Grp}$ is left adjoint to the abelianisation functor, $\mathsf{Grp} \to \mathsf{Ab}$, defined by $G \mapsto G/[G, G]$.
-

We will now give the two most common alternative definitions of adjunctions, which we derive here from our definition. Given an adjunction, $L \dashv R$, we have an isomorhpism

$$\mathsf{C}(L-, -) \cong \mathsf{D}(-, R-). \tag{1.4.9}$$

To make this a bijection of sets we need an object of $\mathsf{D} \times \mathsf{C}$ to apply these functors to. Suppose that we have some object $x \in \mathsf{D}$, then we can produce an object of C using L, namely $Lx \in \mathsf{C}$. We can then apply these functors to $(x, Lx) \in \mathsf{D} \times \mathsf{C}$, and we get

$$\mathsf{C}(Lx, Lx) \cong \mathsf{D}(x, RLx). \tag{1.4.10}$$

The nice thing about this is that we know a special element of $\mathsf{C}(Lx, Lx)$, namely id_{Lx}. In fact, looking at where identities map is a good way to study hom-functors in general, and is what underpins the Yoneda lemma. Let $\eta_x : x \to RLx$ be the image of id_{Lx} under this isomorphism. We call η_x a **universal morphism**.

Similarly, given $y \in \mathsf{C}$ we may produce $Ry \in \mathsf{D}$ and then we can apply these functors to $(Ry, y) \in \mathsf{D} \times \mathsf{C}$, giving the bijection

$$\mathsf{C}(LRy, y) \cong \mathsf{D}(Ry, Ry). \tag{1.4.11}$$

Let $\varepsilon_y : LRy \to y$ be the image of id_{Ry} under this isomorphism.

It turns out that these universal morphisms satisfy the property that for all morhpisms $f : Ly \to x$ there exists a unique morphism $g : y \to Rx$ such that

$$\begin{tikzcd} Fy \arrow[d, dashed, "Lg"'] \arrow[dr, "f"] & \\ LRx \arrow[r, "\varepsilon_x"'] & x \end{tikzcd} \tag{1.4.12}$$

commutes and for all morphisms $g\colon y \to Rx$ there exists a unique morphism $f\colon Ly \to x$ such that

$$\begin{tikzcd} y \arrow[r, "\eta_y"] \arrow[dr, "g"'] & RLy \arrow[d, dashed, "Rf"] \\ & Rx \end{tikzcd} \tag{1.4.13}$$

commutes.

An assignment of such universal morhpisms for all objects of C and D is actually equivalent to defining the adjunction by taking $\Phi\colon \mathsf{C}(L-,-) \Rightarrow \mathsf{D}(-,R-)$ to be defined by $\Phi_{y,x}(f) = Gf \circ \eta_y$ and $\Phi^{-1}_{y,x}(g) = \varepsilon_x \circ Fg$. Thus, one may define an adjunction via universal morphisms.

We can take the universal morphisms, η_x and ε_y, for all $x \in \mathsf{D}$ and $y \in \mathsf{C}$, and combine them into natural transformations $\eta\colon \mathrm{id}_{\mathsf{D}} \Rightarrow RL$ and $\varepsilon\colon LR \Rightarrow \mathrm{id}_{\mathsf{C}}$. These are called the **unit** and **counit** of the adjunction, and they make the diagrams

$$\begin{tikzcd} L \arrow[r, Rightarrow, "L\eta"] \arrow[dr, Rightarrow, "\mathrm{id}_L"'] & LRL \arrow[d, Rightarrow, "\varepsilon L"] \\ & L \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} R \arrow[r, Rightarrow, "\eta R"] \arrow[dr, Rightarrow, "\mathrm{id}_R"'] & RLR \arrow[d, Rightarrow, "R\varepsilon"] \\ & R \end{tikzcd} \tag{1.4.14}$$

commute. Specifying a unit and counit such that these diagrams commute is the same as specifying an adjunction. Thus, one may define an adjunction via its unit and counit.

In components the diagrams above become

$$\begin{tikzcd} Ly \arrow[r, "L\eta_y"] \arrow[dr, "\mathrm{id}_{Ly}"'] & LRLy \arrow[d, "\varepsilon_{Ly}"] \\ & Ly \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} Rx \arrow[r, "\eta_{Rx}"] \arrow[dr, "\mathrm{id}_{Rx}"'] & RLRx \arrow[d, "R\varepsilon_x"] \\ & Rx. \end{tikzcd} \tag{1.4.15}$$

Applying R to the left diagram and L to the right diagram these become

$$\begin{tikzcd} RLy \arrow[r, "RL\eta_y"] \arrow[dr, "\mathrm{id}_{RLy}"'] & RLRLy \arrow[d, "R\varepsilon_{Ly}"] \\ & RLy \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} LRx \arrow[r, "L\eta_{Rx}"] \arrow[dr, "\mathrm{id}_{LRx}"'] & LRLRx \arrow[d, "LR\varepsilon_x"] \\ & LRx. \end{tikzcd} \tag{1.4.16}$$

Going back to natural transformations these become

$$\begin{tikzcd} RL \arrow[r, Rightarrow, "RL\eta"] \arrow[dr, Rightarrow, "\mathrm{id}_{RL}"'] & RLRL \arrow[d, Rightarrow, "R\varepsilon L"] \\ & RL \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} LR \arrow[r, Rightarrow, "L\eta R"] \arrow[dr, Rightarrow, "\mathrm{id}_{LR}"'] & LRLR \arrow[d, "LR\varepsilon"] \\ & LR. \end{tikzcd} \tag{1.4.17}$$

These are exactly the diagrams that must commute for $(RL, R\varepsilon L, \eta)$ and $(LR, L\eta R, \varepsilon^{-1})$ into monads! So, every adjunction gives rise to a monad, and conversely every monad gives rise to (typically) multiple adjunctions, but we don't have time for that. We should at least define a monad though.

Definition 1.4.18 — Monad A monad is a monoid in a category of endofunctors.

Ok, not the most useful definition immediately, here's what it means if we spell it out. We have the monoidal category $[\mathsf{C}, \mathsf{C}]$, whose objects are endofunctors, $\mathsf{C} \to \mathsf{C}$, and whose morphisms are natural transformations. The monoidal product is composition of endofunctors, and the monoidal unit is the identity functor, id_{C}. A monad is a triple, (T, μ, η) consisting of an endofunctor, $T\colon \mathsf{C} \to \mathsf{C}$, and natural transformations $\mu\colon T^2 \Rightarrow T$ and $\eta\colon \mathrm{id}_{\mathsf{C}} \Rightarrow T$. These are, of course, subject to the fact that the relevant diagrams must commute, but we won't go into those details.

1.5 Details of $\mathsf{Vect}_{\Bbbk}$

Much of our work will take place in the context of $\mathsf{Vect}_{\Bbbk}$, which is a symmetric monoidal category under the tensor product. In this section we'll look at some results specific to $\mathsf{Vect}_{\Bbbk}$.

First, $\mathsf{Vect}_{\Bbbk}$ is an abelian category, which for our purposes simply means that it has coincident initial and terminal objects, specifically the zero vector space, 0. It also has all finite products and coproducts, which again coincide, and are given by the direct sum. Another part of the definition of an abelian category is that it is Ab-enriched, meaning that the hom sets are actually abelian groups in a way that is compatible with composition. For $\mathsf{Vect}_{\Bbbk}$ this abelian group structure is simply pointwise addition of linear maps. In fact, $\mathsf{Vect}_{\Bbbk}$ is actually a **closed**[4] symmetric monoidal category, meaning that it's actually enriched over itself, so the functor

[4] The functor $- \otimes a$ has a right adjoint for all objects a.

$$\hom(-,-) = \mathrm{Hom}_{\Bbbk}(-,-) = \mathsf{Vect}_{\Bbbk}(-,-) \tag{1.5.1}$$

takes values in $\mathsf{Vect}_{\Bbbk}$, with scalar multiplication of linear maps given pointwise.

The additive structure of $\mathsf{Vect}_{\Bbbk}$ is actually relatively simple. One way we can see this is that every short exact sequence

$$0 \to U \to V \to W \to 0 \tag{1.5.2}$$

splits. That is, we always have[5] $V \cong U \oplus W$. This is not the case in the more general setting of R-Mod where short exact sequences do not always split.

[5] More rigorously, we should say that the short exact sequence above is isomorphic to a short exact sequence $0 \to U \to U \oplus W \to W \to 0$, where $U \to U \oplus W$ is inclusion and $U \oplus W \to W$ is projection. This means that not only is $V \cong U \oplus W$ but that composing this isomorphism with $U \to V$ gives inclusion and this isomorphism with projection gives $V \to W$.

More concretely, given the above short exact sequence of vector spaces we can always take $W = U^{\perp}$ with respect to some inner product which we can induce from a basis for U carried through the image and then extended to a basis for V. So the fact that not all R-modules are free comes into play here.

There are some common adjunctions that one may meet when working with vector spaces, we've already mentioned these, but we state them here for completeness.

Lemma 1.5.3 There is an adjunction $- \otimes V \dashv \hom(V, -)$.

Lemma 1.5.4 There is an adjunction $F \dashv U$ where $F\colon \mathsf{Set} \to \mathsf{Vect}_{\Bbbk}$ is the free vector space functor and $U\colon \mathsf{Vect}_{\Bbbk} \to \mathsf{Set}$ is the forgetful functor.

The free vector space may be defined by

$$FX = \{\alpha : X \to \Bbbk \mid \alpha \text{ is finitely supported}\}. \tag{1.5.5}$$

That is, α is a function $X \to \Bbbk$ such that $\alpha(x) = 0$ for all but finitely many $x \in X$. The idea is that $\alpha(x)$ is the coefficient of the basis vector x in the expansion of α in the basis X. Note that if V is a vector space with basis B then we have that $V \cong FB$, which says that every vector space is free. The same is not true in the more general setting of R-Mod, in which finding a basis is not always possible.

The dual space is defined by

$$V^* = \hom(V, \Bbbk). \tag{1.5.6}$$

For finite dimensional vector spaces we have a (non-canonical) isomorphism $V \cong V^*$, and a canonical isomorphism $V \cong V^{**}$. For infinite dimensional vector spaces we have an inclusion $V \hookrightarrow V^*$, but this is not an isomorphism.

Note that for a fixed vector space, W, we have the functors

$$W \otimes -, \quad - \otimes W, \quad \hom(W, -), \tag{1.5.7}$$

which are all functors $\mathsf{Vect}_{\Bbbk} \to \mathsf{Vect}_{\Bbbk}$, and the functor $\hom(-, W) : \mathsf{Vect}_{\Bbbk}^{\mathrm{op}} \to \mathsf{Vect}_{\Bbbk}$. These are all **exact** meaning that they preserve short exact sequences, that is, if

$$0 \to V' \to V \to V'' \to 0 \tag{1.5.8}$$

is a short exact sequence then $F : \mathsf{Vect}_{\Bbbk} \to \mathsf{Vect}_{\Bbbk}$ is an exact functor if

$$0 \to FV' \to FV \to FV'' \to 0 \tag{1.5.9}$$

is a short exact sequence, and $G : \mathsf{Vect}_{\Bbbk}^{\mathrm{op}} \to \mathsf{Vect}_{\Bbbk}$ is an exact functor if

$$0 \to GV'' \to GV \to GV' \to 0 \tag{1.5.10}$$

is a short exact sequence.