#### Flat Modules

Jujian Zhang

 $-\otimes M$  is always right exact

$$0 \longrightarrow A \xrightarrow{f} B \xrightarrow{g} C \longrightarrow 0$$

$$0 \longrightarrow A \otimes_{R} M \xrightarrow{f \otimes 1} B \otimes_{R} M \xrightarrow{g \otimes 1} C \otimes_{R} M \longrightarrow 0$$

...but not always left exact

For example  $\mathbb{Z}/(2)$  as an  $\mathbb{Z}$ -module

$$0 \longrightarrow \mathbb{Z} \stackrel{\times 2}{\longrightarrow} \mathbb{Z} \longrightarrow \mathbb{Z}/(2) \longrightarrow 0$$

gives

### When is tensoring exact?

```
protected def ses : Prop :=
-\otimes_{\mathbf{R}} M is exact
                                                                                    (1)
                            (tensor_right (Module.of R M)).is_exact
                            protected def inj : Prop :=
                            \forall {|N N' : Module.{u} R|} (L : N \longrightarrow N'),
                                                                                    (2)
-\otimes M preserves mono
                              function.injective L \rightarrow
                              function.injective
                                 ((tensor_right (Module.of R M)).map L)
                            protected def ideal : Prop :=
I \otimes_R M \to R \otimes_R M is
                            \forall (I : ideal R).
                                                                                    (3)
injective for all ideals I
                              function.injective (tensor_embedding M I)
I \otimes_R M \to R \otimes_R M is
                            protected def fg_ideal : Prop :=
injective for all finitely
                            \forall (I : ideal R), I.fg \rightarrow
                                                                                    (4)
generated ideals I
                              function.injective (tensor_embedding M I)
```

```
Fix an arbitrary ideal 1. The set of all finitely generated subideals of 1 is directed
with respect to \leq, i.e. for any two finitely generated subideals J, J', there is
        another finitely generated subideals larger than both, namely J \sqcup J'.
(2)
                                            instance : is_directed
        @[ext]
                                                 (fg_subideal I) (<) :=
                                            { directed := \lambda J J',
        structure fg_subideal :=
(3)
        (to_ideal : ideal R)
                                               \langle\langle J.to\_ideal \sqcup J'.to\_ideal,
        (fg : to_ideal.fg)
                                                 submodule.fg.sup J.fg
        (le : to_ideal \leq I)
                                                J'.fg,
                                                 sorry), sorry) }
(4)
```

Lemma (ideal as colimit of finitely generated subideals)

$$I \cong \underset{J \leq I}{\mathsf{colim}} J$$

where J runs over all finitely generated subideals.

- (2) Proof
- **₩** (3) By using universal property of colimit over directed system, colim  $I \to I$  can
- be realised by lifting the obvious linear map  $J \rightarrow I$ . For the other direction, if
  - $i \in I$ , then the principal subideal  $\langle i \rangle$  is finitely generated, so there is a map  $\langle i \rangle \rightarrow \operatorname{colim} i J$ .
- (4)
- Corollary

 $I \otimes M \cong (\operatorname{colim}_{J \leq I} J) \otimes_R M$ 

```
def as_direct_limit :=
        module.direct_limit (\lambda (i : fg_subideal I), i.to_ideal) $
          \lambda i j hij, (submodule.of_le hij : i.to_ideal \rightarrow_{l}[R] j.to_ideal)
(1)
        def from as direct limit :
          I.as_direct_limit \rightarrow_{I}[R] I :=
       module.direct_limit.lift R _{-} _{-} (\lambda i, submodule.of_le i.le) $
(2)
          \lambda i j hij r, rfl
        @[simps]
(3)
        def to as direct limit :
          I \rightarrow_I [R] I.as\_direct\_limit :=
        \{ to fun := \lambda r.
(4)
            module.direct_limit.of R (fg_subideal I) (\lambda i, i.to_ideal)
               (\lambda _ h, submodule.of_le h) (principal_fg_subideal r)
               (r, mem_principal_fg_subideal r),
          ..sorry }
```

#### Lemma

colimits over direct system commutes with tensor.

(1) Proof

Consider  $(\operatorname{colim}_{i \in \mathcal{T}} i) \otimes_R M$  and  $\operatorname{colim}_{i \in \mathcal{T}} (i \otimes_R M)$ . The forward direction is

done by using universal property of tensor product, we construct a bilinear map  $(\operatorname{colim}_{i \in \mathcal{I}} i) \to M \to \operatorname{colim}_{i \in \mathcal{I}} (i \otimes_R M)$ : for each  $i \in \mathcal{I}$  and  $x \in i$ , there is a

map  $M \xrightarrow{-\otimes x} M \otimes i \longrightarrow \operatorname{colim}_{i \in \mathcal{I}} (i \otimes_R M)$ . The other direction is by

descending the family of maps  $i \otimes_R M \to (\operatorname{colim}_{i \in \mathcal{I}} i) \otimes_R M$  for all  $i \in \mathcal{I}$ .

Corollary

 $I \otimes M \cong \operatorname{colim}_{J \leq I} (J \otimes_R M)$ .









```
(1)
        def
             direct_limit_of_tensor_product_to_tensor_product_with_direct_limit
(2)
          direct_limit (\lambda i, G i \otimes [R] M)
             (\lambda \text{ i j hij, tensor_product.map (f _ hij) (linear_map.id)}) \rightarrow_I
             [R] (direct limit G f) \otimes [R] M :=
(3)
        direct_limit.lift _{-} _{-} _{-} (\lambda \text{ i, tensor_product.map (direct_limit.of})}
             _ _ _ _ _) linear_map.id) $
        \lambda i j hij z, sorry
(4)
```

**=** 1 -- 2 ---- 3 ---- 4 5 - 6 -

```
def
(1)
             tensor_product_with_direct_limit_to_direct_limit_of_tensor_product
        (direct\_limit G f) \otimes [R] M \rightarrow_{\ell} [R] direct\_limit (\lambda i, G i \otimes [R] M)
(2)
          (\lambda \text{ i j hij, tensor_product.map} (f _ hij) (linear_map.id)) :=
        tensor_product.lift \$ direct_limit.lift _ _ _ (\lambda i.
        { to_fun := \lambda g.
(3)
          { to fun := \lambda m, direct_limit.of R \iota (\lambda i, G i \otimes[R] M)
                  (\lambda i j hij, tensor_product.map (f _ hij) (linear_map.id))
             i \$ g \otimes_t m,
(4)
             ..sorry }) ..sorry }
```

**= 1 -- 2 ----- 3 ----- 4 5 - 6 -**

(1)It is a calculation away to show that

(2)

 $\operatorname{colim}_{J \leq I} (J \otimes_R M)$ 

(3)

(4)

where  $\bar{\iota}$  is obtained via the family of maps  $J \otimes_R M \to R \otimes_R M$ . Since each

 $J \otimes_R M \to R \otimes_R M$  is injective,  $\bar{\iota}$  is injective as well, hence  $I \otimes_R M \to R \otimes_R M$ 

is injective.

- We define the character module of M to be  $M^* := \text{Hom}_{\mathbb{Z}}(M, \mathbb{Q}/\mathbb{Z})$ .  $M^*$  is an (1)
- R-module by  $(r \bullet f)(m) := f(r \bullet m)$ . Let  $L : N \to N'$ , then  $L^* : N'^* \to N^*$ defined by  $L^* := - \circ L$  makes taking character module a contravariant functor.
- Theorem
  - If L is a monomorphism then  $L^*$  is an epimorphism.
- (3) Proof.
  - Let  $g \in N^* = \text{Hom}_{\mathbb{Z}}(N, \mathbb{Q}/\mathbb{Z})$ , since  $\mathbb{Q}/\mathbb{Z}$  is an injective group,  $g : N \to \mathbb{Q}/\mathbb{Z}$
- factors as  $N \xrightarrow{L} N' \xrightarrow{\bar{g}} \mathbb{Q}/\mathbb{Z}$  so that  $L^*(\bar{g}) = g$ . (4)

= 1 -- 2 ----- 3 ----- 4 5 - 6 -

**= 1 -- 2 ----- 3 ----- 4 5 - 6 -**

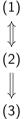
Theorem (injective cogenerator)

For every  $M \ni m \neq 0$ , there is some  $h \in M^*$  such that  $h(m) \neq 0$ .

Proof.

It is sufficient to construct a map  $h': \langle m \rangle \to \mathbb{Q}/\mathbb{Z}$  such that  $h'(m) \neq 0$  by injectivity of  $\mathbb{Q}/\mathbb{Z}$ . Either the additive order of m is finite or infinite:

- ▶ If m has finite order n, then h' is defined as  $k \bullet m \mapsto \frac{k}{n} \mod 1$ . This is well-defined because if  $k \bullet m = k' \bullet m$  then  $k - k' \mid n$  thus  $\frac{k}{n} \equiv \frac{k'}{n} \mod 1$ .
- ▶ If m has infinite order, then h' is defined as  $k \bullet m \mapsto \frac{k}{37}$ . In this case, it is well defined because the choice of k is unique.





(4)

```
lemma non_zero \{m : M\} (hm : m \neq 0) : \exists (h : character_module M),
(1)
            h m \neq 0 :=
        begin
          let M': submodule \mathbb{Z} M:= submodule.span \mathbb{Z} {m}.
(2)
          suffices : \exists (h' : M' \rightarrow_I[\mathbb{Z}] rat_circle), h' \langlem,
↓ (3)
             submodule.subset_span (set.mem_singleton _) \neq 0,
          { sorry }.
          by_cases h_order : add_order_of m \neq 0,
          { sorry },
          { sorry },
(4)
        end
```

#### Theorem (Tensor-Hom adjunction)

- Let R, S be two commutative rings and X and R, S-bimodule, then (1) $-\otimes_R X \dashv \operatorname{Hom}_S(X,-)$ , where for R-module Y,  $Y \otimes_R X$  has the S-module **(2)** structure given by  $s \bullet (v \otimes x) := v \otimes (s \bullet x)$  and for S-module Z. Hom<sub>s</sub>(X, Z) has the R-module structure given by  $r \bullet I := x \mapsto I(r \bullet x)$ .
  - Proof
- Let Y be an R-module and Z an S-module. Any  $I: Y \otimes X \to Z$  also gives a (3)map  $Y \to \operatorname{Hom}_S(X, Z)$  by  $y \mapsto x \mapsto I(y \otimes x)$ . Conversely, for any R-linear
  - map  $I: Y \to \operatorname{Hom}_S(X, Z)$ ,  $y \otimes x \mapsto I(y)(x)$  defines an S-linear map.
- Corollary (4)

 $\operatorname{Hom}_R(N, M^*) \cong (N \otimes_R M)^*$ 

= 1 -- 2 ----- 3 ---- 4 5 - 6 -

16/24

```
@[simps]
        def hom_equiv.inv_fun' {Y : Module.{v} R'} {Z : Module.{v} S'} (1 :
             Y \rightarrow_{\iota} \lceil R' \rceil (X' \rightarrow_{\iota} \lceil S' \rceil Z)) :
(1)
          ((tensor_functor R' S' X').obj Y \longrightarrow Z) :=
        { to_fun := (add_con_gen _).lift (free_add_monoid.lift $ show Y ×
            X' \rightarrow Z, from \lambda p, 1 p.1 p.2) $
(2)
             add_con.add_con_gen_le \lambda p p' (h : eqv R' Y X' p p'), _,
           ..sorry }
(3)
        def tensor_product.lift {R : Type u_1} [comm_semiring R]
          {M : Type u_4} {N : Type u_5} {P : Type u_6}
           [add_comm_monoid M] [add_comm_monoid N]
(4)
           [add comm monoid P] [module R M]
           [module R N] [module R P]
           (f : M \rightarrow_{I} [R] N \rightarrow_{I} [R] P) : tensor\_product R M N \rightarrow_{I} [R] P
```

**=** 1 -- 2 ----- 3 ----- 4 5 - 6 -

- Theorem (Lambek)
  - If  $M^*$  is injective, then M is flat.
- Proof.
  - Let A, B be R-modules and L:  $A \rightarrow B$  an injective R-linear map. If  $A \otimes_R M \ni z \neq 0$  is in the kernel of  $L \otimes 1$ , then there would be some
- (1) (2) (3)  $g \in (A \otimes M)^*$  such that  $g(z) \neq 0$ . Let  $f: A \to M^*$  be defined as
  - $a \mapsto m \mapsto g(a \otimes m)$ , since  $M^*$  is injective and L is mono, f factors through  $f': B \to M^*$ , let  $g' \in (B \otimes_R M)^*$  be the corresponding map of f' under
    - bijection  $\operatorname{Hom}_R(B, M^*) \cong (B \otimes_R M)^*$ . By writing z as  $\sum_i a_i \otimes m_i$ , since
- (4)  $(L \otimes 1)(z) = 0$ , we derive  $g'(\sum_i L(a_i) \otimes m_i) = g'(0) = 0$ .

= 1 -- 2 ----- 3 ----- 4 5 - 6 -

#### Theorem (Lambek)

If  $M^*$  is injective, then M is flat.

Proof.

(2)

(1)

(3)

(4)

$$g'\left(\sum_{i}L(a_{i})\otimes m_{i}\right)=\sum_{i}g'(L(a_{i})\otimes m_{i})=\sum_{i}f'(L(a_{i}))(m_{i})$$

$$=\sum_{i}f(a_{i})(m_{i})=\sum_{i}g(a_{i}\otimes m_{i})$$

$$=g\left(\sum_{i}a_{i}\otimes m_{i}\right)=g(z)\neq0.$$

This is contradiction, thus z must be zero which means  $\ker(L \otimes 1) = 0$ .

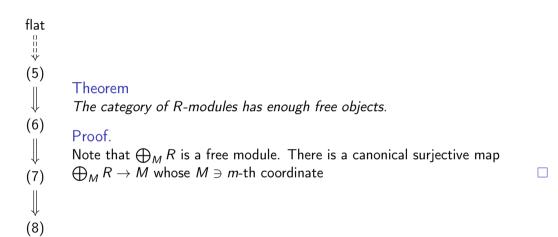
- Corollary
  - If  $\iota: I \otimes_R M \to R \otimes_R M$  is injective for all ideals I, then M is flat.
- Proof
- By Baer's criterion, it is sufficient to show the restriction map
- (3)  $\operatorname{Hom}_R(R, M^*) \to \operatorname{Hom}_R(I, M^*)$  is surjective for all ideals I. Fix an ideal I, let
  - $f: I \to M^*$  corresponding to  $f' \in (I \otimes_R M)^*$ . Since  $\iota$  is injective,  $\bar{\iota}$  is
- surjective so that there is an  $F \in (R \otimes_R M)^*$  such that  $\bar{\iota}(F) = f'$ . Then F
- induces the required  $Hom_R(R, M^*)$ . (4)

#### In terms of Tor functor

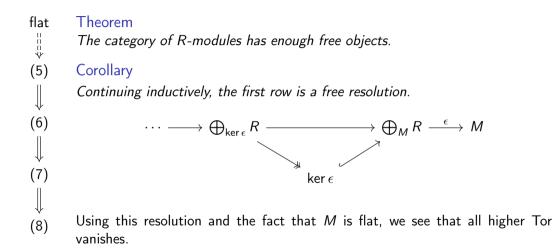
```
\forall (n : \mathbb{N}) (hn : 0 < n) (N : Module.{u} R),
\operatorname{Tor}_{i}^{R}(N,M)\cong 0 for all
                                                                                                 (5)
                                  nonempty
R-modules N and 1 < i
                                  (((Tor' (Module.{u} R) n).obj N).obj M \cong 0)
\operatorname{\mathsf{Tor}}^R_1(\mathsf{N},\mathsf{M})\cong 0 for all \forall (N : Module.{u} R), nonempty
                                                                                                 (6)
                                  (((Tor' (Module.{u} R) 1).obj N).obj M \cong 0)
R-modules N
                               \forall (I : ideal R), nonempty
\operatorname{Tor}_1^R(R/I,M)\cong 0 for
                                  (((Tor' (Module.{u} R) 1).obi
                                                                                                 (6)
all ideals I
                                     (Module.of R (R / I))).obj M \cong 0)
\operatorname{Tor}_{1}^{R}(R/I, M) \cong 0 \quad \forall \text{ (I : ideal R), nonempty}
                                  (((Tor' (Module.{u} R) 1).obj
                                                                                                 (8)
for all finitely generated
                                     (Module.of R (R / I))).obj M \cong 0)
ideals I
```

**= 1 -- 2 ---- 3 ---- 4 5 - 6 -**

20/24



```
flat
(5)
        Theorem
        The category of R-modules has enough free objects.
(6)
        def afree : Module.{u} R := Module.of R $ ⊕ (m : M), R
        def from afree : M.afree \longrightarrow M :=
       direct_sum.to_module _ _ _ $\( \lambda\) m, { to_fun := \lambda\) r, r \cdot m \..sorry}
(7)
(8)
```



```
flat
        def free_res.chain_complex.Xd_aux :
        \mathbb{N} \to
        Σ' (N_prev N_next : Module.{u} R)
(5)
          (h : module.free R N_prev ∧ module.free R N_next),
          N_next \longrightarrow N_prev :=
        @nat.rec
          (\lambda_{-}, \Sigma') (N_prev N_next : Module.{u} R)
(6)
             (h : module.free R N_prev ∧ module.free R N_next),
            N_{next} \longrightarrow N_{prev}
        (M.afree, (kernel M.from_afree).afree,
          (Module.afree_is_free _, Module.afree_is_free _),
          Module.from_afree \rightarrow kernel.\iota \rightarrow $ \lambda n P,
        \langle P.2.1, (kernel P.2.2.2).afree, \langle P.2.2.1.2, Module.afree_is_free__\rangle,
(8)
            Module.from_afree _ >> kernel.t _>
```

### (8) implies flatness

Let I be any finitely generated ideal, consider the exact sequence flat  $(5) \Longrightarrow (6) \Longrightarrow (7) \Longrightarrow$  $0 \rightarrow I \rightarrow R \rightarrow R/I \rightarrow 0$ . def ses\_of\_ideal (I : ideal R) : short\_exact\_sequence (Module.{u}) R.) :={ fst := Module.of R I, snd := Module.of R R. trd := Module.of R (R / I),f := Module.of\_hom  $\langle coe, \lambda \_\_, rfl, \lambda \_\_, rfl \rangle$ , g := submodule.mkq I, mono' := sorry, epi' := sorry, (8)exact' := sorry }

#### (8) implies flatness

Let I be any finitely generated ideal, consider the exact sequence

flat 
$$0 \to I \to R \to R/I \to 0$$
,

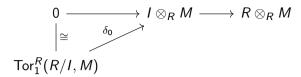
(5) it induces

$$\cdots \to \operatorname{Tor}_1^R(R/I, M) \xrightarrow{\delta} I \otimes_R M \to R \otimes_R M \to R/I \otimes_R M.$$

(6) def  $\delta_0$  (A: short\_exact\_sequence C) :=  $\delta$  F 0 A >> (left\_derived\_zero\_iso\_self F).hom.app A.1

[7] lemma seven\_term\_exact\_seq (A: short\_exact\_sequence C): exact\_seq D [
 (F.left\_derived 1).map A.f, (F.left\_derived 1).map A.g,  $\delta_0$  F A,
 F.map A.f, F.map A.g, (0: F.obj A.3)]

In particular,



is exact. Since this is true for any finitely generated ideal I, (8) implies (4). So finally, all definitions are equivalent.

