Extensions of Modules

MMath Project Dissertation

Student:
David Robertson

Supervisor:
Prof. Peter Jørgensen

School of Mathematics
and Statistics

Newcastle University
1st May 2014

Abstract

A *module* is an algebraic structure which simultaneously generalises the idea of a vector space and an Abelian group. Two or more modules may be combined in a specific way to form larger modules called *extensions*. There are often many ways to do this, resulting in different extensions—how can we classify them?

Given two modules to extend, it is difficult to produce a list of all possible extensions. However, an object called 'Ext' can provide such a list! We introduce the appropriate language from category theory to describe Ext. The project concludes by demonstrating the equivalence between extension lists and Ext.

Contents

Chapter 1

Modules

We introduce modules, giving their definition and some examples. Next we investigate how to form sub-, quotient and product modules. This will give us a method to combine two small modules into one larger structure. The rest of the chapter is concerned with distinguishing between such combinations.

We follow the path laid out by Hilton and Stammbach [3], using much of their notation and terminology.

1.1 Introduction

We're familiar with the idea of rings, vector spaces, and groups. Each consists of a set of elements and a way to combine two elements. Ring elements may be added and multiplied; vectors may be added; and group elements may be either added or multiplied, depending on how we think of the group in question.

Our rings will always contain a multiplicative identity 1. Their multiplication need not be commutative.

Each structure is inherently different. For example, groups consist of a single operation whereas rings and vector spaces have two operations. Even then, in a ring we multiply pairs of ring elements; in a vector space we multiply a single vector by a scalar.

Modules are a generalisation encompassing all three of these structures. At first sight, their definition resembles that of a vector space.

In truth, modules only generalise *Abelian* groups.

Definition 1.1. Let Λ be a ring. A (left) Λ-*module* is an Abelian group $(M, +)$ equipped with a binary operation

$$\cdot : \Lambda \times M \to M$$

satisfying the following axioms for all elements $a, b \in M$ and $\lambda, \lambda' \in \Lambda$.

These axioms ensure that the scaling operation behaves as a ring action.

M1. $(\lambda + \lambda')a = \lambda a + \lambda' a$

M2. $(\lambda\lambda')a = \lambda(\lambda' a)$

M3. $1_\Lambda a = a$

M4. $\lambda(a + b) = \lambda a + \lambda b$

Using the word 'resembles' is putting it politely! This is a carbon copy of the definition of a vector space except for one word: *ring*. Accordingly, we may think of modules as being vector spaces, but with scalars belonging to a ring instead of a field. This radically changes the properties of our 'scalars'—they are no longer guaranteed to be invertible, and need not commute with one another. Let us see some examples.

1.1.1 Examples

Example 1.2 (Choosing the ring)**.** Let M be a Λ-module.

1. Choose Λ to be a field, K say. Then (as noted above) the definition becomes that of a vector space over K.

2. Choose M to be the ring itself, Λ. We may do so because rings form Abelian groups under their addition. If we define scaling by

$$\underbrace{\lambda \cdot a}_{\text{module scaling}} \quad = \quad \underbrace{\lambda * a}_{\text{ring multiplication}}$$

   we obtain the so-called left regular module of Λ. (Each of the module axioms follows from the ring axioms.) Note that there's no distinction between 'λ' and 'a' any more, as both elements lie in the same ring.

3. If we choose $\Lambda = \mathbb{Z}$, then we have no choice in how to define our scaling. The module axioms ensure that

$$\begin{aligned} 2a &= (1+1)a \\ &= 1a + 1a && \text{(by M1)} \\ &= a + a. && \text{(by M3)} \end{aligned}$$

   More generally, na must be the sum of n copies of a (i.e. the nth multiple of a) for any positive integer n. We are also forced to take $0a = 0$ and $(-n)a = -(na)$ due to proposition 1.3 (to follow shortly). The scaling operation is determined the moment we declare M to be a $\mathbb{Z}$-module! It's straightforward to check that axioms M1 to M4 are satisfied by this scaling.

   In a non-Abelian group, we would say *power* rather than multiple.

   There are two different structures in play: a group $(M, +)$ and the associated $\mathbb{Z}$-module $(M, +, \cdot)$. Each structure entails the other: we can form the $\mathbb{Z}$-module from the group as above; for the other way round, simply forget about the scaling operation.

   The scaling merely produces multiples of group elements without adding any additional structure to the group. Hence we may think of Abelian groups and $\mathbb{Z}$-modules as being the same.

4. For any ring Λ we can equip the trivial group 0 with the trivial scaling $\lambda 0 = 0$. This is the *trivial* or *zero* Λ-module.

As promised, we now show that zero and negative scalars act as we might expect them to.

Proposition 1.3 (Scaling by zero and negatives)**.** *In any Λ-module M, $0a = 0$ and $(-\lambda)a = -(\lambda a)$.*

Proof. Since 0_Λ is the additive identity of Λ, we have $0_\Lambda + 0_\Lambda = 0_\Lambda$. Then $(0_\Lambda + 0_\Lambda)a = 0_\Lambda a$, meaning $0_\Lambda a + 0_\Lambda a = 0_\Lambda a$ (by M1.) We don't know which group element $0_\Lambda a$ is, but we do know that it has an inverse. Adding this inverse to both sides yields $0_\Lambda a = 0_M$.

We also have that $\lambda + (-\lambda) = 0_\Lambda$. Invoking M1 again, we see that $\lambda a + (-\lambda)a = 0_\Lambda a = 0_M$. Adding the inverse element $-(\lambda a)$ to both sides allows us to conclude that $(-\lambda)a = -(\lambda a)$. □

This result means that the integers $\mathbb{Z}$ don't 'have enough room' to give us anything other than Abelian groups when we form $\mathbb{Z}$-modules.

* * *

The importance of Λ is huge: different choices of rings yield very different structures. At the same time, it is possible to view the same Abelian group M as a module over *different* rings.

Example 1.4 (Same group, different rings)**.** Let M be a vector space over a field K. We can already view M as a K-module.

1. Let T be a linear transformation of M. We may view M as a module over the polynomial ring $\Lambda = K[T]$ if we equip it with the scaling

$$(c_n T^n + \cdots + c_1 T + c_0)a = c_n T^n(a) + \cdots + c_1 T(a) + c_0 a.$$

On the left, T^n is a placeholder to attach coefficients to; on the right, T^n stands for repeated application of T.

   For a concrete example, choose the vector space $M = \mathbb{R}^2$ over $K = \mathbb{R}$ and let T be (given by) the matrix $T = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Choose the point $a = \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ and polynomials $p_1(T) = T^2 + 1$, $p_2(T) = T - 2$. Then we may compute the scaling $p_2(p_1 a)$.

$$p_1 a = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

$$p_2(p_1 a) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \end{pmatrix} - 2 \begin{pmatrix} 5 \\ 2 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \end{pmatrix} - \begin{pmatrix} 10 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

   The polynomial product is $p_2 p_1(T) = (T - 2)(T^2 + 1) = T^3 - 2T^2 + T - 2$. The resulting polynomial may also be used to scale a, giving

$$\begin{aligned}(p_1 p_2)(a) &= \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^3 \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 2 \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 8 \\ 1 \end{pmatrix} - 2 \begin{pmatrix} 4 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad = \begin{pmatrix} 0 \\ -2 \end{pmatrix}.\end{aligned}$$

   We see that $(p_1 p_2)(a) = p_1(p_2 a) = \left(\begin{smallmatrix} 0 \\ -2 \end{smallmatrix}\right)$. This is no coincidence: the result is guaranteed because of axiom M2.

2. Let $\Lambda = \mathcal{T}$ be the set of all linear transformations of M. Let us add transformations $T_1, T_2 \in \mathcal{T}$ pointwise and 'multiply' transformations by composing them.

$$\begin{aligned}(T_1 + T_2)(a) &= T_1(a) + T_2(a) \\ (T_1 \circ T_2)(a) &= T_1(T_2(a))\end{aligned}$$

   Together, $\mathcal{T}$ and these operations form a ring. We can view our original vector space M as a $\mathcal{T}$-module by defining the scaling Ta to be the evaluation $T(a)$.

3. This construction also works if we choose our set of transformations Λ to be a subring of $\mathcal{T}$.

1.1.2 Motivation

In this text, we investigate modules to develop their properties and structure, rather than using them to make deductions about other objects. With that said, let us mention a few reasons why modules are worth studying.

Multitasking Because modules describe pre-existing structures (rings, Abelian groups, vector spaces), we may investigate all of these objects at once with one theory. Further, the higher-level viewpoint of module theory lets us determine how these objects are different. For example, every vector space has a basis, but only certain Abelian groups—those which are called 'free Abelian'—have bases. Module theory can provide an explanation for why differences such as these occur.

Representation Theory Groups (and modules, as we shall see later) can be described by means of a presentation $G = \langle X \mid R \rangle$. From practical experience, we know that presentations can be awkward to work with. A formal example of this is the *word problem*, one of Max Dehn's three group decision problems. [1]

Each $x_i \in X$ and $\epsilon_i \in \{\pm 1\}$

> Given a finite presentation $G = \langle X \mid R \rangle$ of a group G and a word $w = x_{i_1}^{\epsilon_1} \dots x_{i_n}^{\epsilon_n}$, decide whether or not $w = 1$ in G.

This problem was shown to be undecidable by Novikov in [5]. 'Undecidable' means that no algorithm exists which is capable of correctly answering 'Yes' or 'No' for all possible input presentations.

It can be useful to have alternative ways to represent group elements. In representation theory, group elements are represented by linear transformations of a given vector space. We have seen how sets of linear transformations can form rings. By studying modules over this ring, we can also study the represented group, and vice versa.

Homological Algebra Modules are the primary type of structure in homological algebra. Weibel introduces the subject in [7] as "a tool used to prove non-constructive existence theorems in algebra (and in algebraic topology)" which allows us to determine when "various kinds of constructions [...] are possible." After emerging from parts of topology in the 1940s, it became a valuable field in its own right, and today is used in a variety of different parts of mathematics.

1.2 Working with modules

Most mathematical structures we're familiar with have some notion of being broken down into smaller components. Often we can perform some kind of identification of elements to yield a quotient structure. Finally, we can usually combine two or more structures as building blocks, to form some kind of product. Modules are no exception!

1.2.1 Sub- and quotient modules

Submodules are defined as we might expect them to be: a subset that is itself a module when it inherits the operations of its parent.

Definition 1.5. Let M be a Λ-module. A subset $M' \subseteq M$ is called a *submodule* (also over Λ) if it is closed under addition and scaling. That is, $a + b \in M'$ and $\lambda a \in M'$ for every $\lambda \in \Lambda$, $a, b \in M'$.

Note that submodules are necessarily subgroups, since they must contain $0a = 0$ and $(-1)a = -a$.

Just like groups, rings and vector spaces, there are always at least two submodules of a given module M. Over the appropriate ring, the zero module is the trivial submodule of M. Additionally, we can always view M as an 'improper' submodule of itself.

Quotient modules are also defined as we might expect: by imprinting the parent module's structure onto a collection of cosets.

Definition 1.6 (Quotient module)**.** Let M' be a submodule of a Λ-module M. The *quotient module* M/M' (also over Λ) has as its underlying group the quotient *group* M/M'. Scaling is defined by is defined by $\lambda(m + M') = (\lambda m) + M'$.

Details. We can always form the group quotient M/M', because any subgroup M' is normal in M. This is because conjugation does nothing in an Abelian group:

$$g + h - g = g - g + h = h$$

Thus M/M' is closed under addition.

Before proceeding, we should check that scaling is well-defined. Let m_1 and m_2 represent the same coset. Then $m_1 = m_2 + m'$ for some $m' \in M'$. Hence $\lambda(m_1 + M') = (\lambda m_2 + \lambda m') + M' = \lambda(m_2 + M') + (\lambda m') + M' = \lambda(m_2 + M')$. The last equality follows from the fact that $\lambda m' \in M'$ since M' is closed under scaling.

We should also confirm that the scaling satisfies the module axioms. In short, each axiom holds in the quotient because it holds in the original group M.

M1.
$$\begin{aligned}(\lambda_1 + \lambda_2)(m' + M') &= (\lambda_1 + \lambda_2)(m') + M' \\ &= \lambda_1 m' + \lambda_2 m' + M' && \text{(by M1 in } M\text{)} \\ &= (\lambda_1 m' + M') + (\lambda_2 m' + M')\end{aligned}$$

M2.
$$\begin{aligned}(\lambda_1 \lambda_2)(m' + M') &= (\lambda_1 \lambda_2)m' + M' \\ &= \lambda_1(\lambda_2 m') + M' && \text{(by M2 in } M\text{)} \\ &= \lambda_1(\lambda_2 m' + M')\end{aligned}$$

M3.
$$\begin{aligned}1(m' + M') &= 1m' + M' \\ &= m' + M' && \text{(by M3 in } M\text{)}\end{aligned}$$

M4.
$$\begin{aligned}\lambda(m_1' + M' + m_2' + M') &= \lambda(m_1' + m_2' + M') \\ &= \lambda(m_1' + m_2') + M' \\ &= (\lambda m_1' + \lambda m_2') + M' && \text{(by M4 in } M\text{)} \\ &= \lambda(m_1' + M') + \lambda(m_2' + M')\end{aligned}$$

Example 1.7. Let us illustrate both sub- and quotient modules at once with some concrete examples.

1. The group of integers $\mathbb{Z}$ has $10\mathbb{Z} = \{10z : z \in \mathbb{Z}\}$ as a subgroup. The quotient $\mathbb{Z}/10\mathbb{Z}$ gives us $\mathbb{Z}_{10}$, the integers modulo ten.

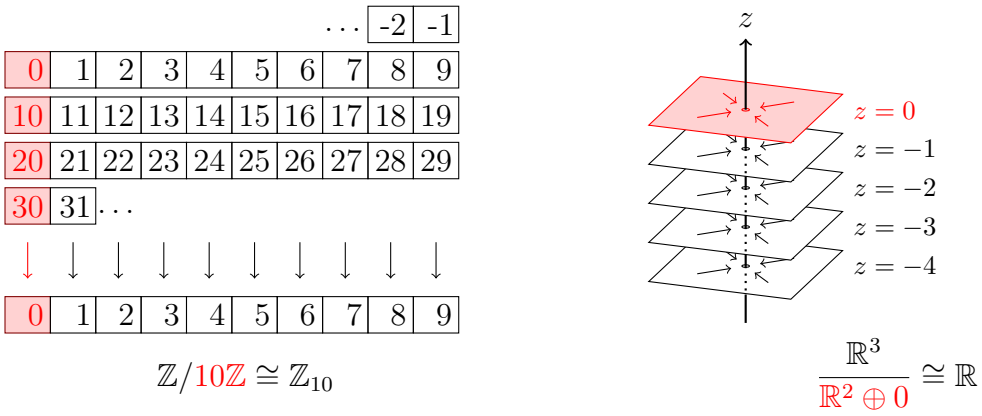


Figure 1.1: Illustrations of two quotient modules in example 1.7. On the left, we factor out $10\mathbb{Z}$ by folding all the rows on top of each other. On the right, we factor out $\mathbb{R}^2 \oplus 0$ by compressing each of the planes down to a point.

2. The vector space $\mathbb{R}^3$ has $\mathbb{R} \times \mathbb{R} \times \{0\} = \{(x, y, 0) : x, y \in \mathbb{R}\}$ as a subspace. The quotient space we obtain is isomorphic to $\mathbb{R}$.

3. Let $\Lambda = \mathcal{M}_2(\mathbb{R})$ be the ring of real 2×2 matrices, and view Λ as a module over itself. Consider the subset $S = \{\left(\begin{smallmatrix} a & 0 \\ b & 0 \end{smallmatrix}\right) : a, b \in \mathbb{R}\}$ of matrices with zeroes in the right column. This is a submodule of Λ since it is closed under addition and scaling (multiplication on the left).

Note that we choose scalars—that is, matrices—from the entirety of Λ, not just the subset S.

$$\begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix} + \begin{pmatrix} a' & 0 \\ b' & 0 \end{pmatrix} = \begin{pmatrix} a + a' & 0 \\ b + b' & 0 \end{pmatrix} \in S \tag{1.1a}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a' & 0 \\ c' & 0 \end{pmatrix} = \begin{pmatrix} aa' + bc' & 0 \\ ca' + dc' & 0 \end{pmatrix} \in S \tag{1.1b}$$

Thus we can make Λ/S into a quotient module; we compute this quotient in example 1.14. Note that products of the opposite order to (1.1b) need not lie in S. For instance:

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \notin S$$

This means we cannot make the set of cosets Λ/S into a quotient *ring*.

1.2.2 Homomorphisms

We now provide the appropriate notion of homomorphism—'structure-preserving map'—for modules.

We typically just say 'homomorphism' when it's clear that we're working with modules over a given ring.

Definition 1.8. Let M and N be modules over the same ring Λ. A Λ-*module homomorphism* is a map $f\colon M \to N$ satisfying

1. $\varphi(a + b) = \varphi(a) + \varphi(b)$
2. $\varphi(\lambda a) = \lambda\varphi(a)$

for every $a, b \in M$ and every $\lambda \in \Lambda$.

In other words, a module homomorphism is an ordinary group homomorphism between the underlying groups M and N which additionally respects the scaling provided by Λ.

Example 1.9 (Noteworthy homomorphisms)**.** Let us mention a few special homomorphisms that exist for all choices of ring Λ. Let M and N be Λ-modules and let M' be a submodule of M.

1. The identity homomorphism (in fact isomorphism) $\mathrm{Id}_M \colon M \to M$ maps m to m.
2. The zero homomorphism $0_{MN} \colon M \to N$ maps every element m to 0_N.
3. The inclusion $\iota \colon M' \to M$ of a submodule M' into M is a homomorphism.
4. The canonical projection $\pi \colon M \to M/M'$ onto a quotient maps elements m to their cosets $m + M'$. All such maps are homomorphisms.

We often drop the subscripts on zero maps if there's no need to highlight the domain or codomain.

The idea of an isomorphism carries through to modules easily.

Definition 1.10. Let $\varphi \colon M \to N$ be a Λ-module homomorphism. We call φ an *isomorphism* if it is additionally a bijection (i.e. invertible). We write $M \cong N$ to denote the existence of an isomorphism $M \to N$.

1.2.3 Kernel, image and cokernel

Yet more notions from algebra have their place in module theory.

Definition 1.11 (Kernel and image)**.** To any Λ-module homomorphism $\varphi \colon M \to N$ there are two associated Λ-modules: the *kernel* $\ker \varphi = \{m \in M : \varphi(m) = 0\}$ and the *image* $\operatorname{Im} \varphi = \{\varphi(m) : m \in M\}$.

Details. Group theory tells us that these two sets are subgroups of M and N respectively. To see that the kernel is closed under scaling, let $m \in \ker \varphi$. Then $\varphi(\lambda m) = \lambda \varphi(m) = \lambda 0 = 0$, so $\lambda m \in \ker \varphi$. For the image, a scaling of an image element $\varphi(m)$ looks like $\lambda \varphi(m) = \varphi(\lambda m) \in \operatorname{Im} \varphi$.

The kernel and the image are related as modules in the same way they are for groups, vector spaces and rings. This is summarised in the following theorem.

Theorem 1.12 (First isomorphism theorem: modules)**.** *Let $\varphi \colon M \to N$ be a Λ-module homomorphism. The quotient $M/\ker \varphi$ is isomorphic to $\operatorname{Im} \varphi$.*

Proof. Let $\theta \colon M/\ker \varphi \to \operatorname{Im} \varphi$ be given by

$$\theta(m + \ker \varphi) = \varphi(m).$$

The proof of this theorem for groups establishes that θ is both well-defined and a group isomorphism; we just need to determine how θ interacts with scaling. We compute

$$\theta(\lambda[m + \ker \varphi]) = \theta(\lambda m + \ker \varphi) = \varphi(\lambda m) = \lambda \varphi(m) = \lambda \theta(m + \ker \varphi).$$

Hence θ is an isomorphism of modules. □

Corollary 1.13. *For any Λ-module M, $M/0 \cong M$ and $M/M \cong 0$.*

Proof. Apply the theorem to the identity and zero homomorphisms Id_M and 0_{MN}. □

We now use this result to identify a quotient module from a previous example, by identifying a suitable homomorphism that does the hard work for us.

Example 1.14. In example 1.7.3 we viewed the ring $M_2(\mathbb{R})$ as a module over itself and considered the quotient module $M_2(\mathbb{R})/S$, where S was the submodule $S = \{\left(\begin{smallmatrix} a & 0 \\ b & 0 \end{smallmatrix}\right) : a, b \in \mathbb{R}\}$. To see what this quotient looks like, consider the homomorphism $\varphi\colon M_2(\mathbb{R}) \to M_2(\mathbb{R})$ which puts zeros in the *left* column:

$$\varphi \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & b \\ 0 & d \end{pmatrix}$$

This really is a homomorphism, since addition is preserved:

$$\varphi\left[\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}\right] = \varphi \begin{pmatrix} a+a' & b+b' \\ c+c' & d+d' \end{pmatrix} = \begin{pmatrix} 0 & b+b' \\ 0 & d+d' \end{pmatrix}$$
$$\varphi \begin{pmatrix} a & b \\ c & d \end{pmatrix} + \varphi \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} = \begin{pmatrix} 0 & b \\ 0 & d \end{pmatrix} + \begin{pmatrix} 0 & b' \\ 0 & d' \end{pmatrix} = \begin{pmatrix} 0 & b+b' \\ 0 & d+d' \end{pmatrix}$$

and scaling is preserved:

$$\varphi\left[\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}\right] = \varphi \begin{pmatrix} aa'+bc' & ab'+bd' \\ ca'+dc' & cb'+dd' \end{pmatrix} = \begin{pmatrix} 0 & ab'+bd' \\ 0 & cb'+dd' \end{pmatrix}$$
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\varphi \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 0 & b' \\ 0 & d' \end{pmatrix} = \begin{pmatrix} 0 & ab'+bd' \\ 0 & cb'+dd' \end{pmatrix}.$$

The map switching left and right columns provides an isomorphism from $\operatorname{Im}\varphi$ to S.

The kernel and image are $\ker\varphi = \{\left(\begin{smallmatrix} a & 0 \\ c & 0 \end{smallmatrix}\right) : a, c \in \mathbb{R}\} = S$ and $\operatorname{Im}\varphi = \{\left(\begin{smallmatrix} 0 & b \\ 0 & d \end{smallmatrix}\right) : b, d \in \mathbb{R}\}$. Thus we can immediately conclude that $M_2(\mathbb{R})/S \cong \operatorname{Im}\varphi$. In fact, $\operatorname{Im}\varphi$ is isomorphic to S, so $M_2(\mathbb{R})/S \cong S$.

* * *

Now we introduce the *cokernel*, which is an example of a 'dual' or 'opposite' construction.

Definition 1.15. Let $\varphi : M \to N$ be a Λ-module homomorphism. The *cokernel* of φ is defined to be the quotient $\operatorname{coker}\varphi = N/\operatorname{Im}\varphi$.

Example 1.16. Let us return to example 1.9, illustrating the kernels, images and cokernels of each homomorphism.

Homomorphism	Kernel	Image	Cokernel
$\mathrm{Id}_M\colon M \to M$	0	M	$M/M \cong 0$
$0_{MN}\colon M \to N$	M	0	$N/0 \cong N$
$\iota\colon M' \to M$	0	M'	M/M'
$\pi\colon M \to M/M'$	M'	M/M'	$\dfrac{M/M'}{M/M'} \cong 0$

Mac Lane gives a formal treatment of duality in [4, sec. II.1]

The cokernel is said to be 'dual' to the kernel. This means that the cokernel is (in some sense) a mirror image of the kernel, and has properties which mirror those of the kernel. Here is an example of a mirrored property: compare the following proposition to the result that $\ker\varphi = 0$ if and only if φ is injective.

Proposition 1.17 (Cokernel measures surjectivity)**.** *Let $\varphi : M \to N$ be a homomorphism of Λ-modules. Then φ is surjective if and only if* $\operatorname{coker}\varphi = 0$ *is the trivial Λ-module.*

Proof. The cokernel is $N/\operatorname{Im}\varphi$. Supposing that φ is surjective means that $\operatorname{Im}\varphi = N$. Then $\operatorname{coker}\varphi = N/N \cong 0$. In the other direction, suppose that $N/\operatorname{Im}\varphi$ is trivial. Then the quotient's underlying set has size one, so it must consist of a single coset $N/\operatorname{Im}\varphi = \{\operatorname{Im}\varphi\}$. But N is equal to the union of all of these cosets, so $N = \operatorname{Im}\varphi$; that is, φ is surjective. □

In short, the cokernel 'detects' whether or not a map is surjective. But 'measures' is a better choice of word: if we use the cokernel to see if a map is surjective, we get more than a binary 'Yes' or 'No' response. To give a somewhat contrived example, let K be a finite field and define maps

e.g. $K = \mathbb{Z}_p$ for p prime.

$$f\colon K \to K^2 : k \mapsto (k, 0),$$
$$g\colon K \to K^3 : k \mapsto (k, 0, 0).$$

Each of these maps is a linear transformation between K-vector spaces. These maps are similar in that they both embed K into the first coordinate of their codomain. Their cokernels are different, however: we have $\operatorname{coker} f \cong K$ (the second coordinate) and $\operatorname{coker} g \cong K^2$ (the last two coordinates). The fact that K^2 is a larger set than K is reflects the observation that g is 'further from being surjective' than f is.

1.2.4 Direct products and sums

We describe two methods of constructing larger modules from smaller building blocks. The two structures are similar, but differ in important respects. We begin with the direct product, which is slightly easier to define.

Definition 1.18. Let A and B be Λ-modules. The *direct product* $A \times B$ is the Cartesian product of sets $A \times B$ equipped with entrywise operations

$$(a, b) + (a', b') = (a + a', b + b'),$$
$$\lambda(a, b) = (\lambda a, \lambda b).$$

Details. Each axiom (the Abelian group axioms and M1–M4) holds for the product because it holds in both of the factors A and B. The zero element is $0 = (0_A, 0_B)$ and negatives are given by $-(a, b) = (-a, -b)$.

As is often the case, there is nothing special about the number two—the direct product can be formed from any number of modules. If we wish, we can construct uncountably large products.

Definition 1.19 (Arbitrary direct product)**.** Let $\{A_j\}_{j\in J}$ be a family of Λ-modules. The *direct product* $\prod_{j\in J} A_j$ has as its ground set the Cartesian product $\prod_{j\in J} A_j$. This is the set of all J-tuples $(a_j)_{j\in J}$, where each entry a_j belongs to A_j.)

Formally, a J-tuple is defined as a map from J to a certain set. The map takes j to the jth entry a_j.

The product is equipped with entrywise operations:

$$(a_j) + (a'_j) = (a_j + a'_j),$$
$$\lambda(a_j) = (\lambda a_j).$$

It is straightforward to verify that $\prod A_j$ is a module. By making a slight alteration, we obtain the closely related object known as the direct sum.

Definition 1.20 (Arbitrary direct sum)**.** Let $\{A_j\}_{j\in J}$ be a family of Λ-modules. The *direct sum* $\bigoplus_{j\in J} A_j$ is the submodule of the direct product $\prod_{j\in J} A_j$, consisting of tuples $(a_j)_{j\in J}$ whose entries are zero almost everywhere. That is, $(a_j)_{j\in J}$ belongs to the direct sum if all but a finite number of entries a_j are zero.

Details. We verify that $\bigoplus A_j$ is closed under addition and scaling. Let (a_j) and (b_j) both belong to the direct sum, and suppose they have m and n non-zero entries respectively. Then their sum $(a_j + b_j)$ has at most $m + n$ non-zero entries.

As for scaling: when we multiply by λ, we cannot turn a zero entry into a non-zero entry because $\lambda 0 = 0$. Thus (λa_j) has at most the same number of non-zero entries as (a_j).

* * *

There is a very good reason why we distinguish these two modules. While they appear similar—indeed, they are identical when we have a finite index set J—these objects interact differently with homomorphisms. In short, products are nice to map into and sums are nice to map out of.

Where all three families are indexed by J.

Let us be more specific. Suppose we have a family $\{A_j\}$ of Λ-modules, and suppose we have two families of homomorphisms $\{\varphi_j \colon A_j \to B\}$ and $\{\psi_j \colon C \to A_j\}$. We may combine (or 'extend') these families into a single map using the direct product and sum.

The second family is easier to extend. Define $\Psi \colon C \to \prod A_j$ by mapping an element c under all of the homomorphisms ψ_j at once and recording the results in a tuple.

$$\Psi(c) = (\psi_j(c))_{j\in J}$$

For example, if $J = \mathbb{N}$ we would calculate each $\psi_n(c)$ and store the result in the nth coordinate of a sequence, producing $(\psi_1(c), \psi_2(c), \psi_3(c), \dots)$. It is straightforward to check that Ψ is a homomorphism (it follows from the fact that each ψ_j is a homomorphism.)

The direct product is associated with a family of homomorphisms $\{\pi_i \colon \prod A_j \to A_i\}_{i\in J}$ known as projections. These simply pick out the jth coordinate from a given tuple. In symbols,

$$\pi_i((a_j)_{j\in J}) = a_i.$$

We can express the fact that Ψ extends the family $\{\psi_j\}$ by writing

$$\pi_i \Psi = \psi_i \quad \text{for all } i \in J. \tag{1.2}$$

If we try to extend the other family $\{\varphi_j \colon A_j \to B\}$ in a similar manner, we encounter a problem. Let us again use $J = \mathbb{N}$ as an example. We would like an extension $\Phi \colon \prod A_j \to B$ to be a homomorphism which maps an element of the product in the following manner.

$$\Phi(a_1, a_2, a_3, \dots) = \Phi \begin{pmatrix} & (a_1, 0, 0, \dots) \\ + & (0, a_2, 0, \dots) \\ + & (0, 0, a_3, \dots) \\ + & \vdots \end{pmatrix} = \begin{pmatrix} & \Phi(a_1, 0, 0, \dots) \\ + & \Phi(0, a_2, 0, \dots) \\ + & \Phi(0, 0, a_3, \dots) \\ + & \vdots \end{pmatrix}$$

$$= \varphi_1(a_1) + \varphi_2(a_2) + \varphi_3(a_3) + \cdots = \sum_{j=1}^{\infty} \varphi_j(a_j)$$

The second equality should hold because Φ should be a homomorphism. As for the third equality, we would like Φ to act as φ_j if all but the jth coordinate is empty.

Unfortunately this 'definition' of Φ does not work, because the final result is an infinite sum. In general, this has no meaning in a group. For instance, consider the expression

$$\sum_{j=1}^{\infty} 1 = 1 + 1 + 1 + \cdots$$

in $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$, the integers modulo n. Which of the n elements should we declare the sum's value to be? No choice makes sense!

To avoid this problem, we must restrict Φ to be defined on a smaller module for which the infinite sum $\sum \varphi_j(a_j)$ always makes sense. But since we have no prior information on how addition works in B, we must take drastic action. We require that our tuples (a_j) belong to the direct sum $\bigoplus A_j$, so that they are zero almost everywhere. Then all except for a finite number of summands $\varphi_j(a_j)$ are zero, and $\sum \varphi_j(a_j)$ collapses to a finite sum.

In some sense, we are forcing our 'series' to 'converge'.

In summary, we may extend the maps $\{\varphi_j\}$ into $\Phi \bigoplus A_j \to C$, defined by

$$\Phi((a_j)_{j\in J}) = \sum_{j\in J} \varphi_j(a_j).$$

Again, it is straightforward to check that Ψ is a homomorphism.

Like the product, the direct sum is associated with a family of homomorphisms $\{\iota_i \colon A_i \to \bigoplus A_j\}_{i\in J}$. Each map ι_i (known as an injection) maps an element of A_i into the ith coordinate of an otherwise empty tuple. Symbolically, $\iota_i(a_i) = (b_j)_{j\in J}$, where $b_i = a_i$ and all other b_j are zero. This allows us to state the extension property analogous to (1.2) as

$$\Phi\iota_i = \varphi_i \quad \text{for all } i \in J. \tag{1.2'}$$

The similarity of (1.2) and (1.2') is another instance of a dual or mirror property; the direct product and sum are dual to each other.

* * *

The usefulness of these extended homomorphisms Ψ and Φ is twofold. Firstly, they exist, allowing us to extend homomorphisms in the first place. Secondly, they are unique: if the projections π_i and injections ι_i are given to us, there is only one way to construct Ψ and Φ which satisfy (1.2) and (1.2'). Indeed, these two equations *characterise* the direct product and sum, respectively.

For the most part, we will only require finite products and sums. We use a kind of matrix notation in such cases, writing

$$\Phi = (\varphi_1, \ldots, \varphi_n) \quad \text{and} \quad \Psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_n \end{pmatrix}$$

for the homomorphisms out of the sum and into the product, respectively. This is particularly useful since the composition of maps $\Phi \circ \Psi$ is given by the matrix product $\Phi\Psi = \sum_{i=1}^{n} \varphi_i\psi_i$.

E.g. $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \gamma = \begin{pmatrix} \alpha\gamma \\ \beta\gamma \end{pmatrix}$, $(\alpha, \beta) \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \alpha\gamma + \beta\delta$.

1.3 Extensions of Modules

Choose any pair of Λ-modules A and B and combine them to form the direct sum $E = A \oplus B$. This contains B as a submodule, though it is disguised in the form $B \cong 0 \oplus B$. The corresponding quotient module

$$\frac{A \oplus B}{0 \oplus B}$$

is isomorphic to A. We can think of E as consisting of copies of A, one for each element of B. Each copy looks like $A \oplus \{b\}$.

We embedded B into E above by using the map $\iota\colon b \mapsto (0, b)$.

The module E is said to extend A by B. More generally, any module E for which $E/B \cong A$ (after B has been suitably embedded into E) is called an extension of A by B.

The direct sum always provides such an extension; are there other ways to extend one module by another? If so, how should we decide if one extension is the same as another—do we require any information other than A, B, and E?

1.3.1 Sequences and extensions

Before we tackle these questions, let us introduce some notation and terminology to describe and work with extensions. We begin with exact sequences.

Definition 1.21 (Exact sequence)**.** A *sequence* is (formally) a list of Λ-module homomorphisms $\varphi_1, \varphi_2, \ldots, \varphi_n$ where each map φ_i maps A_i to A_{i+1}. We often write this as a chain of arrows

$$A_1 \xrightarrow{\varphi_1} A_2 \xrightarrow{\varphi_2} A_3 - \cdots \to A_n \xrightarrow{\varphi_n} A_{n+1}.$$

The sequence above is said to be *exact at* A_i if $\ker \varphi_{i+1} = \operatorname{Im} \varphi_i$ (which implies $\varphi_{i+1} \circ \varphi_i = 0$). If the sequence is exact at each of $A_2, A_3, \ldots A_n$, we say that the sequence itself is *exact*.

The zero module has a lot of influence over an exact sequence: it forces neighbouring homomorphisms to have certain properties.

Proposition 1.22 (Zeroes in exact sequences)**.** *Let $\varphi\colon A \to B$ be a Λ-module homomorphism and let 0 denote the zero module (over Λ).*

1. *The sequence $0 \to A \xrightarrow{\varphi} B$ is exact if and only if φ is injective.*

2. *The sequence $A \xrightarrow{\varphi} B \to 0$ is exact if and only if φ is surjective.*

Proof. 1. The only homomorphism out of the zero module is the map which takes 0 to 0_A. So the sequence is exact if and only if $\ker \varphi = \{0_A\}$, i.e. if and only if φ is injective.

2. Similarly, the only homomorphism mapping into the zero module has to map all elements to zero, so its kernel is B. The sequence if exact if and only if $\operatorname{Im} \varphi$ is equal to this kernel, i.e. $\operatorname{Im} \varphi = B$. But this is the same as saying that φ is surjective. □

We will be particularly interested in 'short exact sequences'. These are exact sequences which look like

$$0 \longrightarrow B \xrightarrow{\kappa} E \xrightarrow{\nu} A \longrightarrow 0. \tag{1.3}$$

Applying the first isomorphism theorem, we see that $E/\ker\nu = \operatorname{Im}\nu$. But we know that $\operatorname{Im}\nu = A$ (as ν is surjective) and that $\ker\nu = \operatorname{Im}\kappa$. Hence $E/\operatorname{Im}\kappa \cong A$. But since κ is injective, $\operatorname{Im}\kappa \cong B$. So E contains a copy of B as a submodule; factoring this out yields a copy of A. This is just a roundabout way of saying E is an extension of A by B.

Caution! It's tempting to replace $\operatorname{Im}\kappa$ by B in the last equation, which would then read "$E/B \cong A$." However, the quotient E/B doesn't strictly exist unless B is a subset of E: it is the image of B that is factored out, not B itself.

For example, take $\kappa\colon \mathbb{Z} \to \mathbb{Z}$ to be multiplication by 2. This would produce the quotient $A = \mathbb{Z}/2\mathbb{Z} \cong \mathbb{Z}_2$. But if we replaced the symbol $2\mathbb{Z}$ with $\mathbb{Z}$, we would conclude $\mathbb{Z}/\mathbb{Z} \cong \mathbb{Z}_2$, which is misleading at best.

This shows we can produce an extension from a short exact sequence. Is there a correspondence in the other direction? Suppose we have modules for which $E/B \cong A$. Then there are two associated homomorphisms. Firstly, there is the inclusion map $\iota\colon B \hookrightarrow E$ which simply takes b to b; secondly, there is the projection $\pi\colon E \to A$ which maps elements e to the image of the coset $e + B$ in A. Computing the following sets

$$\begin{aligned}\ker\pi &= \{e : e + B \text{ is the zero coset}\} = \{e : e \in B\} = B\\ \operatorname{Im}\iota &= \{\iota(b) = b : b \in B\} = B,\end{aligned}$$

we see that $\ker\pi = \operatorname{Im}\iota$. Since the projection is surjective and the inclusion is injective, we may form the short exact sequence below.

$$0 \longrightarrow B \xhookrightarrow{\iota} E \xrightarrow{\pi} A \longrightarrow 0$$

* * *

Hopefully this discussion illustrates that the idea of an extension and of a short exact sequence are roughly equivalent. However, the sequence contains more information (the homomorphisms κ, ν) than the statement $E/B \cong A$. With this in mind we make the following definition.

Definition 1.23. Let A and B be Λ-modules. An *extension* of A by B is a short exact sequence $0 \to B \xrightarrow{\kappa} E \xrightarrow{\nu} A \to 0$. We may refer to an extension by using one of its homomorphisms (κ or ν in this case), or by its central module (E) if the choice of homomorphisms is understood.

Example 1.24. We turn to Abelian groups to provide some examples.

1. As we mentioned earlier, the direct sum $A \oplus B$ is an extension of A by B since we can form the short exact sequence

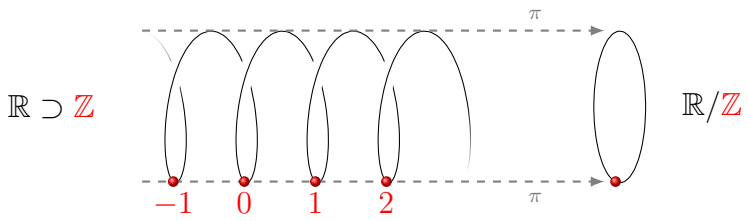
$$0 \longrightarrow B \xrightarrow{\iota_B} A \oplus B \xrightarrow{\pi_A} A \longrightarrow 0.$$

   The injection ι_B maps B into the second coordinate, and the projection π_A extracts A from the first coordinate. Note that this construction works for any ring Λ (not just $\mathbb{Z}$).

2. Consider the map $\kappa\colon \mathbb{Z} \to \mathbb{Z}\oplus\mathbb{Z}$ given by $\kappa(z) = (2z, 0)$. This is an injection, so we have the left part of a short exact sequence. To continue this to the right, define $\nu\colon \mathbb{Z}\oplus\mathbb{Z} \to \mathbb{Z}_2 \oplus \mathbb{Z}$ by $\nu(a, b) = (a \bmod 2, b)$. We can easily check that $\ker\nu = \operatorname{Im}\kappa$, giving us the extension

$$0 \longrightarrow \mathbb{Z} \xrightarrow{\kappa} \mathbb{Z}\oplus\mathbb{Z} \xrightarrow{\nu} \mathbb{Z}_2 \oplus \mathbb{Z} \longrightarrow 0.$$

3. We discussed how given any quotient E/B we can always form an extension using the canonical projection π and inclusion ι. We illustrate the example of $\mathbb{R}/\mathbb{Z}$ by bending the real line $\mathbb{R}$ into a spiral.



The corresponding sequence is

$$0 \longrightarrow \mathbb{Z} \xhookrightarrow{\iota} \mathbb{R} \xrightarrow{\pi} \mathbb{R}/\mathbb{Z} \longrightarrow 0.$$

4. We can construct two extensions of $\mathbb{Z}_2$ by itself, using the two groups of order four. The first uses the direct sum (the Klein *vier* group); the second uses the integers modulo four.

$$0 \longrightarrow \mathbb{Z}_2 \xrightarrow{\iota_1} \mathbb{Z}_2 \oplus \mathbb{Z}_2 \xrightarrow{\pi_2} \mathbb{Z}_2 \longrightarrow 0$$

$$0 \longrightarrow \mathbb{Z}_2 \xrightarrow{\times 2} \mathbb{Z}_4 \xrightarrow{\bmod 2} \mathbb{Z}_2 \longrightarrow 0$$

The two extensions $\mathbb{Z}_2 \oplus \mathbb{Z}_2$ and $\mathbb{Z}_4$ are non-isomorphic groups, since the latter is cyclic but the first is not. This would suggest that these extensions are not the same.

1.3.2 Diagrams and equivalent extensions

To compare extensions we will need to compare their homomorphisms. Things aren't too complicated at the moment, since we only have two maps and one equation ($\ker\nu = \operatorname{Im}\kappa$). But later we'll be working with many maps and many equations relating them.

At the very least, it's useful to have a schematic telling us where maps go to and from!

To make life a little easier, we introduce commutative diagrams. These are a means to visually represent relationships between a collection of maps.

Definition 1.25 (Commutative diagram)**.** A *diagram* is (formally) a directed graph. Vertices represent modules and arrows represent maps between modules—usually homomorphisms. Here are two examples.

$$\begin{array}{ccc} P & & \\ \downarrow{\scriptstyle\pi} & \searrow{\scriptstyle\sigma} & \\ R & \xrightarrow{\rho} & S \end{array} \qquad \begin{array}{ccc} Y & \xrightarrow{\alpha} & A \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\varphi} \\ B & \xrightarrow{\psi} & X \end{array}$$

We say that a diagram *commutes* (or is *commutative*) if all paths between the same start and end point correspond to the same composition of maps. For example, the triangle above commutes if $\rho\pi = \sigma$ and the square commutes if $\varphi\alpha = \psi\beta$. In principle, we need to check all possible paths to see if a diagram commutes. However, it is usually sufficient to break a diagram into simpler shapes (e.g. triangles or squares) and check that each shape commutes.

Arrows are often drawn with decorations to signify that its map has certain properties. We will use the following notation.

Inclusion	Injection	Surjection	Identity map	Map of interest
$\hookrightarrow$	$\rightarrowtail$	$\twoheadrightarrow$	$=$	$\dashrightarrow$

Dotted arrows typically denote maps which we construct for ourselves, i.e. maps which are not given to us.

Hence we write $B \overset{\kappa}{\rightarrowtail} E \overset{\nu}{\twoheadrightarrow} A$ for an extension E of A by B.

With diagrams at our disposal, we are now ready to declare when two extensions are equivalent.

Definition 1.26 (Equivalent extensions)**.** Let $B \overset{\mu}{\rightarrowtail} E \overset{\epsilon}{\twoheadrightarrow} A$ and $B \overset{\mu'}{\rightarrowtail} E' \overset{\epsilon'}{\twoheadrightarrow} A$ be two extensions of A by B. The extensions are said to be *equivalent* if there is a homomorphism $\theta\colon E \to E'$ making the following diagram commute.

$$\begin{array}{ccccc} B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \\ \| & & \downarrow{\scriptstyle\theta} & & \| \\ B & \overset{\kappa'}{\rightarrowtail} & E' & \overset{\nu'}{\twoheadrightarrow} & A \end{array}$$

Here 'the diagram commutes' means $\nu = \nu'\theta$ and $\theta\kappa = \kappa'$.

In short, two extensions are equivalent if there is a nice homomorphism θ between their central modules. By 'nice', we mean that θ makes the two extensions compatible with one another. For example, θ allows us to form a new extension $B \overset{\kappa}{\rightarrowtail} E \overset{\nu'\theta}{\twoheadrightarrow} A$ using parts of both extensions.

If the homomorphism θ exists, it must be an isomorphism. We can prove this fact directly, but we prove it as a corollary of a more general statement.

Lemma 1.27 (Short sequence lemma)**.** *Let $A' \overset{\mu}{\rightarrowtail} A \overset{\epsilon}{\twoheadrightarrow} A''$ and $B' \overset{\mu}{\rightarrowtail} B \overset{\epsilon}{\twoheadrightarrow} B''$ be two short exact sequences. Suppose there exist homomorphisms α, α' and α'' which make the following diagram commute.*

$$\begin{array}{ccccc} A' & \overset{\mu}{\rightarrowtail} & A & \overset{\epsilon}{\twoheadrightarrow} & A'' \\ \downarrow{\scriptstyle\alpha'} & & \downarrow{\scriptstyle\alpha} & & \downarrow{\scriptstyle\alpha''} \\ B' & \overset{\mu'}{\rightarrowtail} & B & \overset{\epsilon'}{\twoheadrightarrow} & B'' \end{array}$$

If any two of the three homomorphisms α', α and α'' are isomorphisms, then the third is an isomorphism too.

There are three distinct cases to be dealt with. We only prove one case directly (the one which shows θ must be an isomorphism). The proofs of the other two are very similar and involve exactly the same type of argument.

Proof. Every map in the diagram is a homomorphism. To prove that α is an isomorphism, we just need to show it is a bijection. We demonstrate injectivity by demonstrating that the map has a trivial kernel.

This type of proof is called a 'diagram chase'—for reasons that should soon become clear!

Suppose that α' and α'' are isomorphisms. We want to show that α is also an isomorphism.

Injectivity Let $a \in \ker \alpha$ be given. We want to show that a must be 0.

An $\Longrightarrow$ arrow means that a substitution is being made on that line.

Since...	we know...		such that/ so then...
	$\alpha a = 0$		$\epsilon' \alpha a = 0$
The right square commutes	$\alpha'' \epsilon = \epsilon' \alpha$	$\Longrightarrow$	$\alpha'' \epsilon a = 0$
α'' is injective			$\epsilon a = 0$
$\ker \epsilon = \operatorname{Im} \mu$	$\exists a' \in A'$		$a = \mu a'$
	$\alpha a = 0$		$\alpha \mu a' = 0$
The left square commutes	$\mu' \alpha' = \alpha \mu$	$\Longrightarrow$	$\mu' \alpha' a' = 0$
μ' is injective			$\alpha' a' = 0$
α' is injective			$a' = 0$
	$a = \mu a'$		$a = 0$

Surjectivity Choose any $b \in B$. We want to find an element a for which $\alpha a = b$. Begin by moving to the right, into B''.

Since...	we know...		such that/ so then...
α'' is surjective	$\exists a'' \in A''$		$\epsilon' b = \alpha'' a''$
ϵ is surjective	$\exists a \in A$		$\epsilon' b = \alpha'' \epsilon a$
The right square commutes	$\alpha'' \epsilon = \epsilon' \alpha$	$\Longrightarrow$	$\epsilon' b = \epsilon' \alpha a$
ϵ' is a homomorphism			$\epsilon'(b - \alpha a) = 0$
$\ker \epsilon' = \operatorname{Im} \mu'$	$\exists b' \in B'$		$b - \alpha a = \mu' b'$
α' is surjective	$\exists a' \in A'$		$b - \alpha a = \mu' \alpha' a'$
The left square commutes	$\mu' \alpha' = \alpha \mu$	$\Longrightarrow$	$b - \alpha a = \alpha \mu a'$
			$b = \alpha(a + \mu a')$

□

Corollary 1.28. *If there is a homomorphism $\theta \colon E \to E'$ which makes two extensions equivalent, then θ must be an isomorphism.*

Proof. The identity maps Id_A and Id_B are isomorphisms. □

Example 1.29 (Equivalent or not?)**.** Let us conclude this section by providing some examples.

1. Consider the following extensions of Abelian groups, where the surjections are given by $\nu(z) = z \bmod 3$ and $\nu'(z) = 2z \bmod 3$.

$$\begin{array}{ccccc} \mathbb{Z} & \xrightarrow{\times 3} & \mathbb{Z} & \xrightarrow{\nu} & \mathbb{Z}_3 \\ \| & & \vdots\ \exists\theta? & & \| \\ \mathbb{Z} & \xrightarrow{\times 3} & \mathbb{Z} & \xrightarrow{\nu'} & \mathbb{Z}_3 \end{array}$$

   Suppose there exists a homomorphism θ making the diagram commute. Then the left square tells us that $\theta(3z) = 3z$ for every integer z. But then $3\theta(z) = 3z$, and hence θ is the identity map $\mathrm{Id}_{\mathbb{Z}}$.

However, this fails to make the right square commute. The top path takes z to $z \bmod 3$; the bottom path takes z to $2\theta(z) \bmod 3 = 2z \bmod 3$. But these results are different when $z \not\equiv 0 \pmod 3$, so the extensions must be inequivalent.

2. In example 1.24.2 we had the following two extensions.

$$\begin{array}{ccccc} \mathbb{Z}_2 & \xrightarrow{\iota_1} & \mathbb{Z}_2 \oplus \mathbb{Z}_2 & \xrightarrow{\pi_2} & \mathbb{Z}_2 \\ \| & & \vdots\ \exists\theta? & & \| \\ \mathbb{Z}_2 & \xrightarrow{\times 2} & \mathbb{Z}_4 & \xrightarrow{\text{mod } 2} & \mathbb{Z}_2 \end{array}$$

The groups $\mathbb{Z}_2 \oplus \mathbb{Z}_2$ and $\mathbb{Z}_4$ are not isomorphic; the latter is cyclic, but the former is not. So the extensions can't possibly be equivalent.

3. An extension E of A and B is said to (be) *split* if it is equivalent to the extension $B \rightarrowtail B \oplus A \twoheadrightarrow A$ (equipped with the canonical injection and projection maps).

   If K is a field, then all extensions of K-modules (K-vector spaces) split. We sketch a proof for finite-dimensional spaces.

$$\begin{array}{ccccc} B & \xrightarrow{\kappa} & E & \xrightarrow{\nu} & A \\ \| & & \downarrow\theta & & \| \\ B & \xrightarrow{\iota_B} & A \oplus B & \xrightarrow{\pi_A} & A \end{array}$$

   Choose bases $\{b_1, \ldots, b_n\}$ and $\{a_1, \ldots, a_m\}$ of A and B respectively. Also choose preimages $e_1, \ldots, e_m$ of the basis elements for A such that $\nu(e_j) = a_j$. Then the set $X = \{\kappa(b_1), \ldots, \kappa(b_n), e_1, \ldots, e_m\}$ can be shown to be a basis for E. Hence we can uniquely represent vectors in E as a list of scalars $\beta_1, \ldots, \beta_n, \alpha_1, \ldots \alpha_n$ which corresponds to the element $e = \sum \beta_i \kappa(b_i) + \sum \alpha_j e_j$.

   We can define the homomorphism $\theta\colon E \to A \oplus B$ by

$$\theta(\beta_1, \ldots, \beta_n, \alpha_1, \ldots \alpha_n) = \Big((\alpha_1, \ldots, \alpha_m), \quad (\beta_1, \ldots, \beta_m)\Big)$$

   (which essentially just reorders the scalars). We can check that θ is a homomorphism and makes the diagram commute. Hence any extension of vector spaces splits to form the direct sum.

1.3.3 The set of equivalence classes

Now we can tell different extensions apart, we would like to produce a list of all extensions for given modules A and B. First, a proposition:

Proposition 1.30. *Let E and E' be two equivalent extensions of A by B. Let $\sim$ denote equivalence of extensions, so that $E \sim E'$. Then $\sim$ is an equivalence relation.*

Here we use E to stand for both the central module and the entire extension.

Proof. There are three properties to demonstrate. Each is illustrated by a diagram in figure 1.2.

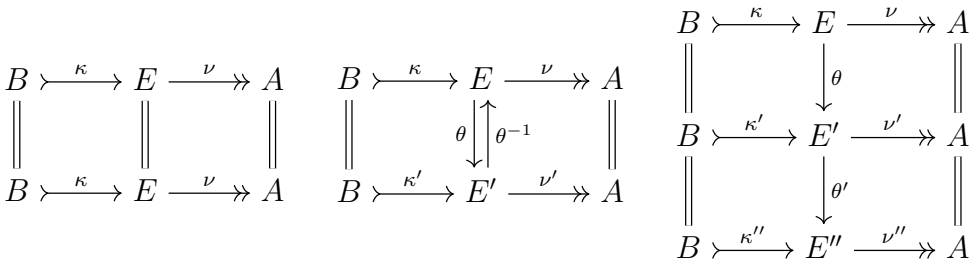


Figure 1.2: Diagrams used in proposition 1.30 to show that $\sim$ is an equivalence relation. Left to right: the diagrams for reflexivity, symmetry, and transitivity.

Reflexivity Any extension is equivalent to itself—choose $\theta = \mathrm{Id}_E$.

Symmetry If $E \sim E'$, then there is a homomorphism $\theta\colon E \to E'$ which makes E equivalent to E'. We showed that θ must be an isomorphism, so its inverse $\theta^{-1}\colon E' \to E$ exists. But then θ^{-1} makes E' equivalent to E.

Transitivity If $E \sim E'$ and $E' \sim E''$, then we have two commutative diagrams. Gluing these together, we obtain one big commutative square (since each of the four smaller squares commute). Thus $E \sim E''$, via the composition $\theta\theta'$. □

Now we really can use the phrase 'equivalent extensions' with a clear conscience. More importantly, we can group together extensions into *equivalence classes*.

Definition 1.31 (Set of equivalence classes)**.** Let A and B be Λ-modules and let E be an extension. The *equivalence class* of E is the set $[E] = \{\text{extensions } E' : E' \sim E\}$. The set of all such classes is denoted by $E(A, B)$.

Ultimately, $E(A, B)$ is the object we're trying to construct. To do this directly, we'd have to consider every possible way to to make an extension of A by B; then decide which equivalence class each extension belongs to—no small feat.

It turns out to be easier to work with a closely related object called $\mathrm{Ext}(A, B)$. In the rest of this project, we construct $\mathrm{Ext}(A, B)$ and prove that it is equivalent to $E(A, B)$.

Chapter 2

Category Theory

Category theory is a framework for working with collections of mathematical objects that share a given type. The field was created in Eilenberg and Mac Lane's 1945 paper [2] as a generalisation of ideas they used in topology. In their own words, category theory "contribute[d] to the current trend towards uniform treatment of different mathematical disciplines."

Our interest in the subject is motivated by Ext. In this chapter, we develop the technical background needed to fully define and understand Ext.

2.1 Introduction

Today, most common mathematical objects are formally defined in terms of sets. For instance, consider a group G. We ask that G be a set equipped with a binary operation. This is a function from $G \times G$ to G. Such a function is defined as a subset of $(G \times G) \times G$. But a product $A \times B$ is defined as a set of ordered pairs, and an ordered pair (a, b) may even be defined as the set $\{\{a\}, \{a, b\}\}$!

Rather than turtles, it's sets all the way down!

By building everything out of sets, we have come to use set theory as the formal foundations of mathematics (at least for the time being). Category theory is a different framework for mathematics. It takes a top-down view, instead of the bottom-up perspective of set theory. Rather than worry about how to construct or describe different objects, we focus on the connections between them.

Usually 'Zermelo-Fraenkel' set theory.

We continue to use Hilton and Stammbach's terminology and notation, though there are others in use. Mac Lane provides a translation dictionary in [4, p. 249].

2.1.1 Categories

Let us begin by defining what a category should be.

Definition 2.1 (Category)**.** A *category* $\mathcal{C}$ consists of three pieces of data:

1. A class of *objects* $A, B, C \ldots$
2. A class of *morphisms* $\alpha, \beta, \gamma, \ldots$
3. A means to *compose* two morphisms to obtain a third morphism.

We say 'class' rather than 'set' to avoid problems like the set of all sets—see Potter's discussion in [6, app. C].

Each morphism α is associated with two objects: a *domain* A and *codomain* B, say. For shorthand, we write $\alpha \colon A \to B$. Given a A and B, there may be only

one morphism $A \to B$, there may be multiple such morphisms, or there may be none whatsoever.

There are some further restrictions (axioms):

1. For each object A there is an *identity morphism* 1_A such that $1_A\varphi = \varphi$ and $\psi 1_A = \psi$ for all $\varphi\colon C \to A$ and $\psi\colon A \to B$.

2. We may compose morphisms only if their (co)domains are compatible. For example, if we have morphisms $\alpha\colon A \to B$ and $\beta\colon B \to C$, then the composition $\beta\alpha\colon A \to C$ exists, but $\alpha\beta$ does not (unless $C = A$).

3. Composition is associative. For any three compatible morphisms α, β, γ, we require that $(\alpha\beta)\gamma = \alpha(\beta\gamma)$. Hence we need not use brackets in compositions.

Compare to how the information in a group $(G, \cdot)$ is given by the operation $\cdot$, not the set of symbols G.

Note how all the information about the category is held in the morphisms—specifically, in the composition rule. Thus we must be careful when comparing morphisms. Two morphisms α and β can be equal only if they share their domain and codomain. For instance, each of the four modulus functions

$$|\cdot|\colon \mathbb{R} \to \mathbb{R}, \qquad |\cdot|\colon \mathbb{R} \to \mathbb{C}, \qquad |\cdot|\colon \mathbb{C} \to \mathbb{R} \quad \text{and} \quad |\cdot|\colon \mathbb{C} \to \mathbb{C}$$

are distinct morphisms, despite representing the 'same' function.

Example 2.2. We list categories which will be important for our purposes, as well as a few illustrative examples. For a more exhaustive list, see [4, p. 293]. We will mostly be working in the categories Λ-Mod of Λ-modules and Ab of abelian groups.

Category	Objects	Morphisms
Set	Sets	Functions
Grp	Groups	Group homomorphisms
Λ-Mod	(left) Λ-modules	Λ-module homomorphisms
Ab $= \mathbb{Z}$-Mod	Abelian groups	Group homomorphisms
K-Vect $=$ K-Mod	Vector spaces over K	Linear transformations
Top	Topological spaces	Continuous maps

In each of these examples the morphisms are functions and the identity morphisms 1_X are the identity maps Id_X. Further, morphism composition is function composition (which we know to be associative).

e.g. $X = \mathbb{N}, \mathbb{Z}, \mathbb{R}$ where $\leq$ has its usual meaning; or $X = \{\text{subsets of } \mathbb{N}\}$ with $\subseteq$ instead of $\leq$.

Note that morphisms need not resemble functions at all! For instance, let $(X, \leq)$ be a partially ordered set. We can turn this into a category by

- taking objects to be elements $x, y, z, \ldots$ of X.

- having precisely one morphism $\varphi\colon x \to y$ if $x \leq y$, or else no such morphism from x to y.

Then a composition of morphisms $\psi\varphi\colon x \to y \to z$ corresponds to the property that if $x \leq y$ and $y \leq z$, then $y \leq z$ (transitivity). Identity morphisms 1_x exist because we require that $x \leq x$ (reflexivity).

2.1.2 Functors

In most of the examples we have mentioned, categories consist of mathematical structures and structure-preserving maps between them. But a category is a type of mathematical structure itself, so how should we define a structure-preserving map between categories?

Definition 2.3. A *functor* F from a category $\mathcal{C}$ to a category $\mathcal{D}$ (written $F\colon \mathcal{C} \to \mathcal{D}$) is a rule which provides:

1. an object $F(C)$ in $\mathcal{D}$ for every object $C \in \mathcal{C}$;
2. a morphism $F(\gamma)\colon F(C) \to F(C')$ in $\mathcal{D}$ for every morphism $\gamma\colon C \to C'$. The result $F(\gamma)$ is called an *induced morphism*.

Brackets may be omitted, so that $F(C) = FC$ and $F(\gamma) = F\gamma$.

Functors are required to send identities to identities, so that $F(1_C) = 1_{F(C)}$. Additionally, a functor must behave like a homomorphism if we give it composable morphisms. We allow functors to reverse the order of composition, meaning that

$$F(\gamma\gamma') = F(\gamma)F(\gamma') \text{ for all composable } \gamma, \gamma', \tag{2.1a}$$
$$\text{or } F(\gamma\gamma') = F(\gamma')F(\gamma) \text{ for all composable } \gamma, \gamma'. \tag{2.1b}$$

If F satisfies (2.1a) (resp. (2.1b)) it is said to be *covariant* (resp. *contravariant*). In this case we may write γ_* (resp. γ^*) for the induced morphism $F(\gamma)$.

Note that is it possible that a functor is co- and contravariant simultaneously.

* * *

Commutative diagrams can also be constructed in categories, where we use objects as vertices and morphisms as arrows. Functors are useful because they map a commuting diagram (i.e. a system of morphism equations) in one category to a commuting diagram in another. Take a square, for instance.

$$\begin{array}{ccc} Y & \xrightarrow{\alpha} & A \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\varphi} \\ B & \xrightarrow{\psi} & X \end{array} \quad \rightsquigarrow^{F} \quad \begin{array}{ccc} F(Y) & \xrightarrow{F(\alpha)} & F(A) \\ \downarrow{\scriptstyle F(\beta)} & & \downarrow{\scriptstyle F(\varphi)} \\ F(B) & \xrightarrow{F(\psi)} & F(X) \end{array}$$

(If F is contravariant, the arrows on the right-hand diagram become reversed.)

Example 2.4. We describe two trivial functors and one not-so-trivial functor.

1. Every category has an identity functor $\mathrm{Id}_{\mathcal{C}}\colon \mathcal{C} \to \mathcal{C}$ which maps an object to itself and a morphism to itself.
2. Many categories' objects are sets embellished with additional structure—forgetful functors U simply forget about this structure. Take Grp (groups and homomorphisms) for example. The functor $U\colon \mathsf{Grp} \to \mathsf{Set}$ maps groups $(G, \cdot)$ to their underlying set G. Homomorphisms $\varphi\colon G \to H$ are unaltered by U; though we think of $U(\varphi)$ as an ordinary map between sets rather than a homomorphism.
3. Let G be a group and consider all possible quotient groups of G. The Abelianisation of G, denoted by $\mathrm{Ab}(G)$, is the largest of these quotients which is Abelian. We can construct an Abelianisation functor $\mathrm{Ab}\colon \mathsf{Grp} \to \mathsf{Ab}$ by modifying homomorphisms accordingly. If $f\colon G \to H$ is a homomorphism, then $\mathrm{Ab}(f)\colon \mathrm{Ab}(G) \to \mathrm{Ab}(H)$ maps a coset $[g]$ to $[f(g)]$.

These three functors are all covariant. We close this section by introducing an important pair of functors—one covariant, the other contravariant.

Often a category's morphisms are homomorphisms, hence the name 'Hom'.

Example 2.5 (The Hom functors)**.** The *hom-set* $\mathrm{Hom}(A, B)$ is the set of morphisms $\varphi\colon A \to B$ in some given category $\mathcal{C}$. For example, take $\mathcal{C} = \mathsf{Ab}$ and $A = \mathbb{Z}$. A (homo)morphism $\varphi\colon \mathbb{Z} \to B$ is completely determined by $\varphi(1)$, which we may choose to be any element in B. So $\mathrm{Hom}(\mathbb{Z}, B)$ is in bijective correspondance with B.

To make Hom into a functor, we must define how to create its induced morphisms. We do so by a kind of partial evaluation. Fix A and choose a morphism $\beta\colon B \to B'$. We define the induced morphism β_* by

$$\begin{aligned} \beta_* = \mathrm{Hom}(A, \beta)\colon \mathrm{Hom}(A, B) &\to \mathrm{Hom}(A, B') \\ \varphi &\mapsto \beta \circ \varphi. \end{aligned}$$

Note that β_* accepts φ as an argument, whereas the original morphism β *composes* with φ.

To check that this gives us a functor, we must verify two conditions. First, if $\beta = 1_B$ we see that $(1_B)_*$ maps morphisms φ to $1_B \circ \varphi = \varphi$. Hence $(1_B)_*$ is the identity map on $\mathrm{Hom}(A, B)$. Secondly, let $\beta'\colon B' \to B''$ be a second morphism. Then the induced morphisms compose according to

$$\begin{aligned} \beta'_*\beta_*(\varphi) = \beta'_*(\beta \circ \varphi) = \beta' \circ (\beta \circ \varphi) \\ \text{and} \quad (\beta'\beta)_*(\varphi) = (\beta' \circ \beta) \circ \varphi. \end{aligned}$$

Strictly speaking, the fact that Hom gives us sets (rather than 'classes') is an axiom of category theory.

We observe that $\beta'_*\beta_* = (\beta'\beta)_*$, since composition of morphisms is associative. This means that $\mathrm{Hom}(A, -)\colon \mathcal{C} \to \mathsf{Set}$ is a covariant functor, given any fixed choice of object A.

On the other hand, suppose we fix B and work with a homomorphism $\alpha\colon A' \to A$. We define α^* by

$$\begin{aligned} \alpha^* = \mathrm{Hom}(\alpha, B)\colon \mathrm{Hom}(A, B) &\to \mathrm{Hom}(A', B) \\ \varphi &\mapsto \varphi \circ \alpha. \end{aligned}$$

Once again 1_A^* becomes the identity map on $\mathrm{Hom}(A, B)$, sending φ to $\varphi \circ 1_A = \varphi$. The behaviour of a composition is slightly different, however. Let $\alpha'\colon A'' \to A'$ be a second morphism; then we have

$$\begin{aligned} \alpha'^*\alpha^*(\varphi) = \alpha'^*(\varphi \circ \alpha) = (\varphi \circ \alpha) \circ \alpha' \\ \text{and} \quad (\alpha\alpha')^*(\varphi) = \varphi \circ (\alpha \circ \alpha'). \end{aligned}$$

This time we see that $\alpha'^*\alpha^* = (\alpha\alpha')^*$—the composition order reverses. This means that $\mathrm{Hom}(-, B)\colon \mathcal{C} \to \mathsf{Set}$ is a *contra*variant functor.

2.1.3 Bifunctors

The hom-sets give us two different Hom functors: one acting on the left, and one acting on the right. The fact that these two functors are separated like this suggests that we might be able to combine them into one big functor. First we explain how to combine categories.

Definition 2.6. Let $\mathcal{C}$ and $\mathcal{D}$ be two categories. The *product category* $\mathcal{C} \times \mathcal{D}$ consists of:

- Objects: ordered pairs (C, D) of objects C, D of $\mathcal{C}$ and $\mathcal{D}$, respectively.
- Morphisms: ordered pairs of morphisms (γ, δ) where γ, δ are morphisms in $\mathcal{C}$ and $\mathcal{D}$, respectively.
- Composition: composition occurs elementwise, so that $(\gamma, \delta)(\gamma', \delta') = (\gamma\gamma', \delta\delta')$ whenever the two compositions exist in $\mathcal{C}$ and $\mathcal{D}$.

Details. Identity morphisms are given by $1_{(C,D)} = (1_C, 1_D)$. The composition is associative because it inherits the associativity of $\mathcal{C}$ in the left entry and $\mathcal{D}$ in the right entry.

The corresponding notion of a 'combined functor' follows from this category. Such functors are given their own name.

Definition 2.7. A *bifunctor* is a functor $F\colon \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ from a product of categories to a category.

There's nothing special about the number two: we may define tri- and multi-functors analogously.

Bifunctors may be contra- or covariant in each argument independently. In the discussion to follow, we work only with covariant functors—though everything we describe applies to contravariant functors and mixed co/contravariant bifunctors. (This is permissible since it is possible to view any functor as a covariant functor by using 'opposite categories'.)

Bifunctors can be specialised to form ordinary single-argument functors. This is done in the same manner as for Hom: by making a partial evaluation that fixes one of the arguments. On the left, we can make $F(C, -)$ into a functor by mapping a morphism $\delta\colon D \to D'$ to $\delta_* = F(1_C, \delta)$. On the right, $F(-, D)$ becomes a functor when we map $\gamma\colon C \to C'$ to $\gamma_* = F(\gamma, 1_D)$. The properties required for these restrictions to be functors all follow from the properties of the original bifunctor F.

This process doesn't just give us two functors: we obtain two *families* of functors: $\{F(C, -)\}_{C\in\mathcal{C}}$ and $\{F(-, D)\}_{D\in\mathcal{D}}$. What about the opposite process—given two families of functors, under what circumstances can they be merged to form a single bifunctor?

Lemma 2.8 (Bifunctor from two functors)**.** *Let $F\colon \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ be a rule taking pairs of objects (C, D) to an object $F(C, D)$, and let $F(C, -)$ and $F(-, D)$ be functors for all choices of C and D.*

See exercise III.2.7 of [3] for more details.

The only way to make F into a bifunctor is to define $F(\gamma, \delta) = \gamma_*\delta_*$, and this works if and only if γ_* and δ_* commute for all morphisms $\gamma\colon C \to C'$, $\delta\colon D \to D'$.

$$\begin{array}{ccc} F(C, D) & \xrightarrow{\delta_*} & F(C, D') \\ \downarrow{\scriptstyle \gamma_*} & & \downarrow{\scriptstyle \gamma_*} \\ F(C', D) & \xrightarrow{\delta_*} & F(C', D') \end{array}$$

Proof. Suppose there *is* some definition of $F(\gamma, \delta)$ making F a bifunctor. By rewriting a pair of morphisms as a composition, we conclude that

$$F(\gamma, \delta) = F[(\gamma, 1_D) \circ (1_C, \delta)] = F(\gamma, 1_D)F(1_C, \gamma) = \gamma_*\delta_*.$$

But we also have that

$$F(\gamma, \delta) = F[(1_C, \delta) \circ (\gamma, 1_D)] = F(1_C, \gamma)F(\gamma, 1_D) = \delta_*\gamma_*,$$

hence it is necessary that $F(\gamma, \delta) = \gamma_* \delta_* = \delta_* \gamma_*$.

To see that this is sufficient, let us try defining the induced map by $F(\gamma, \delta) = \gamma_* \delta_*$. There are two properties to check. Firstly, the identity morphism $(1_C, 1_D)$ must induce the identity morphism $1_{(C,D)}$. Observe that

$$\begin{aligned} F(1_C, 1_D) &= 1_{C*} 1_{D*} \\ &= F(1_C, D) F(C, 1_D) \\ &= 1_{F(C,D)} 1_{F(C,D)} && \text{(Partial evaluations are functors)} \\ &= 1_{F(C,D)} \end{aligned}$$

Hence the identity induces the identity.

We must also check how F applies to a composition. We compute

$$\begin{aligned} F(\gamma\gamma', \delta\delta') &= (\gamma\gamma')_* (\delta\delta')_* \\ &= \gamma_* \gamma'_* \delta_* \delta'_* && \text{(Partial evaluations are functors)} \\ &= \gamma_* \delta_* \gamma'_* \delta'_* && \text{(Commutativity of induced maps)} \\ &= F(\gamma, \delta) F(\gamma', \delta'). \end{aligned}$$

Thus we have made F into a bifunctor. □

Now we confirm that Hom is a bifunctor.

Proposition 2.9. $\mathrm{Hom}(-,-)$ *is a bifunctor in any category. It is contravariant in its first argument and covariant in its second argument.*

Proof. By the previous lemma, we need only to examine whether or not the induced morphisms α^* and β_* commute. Let $\varphi \colon A \to B$ be a morphism. Computing directly from our definitions of induced morphisms, we have

$$\begin{aligned} \alpha^* \beta_*(\varphi) &= \alpha^*(\beta \circ \varphi) = (\beta \circ \varphi) \circ \alpha \\ \text{and} \quad \beta_* \alpha^*(\varphi) &= \beta^*(\varphi \circ \alpha) = \beta \circ (\varphi \circ \alpha). \end{aligned}$$

Since morphism composition is associative, we see that $\alpha^* \beta_* = \beta_* \alpha^*$. □

2.2 More categorical tools

We shall show later how $E(-,-)$ and $\mathrm{Ext}(-,-)$ are bifunctors. The construction of Ext's induced morphisms will require some further machinery. We also need to define how functors may relate to one another, so that we can compare E and Ext.

2.2.1 Pullbacks and pushouts

Pullbacks and pushouts are devices used to 'fill out' a commutative square. Suppose we have two fixed morphisms φ and ψ which share their codomain X. Can we find a pair of morphisms α, β making the following square commute?

That is: can we find morphisms such that $\varphi\alpha = \psi\beta$?

$$\begin{array}{ccc} Y & \xrightarrow{\alpha} & A \\ \downarrow{\scriptstyle \beta} & & \downarrow{\scriptstyle \varphi} \\ B & \xrightarrow{\psi} & X \end{array} \tag{2.2}$$

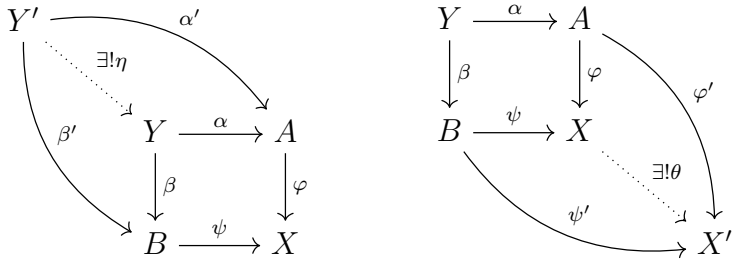
If we were working in Λ-Mod or Grp for instance, we could choose Y freely and take α and β to be zero homomorphisms. But this is cheating—is there a best choice of α and β? Let us specify what we would like 'best' to mean.

Definition 2.10 (Pullback)**.** Suppose that (2.2) commutes. The pair (α, β) is a *pullback* of (φ, ψ) if for any pair of morphisms (α', β') with $\varphi\alpha' = \psi\beta'$ there is a unique morphism η such that $\alpha' = \alpha\eta$ and $\beta' = \beta\eta$. (We call η a *connecting morphism*.)

The condition that there must exist a unique η or θ is known as the *pullback/pushout property*.

This question has a dual counterpart: given α and β, is there a best choice of morphisms φ, ψ making the square commute?

Definition 2.11 (Pushout)**.** Suppose that (2.2) commutes. The pair (φ, ψ) is a *pushout* of (α, β) if for any pair of morphisms (φ', ψ') with $\varphi'\alpha = \psi'\beta$ there is a unique morphism $\theta\colon X \to X'$ such that $\varphi' = \theta\varphi$ and $\psi' = \theta\psi$. (We call θ a *connecting morphism*.)



Pullback Pushout (2.3)

In other words, if another commutative square can be made of the two given edges, it can be 'glued onto' the pullback/pushout square.

Pullbacks need not exist in a given category. However, if a pullback does exist it must be unique up to isomorphism. Briefly, we can use the pullback property to construct two morphisms from each of the pullbacks into the other. Then we may use the uniqueness condition to deduce that these morphisms are each others' inverses. The same idea shows that pushouts are unique up to isomorphism (if they exist).

Remark 2.12. It follows from the uniqueness condition on η that

$$(\alpha\eta = \alpha\eta' \quad \text{and} \quad \beta\eta = \beta\eta') \implies \eta = \eta'.$$

This may be rewritten as

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\eta = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}\eta' \implies \eta = \eta'.$$

Hence the uniqueness condition for the pullback implies that $\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right)$ may be cancelled from the left of a morphism equation. For the pushout, the dual result is that (φ, ψ) must be right-cancellable.

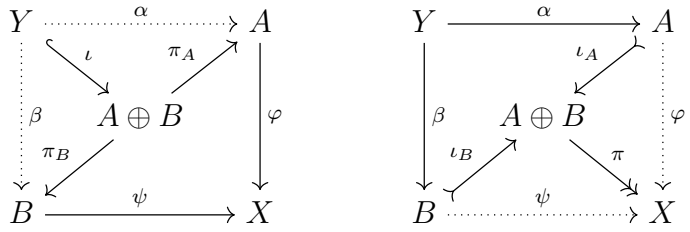
In Λ-Mod, these statements are equivalent to $\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right)$ being injective and (φ, ψ) being surjective. See [3, proposition I.6.1, I.6.2] for the details.

Let us now prove that these objects exist in Λ-Mod by constructing an explicit example. We continue to use the notation of (2.3).

Proposition 2.13 (Constructing a pullback and pushout)**.** *The set* $Y = \{(a, b) : \varphi(a) = \psi(b)\}$ *is a pullback of* (φ, ψ) *when equipped with the projections* $\alpha = \pi_A$*,* $\beta = \pi_B$ *(restricted to* Y*).*

Let $N = \mathrm{Im}\binom{\alpha}{-\beta} = \{(\alpha(y), -\beta(y)) : y \in Y\}$*. Then the quotient module* $X = (A \oplus B)/N$ *is a pushout of* (α, β)*. The associated homomorphisms are the inclusion maps* $\varphi = \iota_A$*,* $\psi = \iota_B$ *(modified to map into the quotient).*

Proof. First note that Y really is a module as it is the kernel of the homomorphism $(\varphi, -\psi)$ out of the direct sum. The homomorphisms α and β make the square commute, since $\varphi\alpha(a, b) = \varphi(a) = \psi(b) = \psi\beta(a, b)$. Let a connecting homomorphism be given by $\eta(y) = (a_y, b_y)$. To ensure $\alpha\eta = \alpha'$ we must take $a_y = \alpha'(y)$; similarly $b_y = \beta'(y)$. This uniquely determines η.



(2.4)

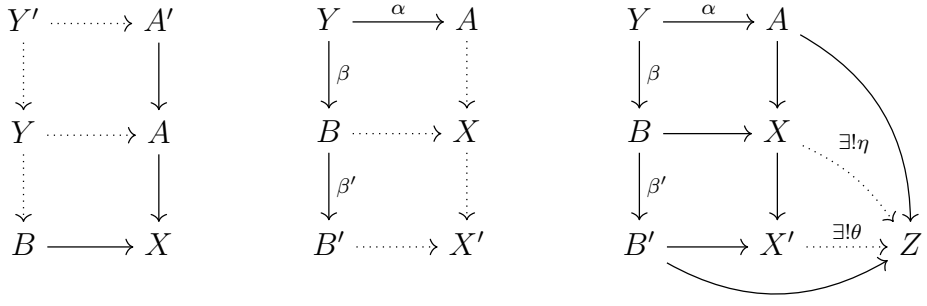
For the pushout, note that $N = \mathrm{Im}\binom{\alpha}{-\beta}$ is the image of a homomorphism and thus may be factored out of of the direct sum. Inside the quotient X, we have $\varphi\alpha(y) = (\alpha(y), 0) = (0, \beta(y)) = \psi\beta(y)$, so the square commutes. A connecting homomorphism θ must satisfy $\theta(a, 0) = \psi'(a)$ and $\theta(0, b) = \psi'(b)$, according to the two triangles in the pushout diagram (2.3). Since we wish θ to be a homomorphism, this forces $\theta(a, b) = \varphi'(a) + \psi'(b)$.

Strictly speaking $\varphi(a)$ is the coset $(a, 0) + N$, and similarly for $\psi(b) = (0, b) + N$.

We must check that θ is well-defined. Suppose (a, b) and (a', b') are equal in X. Then $(a, b) - (a', b') = (\alpha(y), -\beta(y))$ for some $y \in Y$. Applying θ to this equation yields $\theta(a, b) - \theta(a', b') = (\varphi\alpha - \psi\beta)(y) = 0$. Hence $\theta(a, b) = \theta(a', b')$. □

We conclude with a lemma on composing two pullbacks or pushouts.

Lemma 2.14 (Composing pullbacks/pushouts)**.** *Suppose two pullback/pushout squares are joined together as shown. Then the composite rectangle is also a pullback/pushout square.*



Proof (pushout). The outer rectangle $YAX'B'$ commutes because the two squares commute. Suppose we have morphisms $A \to Z$ and $B' \to Z$ such that $YAZB'$ commutes. The pushout property of the upper square gives us a unique morphism $\eta\colon X \to Z$ such that $A \to X \to Z = A \to Z$ and $B \to X \to Z = B \to B' \to Z$.

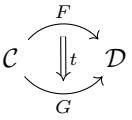
Then the pushout property of the lower square gives us a unique morphism $\theta\colon X' \to Z$ making all components of the diagram commute. Hence $YAX'B'$ is a pushout of $(\alpha, \beta'\beta)$. □

2.2.2 Natural transformations

How can we compare one functor to another? This will depend on how each functor maps both its objects and its morphisms. Is there a relation which respects all this information?

Definition 2.15 (Natural transformation)**.** Let F and G be two functors $\mathcal{C} \to \mathcal{D}$. A *transformation* $t\colon F \Rightarrow G$ is a family of morphisms $\{t_C\colon FC \to GC\}_{C\in\mathcal{C}}$. If every morphism t_C is an isomorphism, t is called an *equivalence*. We may write t for t_C if it makes things clearer.

This is sometimes illustrated as:



A transformation is *natural* if the following diagram commutes for all objects C and all morphisms $\gamma\colon C \to C'$ out of C.

$$\begin{array}{ccc} FC & \xrightarrow{t_C} & GC \\ \downarrow{\scriptstyle \gamma_*} & & \downarrow{\scriptstyle \gamma_*} \\ FC' & \xrightarrow{t_{C'}} & GC' \end{array}$$

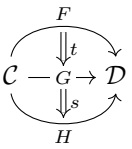
That is, a transformation is natural if it makes no difference whether we transform before or after applying an induced map.

We may think of a natural transformation as a homomorphism between functors; then natural equivalences are the analogue of isomorphisms.

Remark 2.16 (Identities, composition and equivalences). Choose a functor F in any category $\mathcal{C}$. The identity natural transformation $1_F\colon F \Rightarrow F$ consists of morphisms which are all identities, speficically $(1_F)_C = 1_C$.

We may compose transformations $t\colon F \Rightarrow G$ and $s\colon G \Rightarrow H$ to form $st\colon F \Rightarrow H$, whose morphisms are $(st)_C = s_C \circ t_C$. It can be shown that a transformation t is an equivalence if and only if it has an inverse s such that the compositions ts and st are both identity transformations.

We can represent this composition as:



For a meatier example, see section 3.3 where we create a natural equivalence between E and Ext. In the meantime, we prove a lemma which will be needed for its construction.

Lemma 2.17 (Natural iff natural in both arguments)**.** *Suppose we have bifunctors $F, G\colon \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ and a transformation $t\colon F \Rightarrow G$ between them. Then t is natural if and only if it is natural in $\mathcal{C}$ and in $\mathcal{D}$.*

Proof. Consider the following diagram.

$$\begin{array}{ccc} F(C,D) & \xrightarrow{t_{C,D}} & G(C,D) \\ \downarrow{\scriptstyle (\gamma,1)_*} & & \downarrow{\scriptstyle (\gamma,1)_*} \\ F(C',D) & \xrightarrow{t_{C',D}} & G(C',D) \\ \downarrow{\scriptstyle (1,\delta)_*} & & \downarrow{\scriptstyle (1,\delta)_*} \\ F(C',D') & \xrightarrow{t_{C',D'}} & G(C',D') \end{array}$$

(with $(\gamma,\delta)_*$ on the left and right sides)

'Natural in C' means that for any object $D \in \mathcal{D}$ the restricted transformation $t_{-,D}$ is natural. If this is the case, the top square commutes. Likewise, 'natural in D' means that the bottom square commutes. Then the composite rectangle commutes. But the vertical edges of this rectangle are the same as the dotted arrows, so the outer square must commute. This means that t is natural.

The other direction is easier. If t is a natural transformation, any square of the same form as the outer dotted square commutes. But then we can choose $\gamma = 1_C$ and $\delta = 1_D$ to see that the top and bottom squares commute, respectively. Hence t is natural in C and in D. □

Chapter 3

Extensions and the bifunctor Ext

We are now equipped with the tools to fully define E and Ext. After doing so, we verify that they are bifunctors. We conclude the project by constructing a natural equivalence between E and Ext.

3.1 The bifunctor E

3.1.1 Technical Lemmas

Both of the following lemmas come in two versions: one for the pullback and one for the pushout. Hilton and Stammbach prove the pullback versions in [3, section III.1], so we prove the pushout versions here.

Lemma 3.1 (Extensions from pullback/pushout)**.** *If the left square in (3.1) is a pullback, then* $\ker\alpha \cong \ker\psi$ *and* α *is surjective if* ψ *is surjective. If the same square is a pushout, then* $\operatorname{coker}\alpha \cong \operatorname{coker}\psi$ *and* ψ *is injective if* α *is injective.*

$$\begin{array}{ccccc} Y & \xrightarrow{\alpha} & A & \xrightarrow{\rho} & A/\operatorname{Im}\alpha \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\varphi} & & \vdots{\scriptstyle\theta} \\ B & \xrightarrow{\psi} & X & \xrightarrow{\pi} & X/\operatorname{Im}\psi \end{array} \tag{3.1}$$

Proof (pushout). Let $[a]$ and $[x]$ denote cosets in the two quotient modules. Define θ by $\theta[a] = [\varphi(a)]$. To see this is well-defined, choose two representatives a and a' of the same coset. Then $a - a' = \alpha(y)$ for some y, meaning $[\varphi(a)] = [\varphi a' + \varphi\alpha y] = [\varphi a'] + [\psi\beta(y)] = [\varphi(a')]$ (because $\operatorname{Im}\psi$ is factored out by π).

Next, θ is a homomorphism since $\theta[a + a'] = [\varphi(a + a')] = [\varphi(a)] + [\varphi(a')] = \theta[a] + \theta[a']$. Without loss of generality (WLOG), let $X = (A \oplus B)/\operatorname{Im}\binom{\alpha}{-\beta}$ be the pushout constructed in proposition 2.13. If $[a]$ belongs to $\ker\theta$, we have $\varphi(a) = (a, 0)$ is equal in X to $\psi(b) = (0, b)$ for some $b \in B$. Then $(a, -b) = (\alpha(y), -\beta(y))$ for some $y \in Y$. In particular, $a \in \operatorname{Im}\alpha$ means that $[a]$ is the zero coset. So θ is injective.

Recall that $\varphi(a) = (a, 0) + \operatorname{Im}\binom{\alpha}{-\beta}$ and $\psi(b) = (0, b) + \operatorname{Im}\binom{\alpha}{-\beta}$.

Because π and (φ, ψ) are surjective, their composition $(\pi\varphi, \pi\psi) = (\pi\varphi, 0)$ is surjective too. Thus $\pi\varphi$ is surjective, so any $[x]$ looks like $[x] = [\varphi(a)] = \theta[a]$ for some a. Hence θ is surjective.

Finally, suppose α is injective and suppose $\psi(b) = 0$. Use the specific pushout X, again WLOG. Then $(0, b) \in \operatorname{Im}\binom{\alpha}{-\beta}$, so $0 = \alpha(y)$ and $b = \beta(y)$ for some

We can restore full generality by using the fact that any two pushouts are isomorphic via connecting homomorphisms.

$y \in Y$. But α is injective, so $y = 0$ and hence $b = 0$, meaning ψ is injective also. □

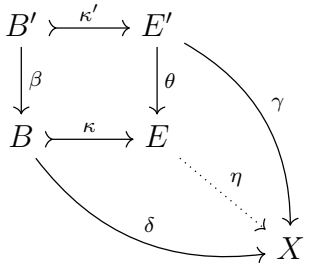
Suppose that the bottom row of (3.1) is short exact, i.e. an extension. The lemma then says that the top row is also an extension. The next lemma works in the other direction.

Lemma 3.2. *Let the following diagram be commutative with exact rows.*

$$\begin{array}{ccccc} B' & \overset{\kappa'}{\rightarrowtail} & E' & \overset{\nu'}{\twoheadrightarrow} & A' \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\theta} & & \downarrow{\scriptstyle\alpha} \\ B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \end{array}$$

If $B' = B$ and $\beta = 1_B$ then the right square is a pullback. If $A' = A$ and $\alpha = 1_A$ then the left square is a pushout.

Proof (pushout). Given any $e \in E$, there is an element $e' \in E'$ with $\nu(e) = \nu'(e')$ by the surjectivity of ν'. Then $\nu(e) = \nu\theta(e')$, meaning $e - \theta(e') \in \ker \nu$. This means $e = \theta(e') + \kappa(b)$ for some $b \in B$.



Now we demonstrate how the pushout property arises from the diagram. Suppose we have maps γ and δ with $\gamma\kappa' = \delta\beta$. If the two triangles above are to commute, then $\eta\theta(e') = \gamma(e')$ and $\eta\kappa(b) = \delta(b)$. Thus if a connecting homomorphism exists, it must be given by $\eta(e) = \eta(\theta(e') + \kappa(b)) = \gamma(e') + \delta(b)$. It's straightforward to check that η is a homomorphism provided we know η is well-defined.

Suppose e has two representations $e = \theta(e'_1) + \kappa(b_1) = \theta(e'_2) + \kappa(b_2)$. Then

$$\theta(e'_1 - e'_2) + \kappa(b_1 - b_2) = 0. \tag{3.2}$$

Applying ν to this equation gives $\nu\theta(e'_1 - e'_2) = \nu'(e'_1 - e'_2) = 0$, so $e'_1 - e'_2 = \kappa'(b')$ for some $b' \in B'$. Substituting into (3.2), we see that $\theta\kappa'(b') + \kappa(b_1 - b_2) = \kappa(\beta(b') + b_1 - b_2) = 0$. Because κ is injective, it follows that $b_1 = b_2 - \beta(b')$. This is enough to show that that the two images of e under η are identical.

$$\begin{aligned} \gamma(e'_1) + \delta(b_1) &= \gamma[e'_2 + \kappa'(b')] + \delta[b_2 - \beta(b')] \\ &= \gamma(e'_2) + \delta(b_2) + \underbrace{(\gamma\kappa' - \delta\beta)}_{=0}(b') \end{aligned}$$

Thus η is well-defined. □

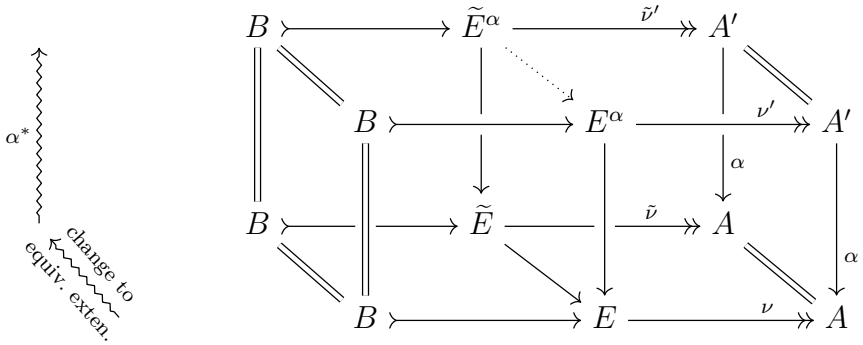
3.1.2 Induced maps

Let $\alpha\colon A' \to A$ be a homomorphism. Suppose we are given an extension $\nu\colon B \overset{\kappa}{\rightarrowtail} E \overset{\nu}{\twoheadrightarrow} A$ representing an extension class $[\nu] \in E(A, B)$. To define the induced map α^*, construct the pullback E^α of (α, ν).

$$\begin{array}{ccccc} B & \overset{\kappa'}{\rightarrowtail} & E^\alpha & \overset{\nu'}{\twoheadrightarrow} & A' \\ \| & & \downarrow{\scriptstyle\theta} & & \downarrow{\scriptstyle\alpha} \\ B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \end{array} \tag{3.3}$$

Apply lemma 3.1. We add the homomorphism κ' to diagram (3.3), which maps B isomorphically onto $\ker \nu'$ and then includes the result into E^α. The lemma also tells us that ν' is surjective, and thus the top row is an extension $\nu'\colon B \overset{\kappa'}{\rightarrowtail} E^\alpha \overset{\nu'}{\twoheadrightarrow} A'$. We define $\alpha^*[\nu] = [\nu']$ to be this extension's equivalence class.

We should check that α^* is well-defined. Suppose we perform the above procedure on an extension $\tilde{\nu}$ equivalent to ν. We obtain diagram (3.4). The back-right square is already a pullback; the bottom-right square is a pullback by lemma 3.2. So their composite rectangle is a pullback of E (lemma 2.14). Hence $\widetilde{E}^\alpha$ is isomorphic to E^α (via the dotted arrow). Looking at the top squares, we see that ν' and $\tilde{\nu}'$ are equivalent extensions.



(3.4)

Have we made a functor out of $E(-, B)$? Set $A' = A$ and $\alpha^* = 1_A$. Then in diagram (3.3) we immediately see that 1_A^* is the identity map on extension classes. Next, a composition $\alpha'^*\alpha^*$ of induced maps would create two pullbacks as in figure 3.1. Then the composite rectangle would be a pullback too, meaning $(E^\alpha)^{\alpha'}$ is isomorphic to the pullback $E^{(\alpha\alpha')}$ created by $(\alpha\alpha')^*$. We conclude that $\alpha'^*\alpha^* = (\alpha\alpha')^*$ and thus $E(-, B)$ is a contravariant functor.

* * *

Given $\beta\colon B \to B'$, the process of constructing $\beta_*[\nu]$ is similar. This time we form the pushout of (κ, β) to obtain the module E_β. Using lemma 3.1 again, we know that κ' is injective and that we may add the morphism ν', forming an extension along the bottom row. We define $\beta_*[\nu] = [\nu']$. A argument similar to that of α^* (involving pushouts rather than pullbacks) shows that β_* is well-defined.

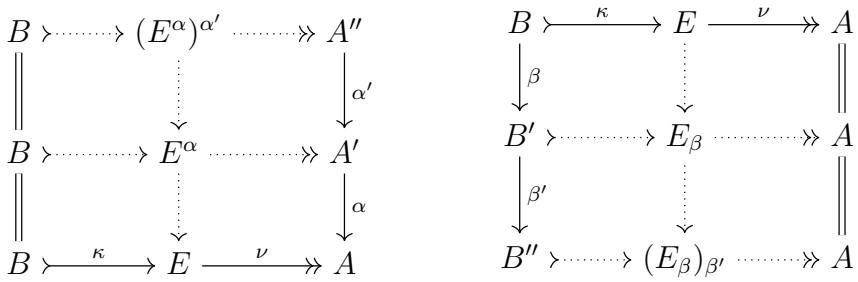
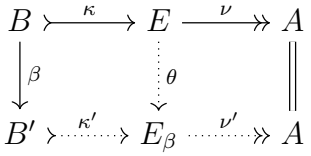


Figure 3.1: Diagrams showing how the induced maps compose. On the left, α^* composes contravariantly, whereas β_* composes covariantly.



Does this give us a functor? As before, set $B' = B$ and $\beta = 1$ to conclude that 1_* is the identity map on $E(A, B)$. Consulting figure 3.1 again, we form two pushouts in constructing the composition $\beta'_*\beta_*$. The composite rectangle is then a pushout, meaning $(E_\beta)_{\beta'} \cong E_{(\beta\beta')}$. Thus $\beta'_*\beta_* = (\beta'\beta)_*$ which tells us that $E(A, -)$ is a covariant functor.

3.1.3 Bifunctorality

One last question about E remains: is it a bifunctor?

Theorem 3.3. *$E(-,-)$ is a bifunctor from Λ-Mod to Set. It is contravariant in its first argument and covariant in its second.*

Sketch proof. By lemma 2.8, we just need to show that $\alpha^*\beta_* = \beta_*\alpha^*$. This involves showing that extensions involving $(E_\beta)^\alpha$ and $(E^\alpha)_\beta$ are equivalent. Hilton and Stammbach do this directly in [3, theorem III.1.4] with a diagram chase.

We shall omit the details. However, in the sections to follow we prove that Ext is a bifunctor and we construct a natural equivalence between E and Ext. By imposing Ext's bifunctor structure onto E, we can show indirectly that E is a bifunctor.

3.2 The bifunctor Ext

The functor E is somewhat clumsy to work with. For instance, there is no 'formula' for $E(A, B)$ that specifies all extension classes. We introduce Ext to address this. First we must adapt the idea of a group presentation to modules.

3.2.1 Projective presentations

A group presentation $\langle X \mid R \rangle$ formally stands for the quotient group $F(X)/N(R)$. Here $F(X)$ is the 'free group', consisting of strings $x_{i_1}^{\pm 1} \dots x_{i_n}^{\pm 1}$ made from the

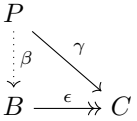
symbols $x_i \in X$. This group has no relations—no x_i interacts with any other x_j. We introduce relations by factoring out $N(R)$, the smallest normal subgroup containing the relator set R.

We need to something like a 'free module'. In fact the more general idea of a projective module is sufficient.

Definition 3.4. Let A be a set. The *free* Λ*-module* $F(A)$ is the direct sum $\bigoplus_{a\in A} \Lambda$ of copies of Λ.

Compare to free Abelian groups.

Definition 3.5. A Λ-module P is *projective* if given any two Λ-module homomorphisms $\epsilon\colon B \twoheadrightarrow C$ and $\gamma\colon P \to C$ with ϵ surjective there is a third homomorphism $\beta\colon P \to B$ with $\epsilon\beta = \gamma$.



Definition 3.6. A *projective presentation* of a Λ-module A is a short exact sequence $R \overset{\mu}{\rightarrowtail} P \overset{\epsilon}{\twoheadrightarrow} A$ with P projective.

Compare to $N(R) \hookrightarrow F(X) \twoheadrightarrow G$

Before proceeding, we demonstrate that every module has a presentation—albeit not a very useful one.

Proposition 3.7. *The free module $P = F(A)$ is projective for any set A.*

Proof. Let e_a be the tuple with 1 at position a and 0 at all other positions. Any element in P is a linear combination $\sum_{a\in A} \lambda_a e_a$ with only finitely many λ_a nonzero.

Compare to the standard basis vectors e_i of $\mathbb{R}^n$.

Assume the setup of definition 3.5. Because ϵ is surjective, there is an element b_a such that $\epsilon(b_a) = \gamma(e_a)$ for every index $a \in A$. Define β by requiring that $\beta(e_a) = b_a$. This uniquely specifies β, since $\{e_a\}_{a\in A}$ is a basis for $F(A)$. Explicitly, we have $\beta(\sum_{a\in A} \lambda_a e_a) = \sum_{a\in A} \lambda_a b_a$. Then ϵ maps this to $\sum_{a\in A} \lambda_a \epsilon(b_a) = \sum_{a\in A} \lambda_a \gamma(e_a) = \gamma(\sum_{a\in A} \lambda_a e_a)$. Hence $\epsilon\beta = \gamma$. □

Corollary 3.8. *Every Λ-module A has a projective presentation.*

Proof. Let P be the free Λ-module $F(A)$. Imposing that ϵ satisfies $\epsilon(e_a) = a$ completely determines ϵ in the same way that β was determined above. Then we may construct the sequence $\ker\epsilon \hookrightarrow F(A) \overset{\epsilon}{\twoheadrightarrow} E$ which is (trivially) short exact. □

3.2.2 The Ext groups

Recall the hom-sets of example 2.5. In Λ-Mod, these consist of Λ-module homomorphisms. We can turn such a set $\mathrm{Hom}(A, B)$ into an Abelian group using the addition $(\varphi + \psi)(a) = \varphi(a) + \psi(a)$. The zero homomorphism is the identity; negatives are given by $(-\varphi)(a) = -(\varphi(a))$; and associativity follows from associativity of module addition.

The maps α^* and β_* induced by Hom then become group homomorphisms, and thus Hom becomes a functor to Ab rather than Set. What happens if we apply this new Hom to a short exact sequence?

Proposition 3.9 ($\mathrm{Hom}(-, B)$ is left-exact)**.** *Let $R \overset{\mu}{\rightarrowtail} P \overset{\epsilon}{\twoheadrightarrow} A$ be a short exact sequence of Λ-modules. For every Λ-module B, the induced sequence below is exact.*

$$0 \longrightarrow \mathrm{Hom}(A, B) \overset{\epsilon^*}{\rightarrowtail} \mathrm{Hom}(P, B) \xrightarrow{\mu^*} \mathrm{Hom}(R, B) \tag{3.5}$$

Note that sequence has reversed direction. This happens because $\mathrm{Hom}(-, B)$ is contravariant.

Proof. There are three facts to demonstrate.

1. Let $\tau \in \ker \epsilon^*$. Then $\epsilon^*(\tau) = \tau\epsilon = 0 = 0\epsilon$. Because ϵ is surjective, we may cancel it on the right. So $\tau = 0$, meaning ϵ^* is injective.

2. Let $\tau\colon A \to B$ be a homomorphism (not neccesarily in $\ker \epsilon^*$. Consider the composition $\tau\epsilon\mu$. On the one hand this is $(\tau\epsilon)\mu = \mu^*\epsilon^*\tau$. On the other hand, this is $\tau(\epsilon\mu) = \tau 0 = 0$. Hence $\operatorname{Im}\epsilon^* \subseteq \ker \mu^*$.

3. For the opposite inclusion. Let $\sigma \in \ker \mu^*$, i.e. $\sigma\mu = 0$. Can we use this to find a $\psi\colon A \to B$ such that $\sigma = \epsilon^*(\psi) = \psi\epsilon$?

$$\begin{array}{ccccc} R & \overset{\mu}{\rightarrowtail} & P & \overset{\epsilon}{\twoheadrightarrow} & A \\ & & \downarrow{\scriptstyle \sigma} & \swarrow_{\psi} & \\ & & B & & \end{array}$$

   Given $a \in A$, use the surjectivity of ϵ to choose an element $p \in P$ with $\epsilon(p) = a$; then define $\psi(a) = \sigma(p)$. This makes ψ a homomorphism because τ is a homomorphism. To check that ψ is well-defined, let p and p' both map to a under ϵ. Because we have an exact sequence, $p - p' = \mu(r)$ for some $r \in R$. But $\sigma\mu = 0$, meaning $\sigma(p - p') = \sigma\mu(r) = 0$. Rearranging this tells us that $\sigma(p) = \sigma(p')$, so ψ is well-defined.

   From this we conclude that $\ker \mu^* \subseteq \operatorname{Im}\epsilon^*$. □

The induced sequence (3.5) is exact but not necessarily *short* exact, as μ^* need not be surjective. For example, take the sequence $\mathbb{Z} \overset{\times n}{\rightarrowtail} \mathbb{Z} \overset{\bmod n}{\twoheadrightarrow} \mathbb{Z}_n$ of $\mathbb{Z}$-modules and take $B = \mathbb{Z}_n$. We obtain the induced sequence

$$0 \longrightarrow \operatorname{Hom}(\mathbb{Z}_n, \mathbb{Z}_n) \overset{(\bmod n)^*}{\underset{\epsilon^*}{\rightarrowtail}} \operatorname{Hom}(\mathbb{Z}, \mathbb{Z}_n) \overset{(\times n)^*}{\underset{\mu^*}{\longrightarrow}} \operatorname{Hom}(\mathbb{Z}, \mathbb{Z}_n).$$

Here μ^* accepts a map $\tau\colon \mathbb{Z} \to \mathbb{Z}_n$ and returns $\tau\mu$, which takes z to $\tau(nz) = n\tau(z) = 0$. So $\mu^*(\tau)$ is always the zero homomorphism. But $\operatorname{Hom}(\mathbb{Z}, \mathbb{Z}_n)$ contains non-zero homomorphisms such as $\tau\colon \mathbb{Z} \to \mathbb{Z}_n\colon z \mapsto z \bmod n$.

To get around this lack of surjectivity, we add an extra term to the right end of (3.5) which makes the sequence exact at $\operatorname{Hom}(R, B)$.

Note that Ext depends on μ, which is given by the presentation of A in specified by E.

Definition 3.10 (Ext groups)**.** For every Λ-module A, choose a projective presentation of A. Call the collection of presentations E.

Define $\operatorname{Ext}^E(A, B)$ to be $\operatorname{coker}\mu^*$ where μ^* is defined as in proposition 3.9. Its elements are cosets $[\varphi]$ of morphisms $\varphi\colon R \to B$. Two classes $[\varphi]$ and $[\varphi]'$ are equal if and only if $\varphi - \varphi' = \tau\mu$ for some $\tau\colon P \to B$.

3.2.3 Induced maps

Let $\beta\colon B \to B'$ be a homomorphism. We can extend (3.5) to form a longer exact sequence using Ext groups.

The maps π are canonical projections.

$$\begin{array}{ccccccc} \operatorname{Hom}(A,B) & \overset{\epsilon^*}{\rightarrowtail} & \operatorname{Hom}(P,B) & \overset{\mu^*}{\longrightarrow} & \operatorname{Hom}(R,B) & \overset{\pi}{\twoheadrightarrow} & \operatorname{Ext}^E(A,B) \\ \downarrow{\scriptstyle \beta_*} & & \downarrow{\scriptstyle \beta_*} & & \downarrow{\scriptstyle \beta_*} & & \vdots{\scriptstyle \beta_*} \\ \operatorname{Hom}(A,B') & \overset{\epsilon^*}{\rightarrowtail} & \operatorname{Hom}(P,B') & \overset{\mu^*}{\longrightarrow} & \operatorname{Hom}(R,B') & \overset{\pi}{\twoheadrightarrow} & \operatorname{Ext}^E(A,B') \end{array}$$

The diagram shows two such sequences, connected by the maps β_* induced by the Hom functor. The rightmost β_* is induced by $\mathrm{Ext}^E(A,-)$, and is defined by $\beta_*[\varphi] = [\beta \circ \varphi]$, making the right square commute. Then $\mathrm{Ext}^E(A,-)$ is a covariant functor—the details are similar to those of $\mathrm{Hom}(A,-)$.

We can address the role of E and construct induced maps on the left at the same time. Let $\alpha\colon A' \to A$ be a homomorphism, and choose presentations $R' \overset{\mu'}{\rightarrowtail} P' \overset{\epsilon'}{\twoheadrightarrow} A'$ of A' and $R \overset{\mu}{\rightarrowtail} P \overset{\epsilon}{\twoheadrightarrow} A$ of A. Because P' is projective, there is a homomorphism π making the right square of (3.6) commute. (As shorthand, we say that π lifts α.) Then π induces a unique homomorphism σ making the left square commute.

$$\begin{array}{ccccccc} & & R' & \overset{\mu'}{\rightarrowtail} & P' & \overset{\epsilon'}{\twoheadrightarrow} & A' \\ & \swarrow & \downarrow{\scriptstyle\sigma} & & \downarrow{\scriptstyle\pi} & & \downarrow{\scriptstyle\alpha} \\ B & \underset{\varphi}{\leftarrow} & R & \overset{\mu}{\rightarrowtail} & P & \overset{\epsilon}{\twoheadrightarrow} & A \end{array} \tag{3.6}$$

The presentations of A' and A are allowed to be in different collections E' and E.

To construct σ, define $\sigma(r')$ to be the unique $r \in R$ such that $\mu(r) = \pi\mu'(r')$. We know that a unique such r exists because $\epsilon\pi\mu'(r') = \alpha\epsilon'\mu'(r') = 0$ and the bottom row is exact.

We can use σ to make a map $\pi^*\colon \mathrm{Ext}^E(A,B) \to \mathrm{Ext}^{E'}(A',B)$ defined by $\pi^*[\varphi] = [\varphi \circ \sigma]$, forming the triangle in (3.6). In principle, π^* may depend on both π and α. We show that π^* is actually independent of π.

Lemma 3.11. *Given the setup of (3.6), π^* is the same for every choice of π.*

Proof. Choose two liftings π_1 and π_2 which induce σ_1 and σ_2 respectively. Since $\epsilon(\pi_1 - \pi_2) = \alpha\epsilon' - \alpha\epsilon' = 0$, there must exist $\tau\colon P' \to R$ with $\pi_1 - \pi_2 = \mu\tau$, because the lower sequence is exact. Then $\mu(\sigma_1 - \sigma_2) = (\pi_1 - \pi_2)\mu' = \mu\tau\mu'$. Since μ is injective, we conclude $\sigma_1 = \sigma_2 + \tau\mu'$. To finish, choose $[\varphi]$ in $\mathrm{Ext}^E(A,B)$. Then $\pi_1^*[\varphi] = [\varphi\sigma_1] = [\varphi\sigma_2 + \varphi\tau\mu'] = [\varphi\sigma_2] = \pi_2^*[\varphi]$. □

Recall that $[\varphi] = [\varphi']$ in $\mathrm{Ext}^{E'}(A',B)$ if $\varphi - \varphi' = \gamma\mu'$ for some γ.

The map π^* can be formed for any Λ-module B. Collecting together all these morphisms π^* as B varies gives us a transformation $t\colon \mathrm{Ext}^E(A,-) \Rightarrow \mathrm{Ext}^{E'}(A',-)$. The transformation is natural in B because

$$\pi^*\beta_*[\varphi] = [(\beta \circ \varphi) \circ \sigma] = [\beta \circ (\varphi \circ \sigma)] = \beta_*\pi^*[\varphi].$$

Let us write $(\alpha; E', E)$ for this transformation t.

Suppose we have a second homomorphism $\alpha'\colon A'' \to A'$. Given a presentation of A'' in E'', we can form a second transformation $(\alpha'; E'', E')$ by choosing a suitable lifting $\pi'\colon A'' \to A'$. But then $\pi \circ \pi'$ lifts $\alpha \circ \alpha'$. This means that the transformations compose according to

$$(\alpha'; E'', E') \circ (\alpha; E', E) = (\alpha \circ \alpha'; E'', E). \tag{3.7}$$

Note how the α maps swap.

Furthermore

$$(1_A; E, E) = 1, \tag{3.8}$$

where 1 is the identity transformation $1\colon \mathrm{Ext}^E(A,-) \Rightarrow \mathrm{Ext}^E(A,-)$.

Now we may define induced maps in A. Let $\alpha\colon A' \to A$ and choose presentations of A' and A from the same collection E. Define $\alpha^* = \mathrm{Ext}^E(\alpha, B)$ to be the natural transformation $(\alpha; E, E)$ evaluated at B. (Explicitly, $\alpha^*[\varphi] = [\varphi \circ \sigma]$.) Then $\mathrm{Ext}^E(-,B)$ becomes a covariant functor due to rules (3.7) and (3.8).

Where σ is induced by a choice of π.

3.2.4 Bifunctorality

Theorem 3.12. $\mathrm{Ext}^E(A,B)$ *is a bifunctor from* Λ-Mod *to* Ab. *It is contravariant in* A *and covariant in* B.

Proof. We just need to show that $\alpha^*\beta_* = \beta_*\alpha^*$. We have $\alpha^*\beta_*[\varphi] = [(\beta\circ\varphi)\circ\sigma)] = [\beta\circ(\varphi\circ\sigma)] = \beta_*\alpha^*[\varphi]$. □

Finally, we demonstrate that the choice of E doesn't affect the functor we construct from it. Set $e_A = (1_A; E', E)$. This is the natural transformation which changes the presentation of A.

Lemma 3.13. *The transformation* $e_A\colon \mathrm{Ext}^E(A,-) \Rightarrow \mathrm{Ext}^{E'}(A,-)$ *is an equivalence.*

Proof. Compose the transformation with $(1_A; E, E')$ using rules (3.7) and (3.8). Both compositions give the identity transformation. Hence they are both equivalences (see remark 2.16). □

This gives us a collection $e = \{e_A\}_{A\in\Lambda\text{-Mod}}$ of equivalences—each natural in B—which change E to E'. The collection itself is natural in A in the sense that there is no difference if we change A before or after we change E to E'. Consider the following diagram of transformations.

$$\begin{array}{ccc} \mathrm{Ext}^E(A,-) & \xRightarrow{e_A} & \mathrm{Ext}^{E'}(A,-) \\ \Big\Downarrow{\scriptstyle \alpha^*} & & \Big\Downarrow{\scriptstyle \alpha^*} \\ \mathrm{Ext}^E(A',-) & \xRightarrow{e_{A'}} & \mathrm{Ext}^{E'}(A',-) \end{array}$$

The top path gives $(\alpha; E', E')(1_A; E', E) = (\alpha; E', E)$ and the bottom gives $(1_{A'}; E', E)(\alpha; E, E) = (\alpha; E', E)$—the same result.

Formally e as a transformation is given by $e_{A,B} = (e_A)|_B$, where $|_B$ means 'evaluated at B'.

Thus the collection e describes a natural equivalence $e\colon \mathrm{Ext}^E \Rightarrow \mathrm{Ext}^{E'}$ of bifunctors. In other words, changing the presentations we use gives us a different description of the same functor. This allows us to drop the E from Ext^E.

3.3 The natural equivalence

Our last task is to demonstrate that E and Ext are equivalent. For the moment, we 'forget' that Ext produces Abelian groups, and just work with two set-valued bifunctors.

Main Theorem 3.14. *There is a natural equivalence* $f\colon E(-,-) \Rightarrow \mathrm{Ext}(-,-)$.

Step 1 (Construction of f). First we must provide a family of homomorphisms $\{f_{A,B}\}$ specifying the transformation. Fix a projective presentation $R \overset{\mu}{\rightarrowtail} P \overset{\epsilon}{\twoheadrightarrow} A$ of A. Given an extension $\nu\colon B \overset{\kappa}{\rightarrowtail} E \overset{\nu}{\twoheadrightarrow} A$, we may construct the top half of the

following commutative diagram.

$$\begin{array}{ccccc} R & \overset{\mu}{\rightarrowtail} & P & \overset{\epsilon}{\twoheadrightarrow} & A \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \pi} & & \| \\ B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \\ \| & & \downarrow{\scriptstyle \theta} & & \| \\ B & \overset{\kappa'}{\rightarrowtail} & E' & \overset{\nu'}{\twoheadrightarrow} & A \end{array} \tag{3.9}$$

Add in a lifting π using the projectivity of P; this uniquely induces $\varphi\colon R \to B$ (in the same way that σ was induced in the previous section). We claim that the coset $[\varphi] \in \mathrm{Ext}(A, B)$ is the same for any choice of π. Suppose we have two liftings π_1 and π_2 which induce φ_1 and φ_2, respectively. Follow the proof of lemma 3.11 to see that $\varphi_1 = \varphi_2 + \tau\mu$ for some $\tau\colon P \to B$. Then $[\varphi_1] = [\varphi_2 + \tau\mu] = [\varphi_2]$.

Let us write f for $f_{A,B}$. We would like to define $f[\nu] = [\varphi]$, but is this well-defined? Suppose an extension ν' is equivalent to ν via a map $\theta\colon E \to E'$ as in (3.9). Choose to use the lifting $\pi' = \theta \circ \pi\colon P \to E'$. According to the diagram, π' induces $1_B \circ \varphi = \varphi$. So $f[\nu'] = [\varphi]$, i.e. f is well-defined.

Step 2 (The inverse g). Given a coset $[\varphi]$ in $\mathrm{Ext}(A, B)$, form the pushout E of (φ, μ). Then use lemma 3.1 to create an extension $\nu\colon B \rightarrowtail E \twoheadrightarrow A$.

Any pushout will do, since any two are isomorphic and produce equivalent extensions of A by B.

$$\begin{array}{ccccc} R & \overset{\mu}{\rightarrowtail} & P & \overset{\epsilon}{\twoheadrightarrow} & A \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \pi} & & \| \\ B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \end{array} \tag{3.10}$$

As before, we propose the definition $g[\varphi] = [\nu]$ and check its validity. Suppose φ' is an alternative representative of the coset $[\varphi]$. Then $\varphi' = \varphi + \tau\mu$ for some $\tau\colon P \to B$. Set $\pi' = \pi + \kappa\tau$. If we replace π and φ with their primed counterparts, (3.10) is still commutative: on the left we have $\kappa\varphi' = \kappa\varphi + \kappa\tau\mu = (\pi + \kappa\tau)\mu = \pi'\mu$, and $\nu\pi' = \nu\pi + \nu\kappa\tau = \epsilon + 0 = \epsilon$ on the right.

By lemma 3.2, the left square of (3.10) is a pushout when its vertical edges are the primed maps. Hence $g[\varphi'] = [\nu]$, meaning g is well-defined.

Step 3 (f is an equivalence). The composition fg takes a coset $[\varphi]$ and applies g, obtaining $[\nu]$ as in (3.10). To apply f we must choose a lifting $P \to E$. Let us choose π, which was constructed by applying g. Now π induces a unique map $R \to B$ making (3.10) commute, namely φ. Hence $fg[\varphi] = f[\nu] = [\varphi]$, so $fg = 1$ where 1 stands for the identity transformation $1_{\mathrm{Ext}}\colon \mathrm{Ext}(-,-) \Rightarrow \mathrm{Ext}(-,-)$.

'$fg = 1$' means '$f_{A,B} \circ g_{A,B} = 1_{A,B}$ for every A and B.'

In the other direction, begin with the extension ν and apply f, forming the top half of (3.9). Lemma 3.2 tells us that the top-left square is a pushout. So $gf[\nu] = g[\varphi] = [\nu]$, hence $gf = 1$. This time, 1 stands for $1_E\colon E(-,-) \Rightarrow E(-,-)$.

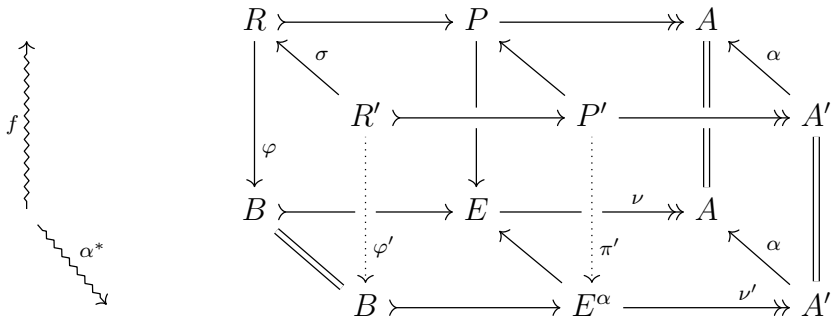
Step 4 (f is natural). First consider naturality in B. Applying f first yields $\beta_* f[\nu] = \beta_*[\varphi] = [\beta \circ \varphi]$. If we apply β_* first, we obtain an extension ν' as below.

$$\begin{array}{ccccc} R & \overset{\mu}{\rightarrowtail} & P & \overset{\epsilon}{\twoheadrightarrow} & A \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \pi} & & \| \\ B & \overset{\kappa}{\rightarrowtail} & E & \overset{\nu}{\twoheadrightarrow} & A \\ \downarrow{\scriptstyle \beta} & & \downarrow{\scriptstyle \theta} & & \| \\ B' & \overset{\kappa'}{\rightarrowtail} & E_\beta & \overset{\nu'}{\twoheadrightarrow} & A \end{array} \tag{3.11}$$

To apply f, we need a lifting $\pi' \colon P \to E_\beta$. We choose $\pi' = \theta \circ \pi$, where π was constructed in applying f to $[\nu]$. This induces a unique homomorphism $\varphi' \colon R \to B'$ making (3.11) commute. But ψ' must be $\beta \circ \varphi$ according to the diagram. Thus $f\beta_*[\nu] = f[\nu'] = [\varphi'] = [\beta \circ \varphi]$.

Now for naturality in A. Begin with the extension class $[\nu]$, displayed as the lower-back row of (3.12). Apply f to obtain the upper-back row and the Ext coset $[\varphi]$. In $\mathrm{Ext}(A, B)$, the induced map α^* gives us the upper-front row and the coset $\alpha^* f[\nu] = \alpha^*[\varphi] = [\varphi \circ \sigma]$.

All squares without any dotted edges commute. (We prove that those with dotted edges must commute too.)



(3.12)

Such a lifting must exist because P' is projective.

For the opposite order, first apply α^* in $E(A, B)$ to obtain the extension class $[\nu']$. To transform via f, we must find a lifting $\pi' \colon P' \to E^\alpha$ and see which map $\varphi' \colon R' \to B$ it induces.

Observe that the compositions $P' \to P \to E \xrightarrow{\nu} A$ and $P' \to A' \xrightarrow{=} A' \xrightarrow{\alpha} A$ are equal. Because E^α is a pullback of (ν, α), there is a (unique) connecting homomorphism π' which makes the central square $PP'E^\alpha E$ and the front-right square $P'A'A'E^\alpha$ commute. Next, π' induces φ', making the front-left square $R'P'E^\alpha B$ commute.

The only square whose commutativity remains undecided is the left end, $RR'BB$. The two compositions $R' \to R \to B \to E$ and $R' \to B \xrightarrow{=} B \to E$ are equal, because each can be 'bent' around the left cube to form the other. Since $B \rightarrowtail E$ is injective, we may cancel it from the equality to conclude that the left end square commutes.

This shows that $\varphi' = \varphi \circ \sigma$ and hence $f\alpha^* = \alpha^* f$, i.e. f is natural in A. Thus f is a natural equivalence.

Finally we check that f is well-defined. Choose $A' = A$ and $\alpha = 1_A$, but let us continue to use (potentially) different presentations A as in (3.12). The fact that f is natural in A tells us that f is independent of the presentation of A. □

3.4 Conclusion and further reading

In this project, we have introduced modules and elementary constructions upon them. We have seen how extensions may be formed and illustrated the difficulty

of classifying extensions directly. This last result allows us to use the 'formula' for $\mathrm{Ext}(A, B)$ to investigate $E(A, B)$.

Furthermore, we can use the equivalences of section 3.3 to impose the group operation of Ext onto E. A direct description of this operation was first given in 1934 by Baer. The details of the 'Baer sum' are outlined in [3, ch. III, ex. 2.5–2.7] and [8].

Even with the Baer sum and knowledge that extension classes come in groups (and Abelian groups at that), it is still often easier to compute $E(A, B)$ indirectly. To this end, Hilton and Stammbach provide tools for working with Ext in [3, sec. III.4, III.5]. There is a notion of extensions for groups, too, but different tools are needed to describe them; see [3, sec. VI.10].

Ext does much more than we have described here. It occurs in a much more general setting as a 'derived functor'; in fact there are many such functors $\{\mathrm{Ext}^1, \mathrm{Ext}^2, \ldots, \mathrm{Ext}^n, \ldots\}$. In certain circumstances, these Exts may be interpreted as describing n-extensions of modules; [3, ch. IV] describes the theory.

Chapter 4

Bibliography

[1] Max Dehn. Über unendliche diskontinuierliche gruppen. *Mathematische Annalen*, 71(1):116–144, 1911.

[2] Samuel Eilenberg and Saunders Mac Lane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[3] P.J. Hilton and U. Stammbach. *A Course in Homological Algebra*. Graduate Texts in Mathematics. Springer-Verlag, second edition, 1997.

[4] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer, second edition, 1998.

[5] Pyotr Sergeyevich Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Matematicheskogo Instituta im. VA Steklova*, 44:3–143, 1955.

[6] Michael Potter. *Set Theory and its Philosophy*. Oxford, 2004.

[7] C.A. Weibel. *An Introduction to Homological Algebra*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1995.

[8] C.A. Weibel. The history of homological algebra. In I.M. James, editor, *The History of Topology*, chapter 28, pages 797–836. Elsevier, 1999. Preprint available at `http://www.math.uiuc.edu/K-theory/0245/`.