MA4JB Commutative Algebra II

===== Overview of homological algebra =====

Everything in the rest of the course involves homological algebra in some form or another: Complexes, exact sequences, what to do when the operations we need break exactness.

== Colloquial overview of Abelian categories ==
Let M,N be modules over a ring, and f: M -> N a homomorphism. We know sub-objet, quotient object ker, image, cokernel, sometimes coimage.

  ker f in M, the quotient M/(ker f) "coimage"
  im f in N, the cokernel N/im f

the map f: M -> N can be broken down into
  0 -> ker f -> M -> M/(ker f) -> 0,
  an isomorphism M/(ker f) iso im f,
  0 -> im f -> N -> coker f -> 0.   (*)

(This is mostly familiar, except that you may not have seen "coimage" = M/(ker f) before -- an artificial construction.)
This is a bit like the rank-nullity result of first year linear algebra: write down some linear equations. How many solutions we get depends on whether the equations are linearly independent, and so on.

There is a level of abstraction even before that: the set Hom(M,N) between objects is an Abelian group under addition (or an R-module), and the direct sum M + N is a module with

maps in i1: M -> M+N given by m |-> (m,0) and similarly i2 for N

maps out p1: M+N -> M given by (m,n) |-> m and similarly p2 for N

together with identifications p1.i1 = id M, p2.i1 = 0, and a few more.

Whenever we use direct sum with these properties, we are in an _additive category_: the Hom sets are Abelian groups of R-modules, categorical product and coproducts M+N as above are defined and coincide (or similar with two objects M,N replaced by finitely many objects).

An _Abelian category_ is an additive category with ker and im, coimage and coker having the properties (*). Whenever you say that a complex of modules is an exact sequence, you are working in an Abelian category, whether you know what that is or not. "An Abelian category is a category satisfying just enough axioms so that the snake lemma holds."

For our purposes, there is no need to pay special attention to these issues, because we only work with modules over a ring. The categorical stuff consists of tautologies that we use all the time. Under appropriate set-theoretic assumptions, it is a theorem that every

Abelian category is equivalent to a category of modules over a ring.

I currently work only with modules, and not in abstract category theory. There are more general abstract categories, where morphisms are not viewed as maps of sets, and all the definitions, starting from 0 and what it means for a morphism to be the inclusion of a suboject, or to be an epimorphism, need rethinking from the ground up.

===== Entry point to homological algebra: the Hom functor ====

The Hom functor Hom_A(-,N) is a contravariant functor in its first entry. It is _left exact_. Its failure to be right exact corresponds to _extensions_, that are controlled by a new functor Ext^1.

Category of modules over a ring R. The functor Hom_R(-,N) takes
  an object M |-> the R-module Hom_R(M,N)
(consisting of R-homomorphisms M -> N), and takes
  a homomorphism M1 -al-> M2 |-> the R-homomorphism (alpha)
    al^*: Hom(M2,N) -> Hom(M1,N),
that consists of composing with al. That is, compose f: M2 -> N with the given al, to get f.al: M1 -> M2 -> N. Functor means compatibility with compositions: al^*.be^* = (be.al)^* and with identity morphisms al^*.id^* = al^*.

Hom(-,N) is a minor generalisation of the dual of a vector space over a field k, where Hom_k(-,k) takes V to its dual V^dual and a k-linear map U -M-> V to its ajoint or transpose Mt: V^dual -> U^dual.

Lemma Hom(-,N) is left-exact. That is, if we apply Hom(-,N) to a s.e.s.
  0 -> A -al-> B -be-> C -> 0
we get an exact sequence
  0 -> Hom(C,N) -be^*-> Hom(B,N) -al^*-> Hom(A,N). (2)

Solemn proof. For f in Hom(C,N) if the composite f.be is zero then f is zero. Because take c in C, lift it to b in B, then f(b) = f.be(c) = 0. The argument is trivial, given that B ->> C is surjective.

Next, exactness at the middle: the composite al^*.be^* = (be.al)^* = 0 so the sequence (2) is a complex. To say that g: B -> N is in the kernel of al^* means that g(b) is well-defined on the coset of b modulo the image of al. This means that if we lift c in C to an element b in B, then apply g to b, we get g(b) in N that does not depend on the choice of lift. This gives a well-defined morphism gbar: C -> N
  c |-> (choice of b) |-> g(c) := g(b)
with be^*(gbar) = g, which proves exactness at the middle.

This was all long-winded and trivial. The key point however: there is no reason why (2) must be exact at the right end: why should an R-module homomorphism A -> N extend to B -> N? This fails in familiar cases:

(1) Consider

  0 -> A -al-> B -> C -> 0 with A = ZZ, B = ZZ and C = ZZ/p
where the first map is multiplication by p. Set N = ZZ/p and consider
the functor Hom(-, N). There is a perfectly nice map p: A -> N that
sends a |-> a mod p. This cannot be of the form g.al^* for any g, since
this takes b to g(p*b) = p*g(b), and multiplication by p takes every
element of N to zero.

(2) In a similar vein, let R = k[x, y]_m be the localisation of k[x,y]
at the maximal ideal m = (x,y), and work in the category of R-modules.
Consider 0 -> A -al-> B -> C -> 0 with al the inclusion m in R and
C = R/m the residue field. Consider the functor Hom(-,N) where N = k.

Now a homomorphism g: B -> N = k necessarily vanishes on the submodule
A in B, because g(x.1) = x*g(1) = 0 in N and ditto for y.

On the other hand, there are plenty of nice nonzero homomorphisms
m -> k (the dual vector space m/m^2). None of these can be
restriction of any g, so that al^* is certainly not surjective.

(3) A wider view of these examples: let I in R be an ideal f: I -> N be
a nonzero homomorphism to an I-torsion module, for example M/IM for an
R-module M. A homomorphism R -> N necessarily vanishes on I, so that it
is certainly not possible to extend the given f: I -> N to a
homomorphism F: R -> N.

===== Failure of exactness gives Ext^1 ====

Consider again a s.e.s. of R-modules
  0 -> A -al-> B -be-> C -> 0.
We get the exact sequence
  0 -> Hom(C,N) -> Hom(B,N) -> Hom(A,N)

Given f: A -> N, construct the _pushout_ diagram

```
0 -> A -> B  -> C -> 0
     |    |     |
     v    v     v
0 -> N -> B' -> C -> 0
```

where B' = (B + N)/im(al, f). If the bottom row is a split s.e.s. of
R-modules (this means B' = N + C, with arrows the inclusion and
projection of the direct sum), we know how to extend f to B by including
B in B' then projecting the direct sum to its first factor.
Exercise: Please think about how to prove the converse.

In the same set-up, one can show that the class of the bottom row
  0 -> N -> B' -> C -> 0
up to isomorphism of s.e.s. is determined by f in Hom(A,N) modulo the
image of al^*(Hom(B,N)). I do not press this point, except to say that
this explains the notation Ext^1(C,N): we can identify the cokernel of
be^* with extensions of C by N.

Summary of narrative so far: Categories, exact sequences. If applying a reasonable functor break exactness, we introduce derived functors such as Ext^1(-,N) to understand the lack of exactness and get some profit from it.

Given a s.e.s.
```
   0 -> A -> B -> C -> 0
```
and a module N, Homming into N gives
```
   0 -> Hom(C,N) -> Hom(B,N) -> Hom(A,N) ->
     -> Ext^1(C,N)
```
In other words, there is a new module Ext^1(C,N) that measures the failure of right exactness. It is a kind of _derived_ Hom. This gives the flavour of what a right derived functor is and does.

===== Projective modules ====

Definition. Let P be an R-module. P is _projective_ if for every surjective homomorphism f: M -> N -> 0 and every homomorphism g: P -> N, there exists a lift G: P -> M, such that f.G = g. As a diagram:
```
    M -f-> N -> 0
     ^     ^

     G \   | g
           P     given f and g, there exist G
                 making the triangle commute.
```

As well as the contravariant form discussed above, Hom_R(M,-) is a covariant functor in its first argument, and is automatically left exact whatever M (please do this as an easy exercise). The condition that P is projective is equivalent to Hom(P,-) an exact functor:
  if 0 -> A -> B -> C -> 0 is a s.e.s.
  then Hom(P,B) ->> Hom(P,C). This just says that a homorphism
  to C can be lifted via B, which is just the projective assumption.

Example-Prop. (1) If P is free then it is projective.
(2) P is projective if and only if P is a direct summand of a free module.
(3) Over a local ring (R,m), a finite projective module P is free. Therefore, a finite projective module is locally free: its localisation P_p at each prime ideal of R is free.
(4*) The converse.
(5) Over a graded ring (graded in positive degrees), a finite graded module that is projective as a graded module is free.

(1) In fact, if P has a basis e_la, take n_la = g(e_la) in N, then lift each n_la to m_la in M with f(m_la) = n_la. We can then define G by setting G(e_la) = m_la. This determines where G takes the basis elements, and R-linearity gives the rest: an element sum a_la.e_la in P maps to sum a_la.m_la. (This works because there are no R-linear relations between the e_la, so we can map then to any elements of M we choose. The argument is exactly the same as for vector spaces.)

(2) If P+Q (direct sum) is free, a map g: P -> N gives rise to
(g,0): P+Q -> N, that we can lift to M by (1), so P is projective.
For the converse, suppose that P is generated by {e_la}. This means that
the map f: M = sum R.f_la -> P from the free module M to P is
surjective. Now suppose P is projective, and consider the identity map
id: P -> P. Applying the definition of projective to it gives G: P -> M.
But now G.f: P -> P is the identity, whereas f.G: M -> M is idempotent
(because f.G.f.G = f.G when we cancel the middle G.f).

Thus M = im(f.G) + ker(f.G) is a direct sum decomposition of the free
module M as P+Q with P = f.G(M) and Q = ker(f.G). QED

(3) A minimal (finite) set of generators of P gives a surjective
homomorphism f: F = A^{oplus n} ->> P. The projective assumption gives a
lift g: P -> F of f, so that F = g(P) direct sum K, with K = ker f. However,
by minimality a relation between the generators cannot have any invertible
coefficients, so the coefficients must be in m. Then K in m*A^n so
K in mK. Then mK = K, so K = 0 by Nakayama's lemma.

(5) is a minor variation on the same proof.

Counterexample (projective but not free): If OK is the ring of integers
of a number field K/Q, and I a fractional ideal, then by definition I is a
free OK-module if and only if it is principal. This usually fails.
However, I is always a locally free OK module. Locally free implies
projective by (4*).

Proof of (4*)
Eisenbud Prop 2.10 on compatibility between localisation and Hom:

A and B and A-algebra
Hom_A(M,N) is an A-module, so B tensor_R Hom_A(M,N) makes sense

Now there is a B-module homomorphism

B tensor_R Hom_A(M,N) -> Hom_B(B tensor_A M, B tensor_A N)

MOREOVER if B is flat over A and M is finitely presented, it
is an isomorphism.

for P in Spec A the localisation LP is free as AP module. So construct
the lift LP -g-> M3_P
Because everything is finite, the construction of g only involves finitely
many denominators, so there is s in A \ P so that g: L[1/s] -> M3[1/s] as
module homomorphism over A[1/s]. Now the same holds at every P in Spec A.
So Spec A is covered by principal open sets (Spec A)_s so that a lift g_s
is defined. The difference g_s1 and g_s2 on the intersection of the two
sets is a homo L[1/s1s2] -> M1[1/s1s2] (the kernel of M2 -> M3.)
[Get into the same argument as structure sheaf of Spec A, coherent modules

over Spec A and coherent H^i = 0 on affine scheme.]

M finitely presented
N2 ->> N3 surjective
  Hom_A(M,N2) -> Hom_A(M,N3) -> coker

In practice, we are mostly interested in local rings or graded rings,
so we almost always work with free modules.