MATHEMATICS FOR INFORMATION SCIENCE

NO.12 CATEGORY THEORY AND DATA TYPE

Tatsuya Hagino

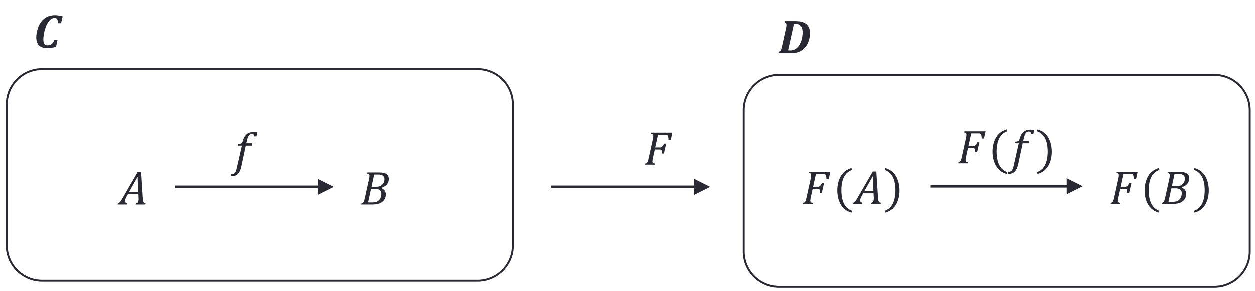
hagino@sfc.keio.ac.jp

Slides URL

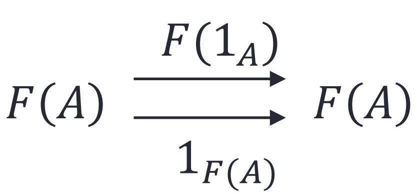
https://vu5.sfc.keio.ac.jp/slide/

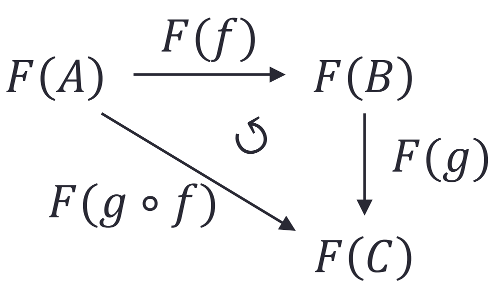
Functor

- For categories $\boldsymbol{C}$ and $\boldsymbol{D}$, a functor $F: \boldsymbol{C} \to \boldsymbol{D}$ is:
  - For any object $A \in \boldsymbol{C}$, $F(A) \in \boldsymbol{D}$.
  - For any arrow $f: A \to B \in \boldsymbol{C}$, $F(f): F(A) \to F(B) \in \boldsymbol{D}$.



  - For any object $A \in \boldsymbol{C}$, $F(1_A) = 1_{F(A)}$.
  - For any arrows $f: A \to B$ and $g: B \to C \in \boldsymbol{C}$, $F(g \circ f) = F(g) \circ F(f)$.





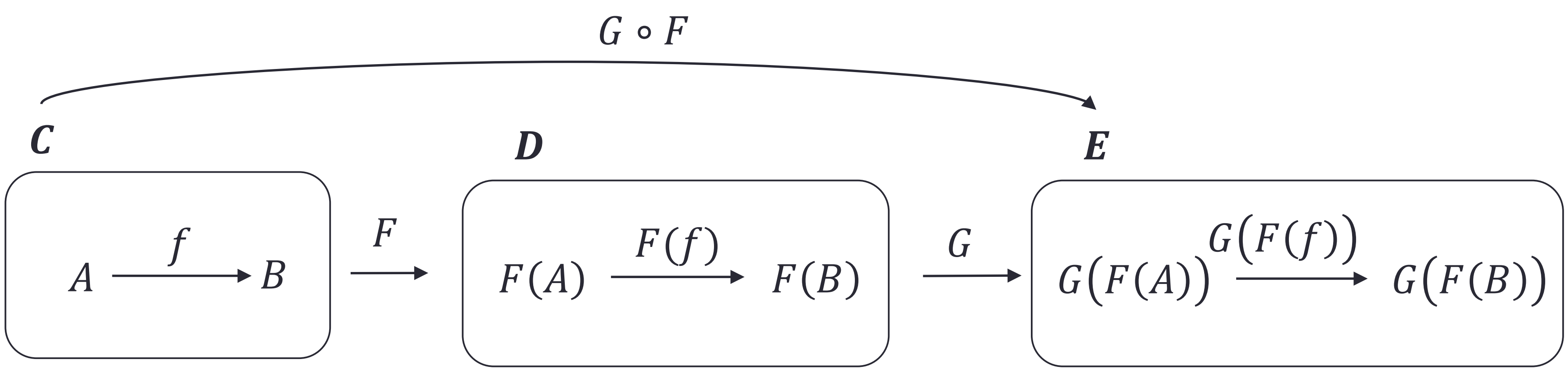
Examples of Functors

- forgetful functor
  - forget the structure
  - $G: \boldsymbol{Grp} \to \boldsymbol{Set}$
    - $G\left(\left(S, \cdot, e, {}^{-1}\right)\right) = S$
    - $G(f) = f$
- free functor
  - give the structure
  - $F: \boldsymbol{Set} \to \boldsymbol{Grp}$
    - $F(S) =$ the free group generated from S
    - $F(f) =$ extend f to homomorphism
- functor between monoids
  - $F: (M, \cdot, e) \to (N, \cdot, e)$
  - $F(e) = e$
  - $F(x \cdot y) = F(x) \cdot F(y)$
  - homomorphism between monoids
- functor between partially ordered sets
  - $F: (D, \sqsubseteq) \to (E, \sqsubseteq)$
  - If $x \sqsubseteq y$, then $F(x) \sqsubseteq F(y)$
  - monotonic function

Composition of Functors

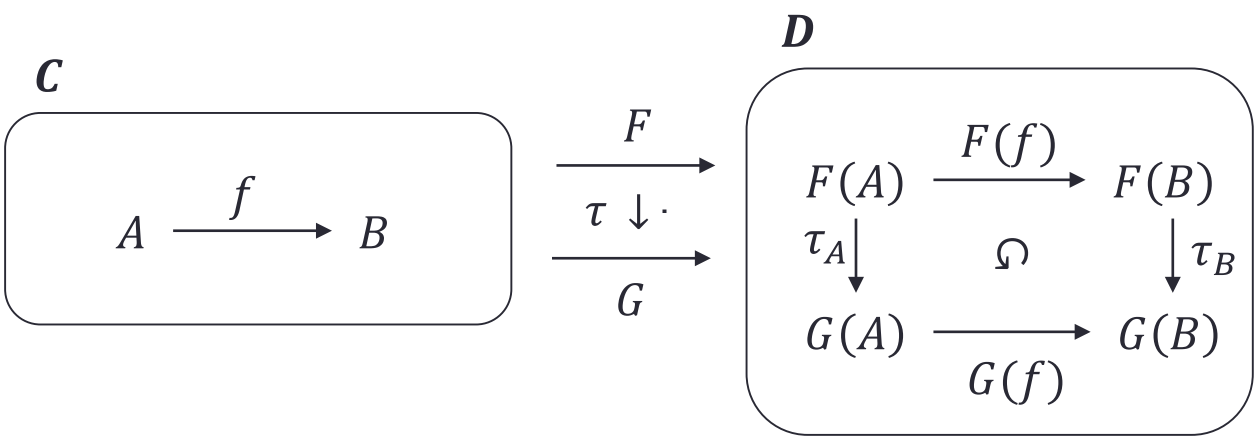
- Composition of functors
  - A functor from $\boldsymbol{C}$ to $\boldsymbol{D}$, $F: \boldsymbol{C} \rightarrow \boldsymbol{D}$
  - A functor from $\boldsymbol{D}$ to $\boldsymbol{E}$, $G: \boldsymbol{D} \rightarrow \boldsymbol{E}$

- $G \circ F: \boldsymbol{C} \rightarrow \boldsymbol{E}$
  - $G \circ F(A) = G\big(F(A)\big)$
  - $G \circ F(f) = G\big(F(f)\big)$

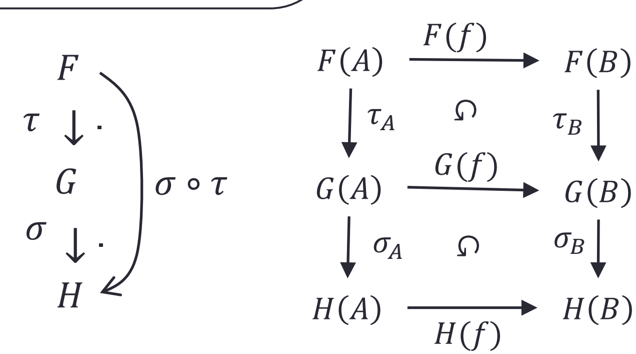


Natural Transformation

- For functors F and G from category $\boldsymbol{C}$ to $\boldsymbol{D}$, $\tau: F \dot{\rightarrow} G$ is a natural transformation if the followings are met:
  - For any object $A \in \boldsymbol{C}$, there exists an arrow $\tau_A: F(A) \rightarrow G(A) \in \boldsymbol{D}$.
  - For any arrow $f: A \rightarrow B \in C$, $\tau_B \circ F(f) = G(f) \circ \tau_A$ holds.

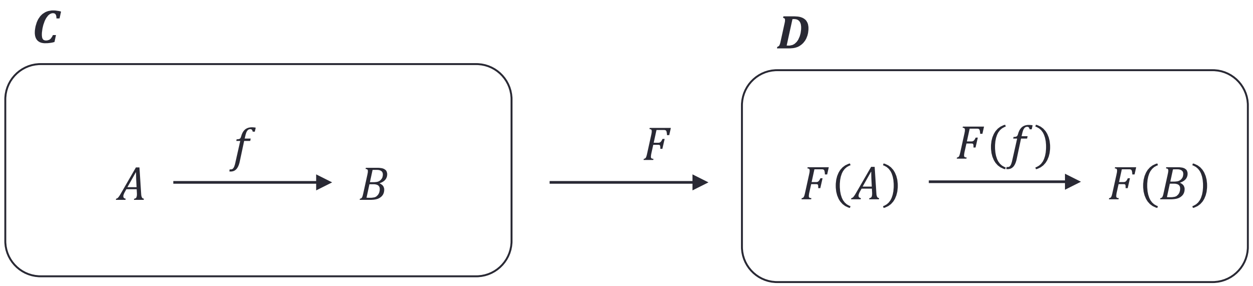


- Category $\boldsymbol{D}^{\boldsymbol{C}}$
  - object: functor from $\boldsymbol{C}$ to $\boldsymbol{D}$
  - arrow: natural transformation



Covariant and Contravariant Functors

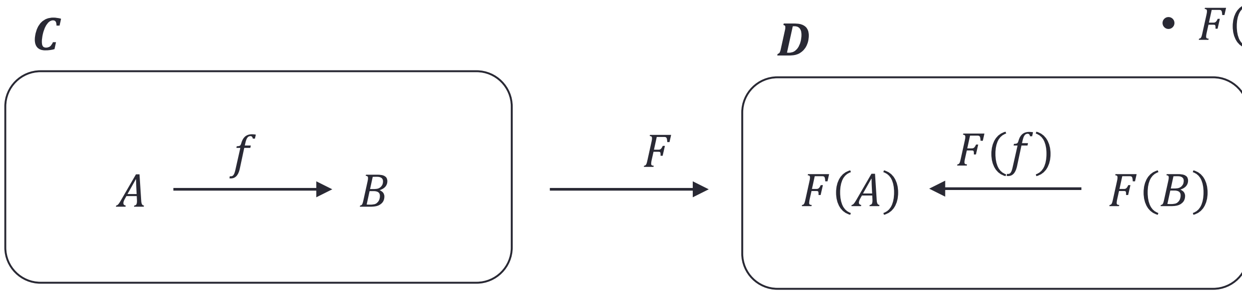
- covariant functor
  - For any object $A \in \boldsymbol{C}$, $F(A) \in \boldsymbol{D}$
  - For any arrow $f\colon A \to B \in \boldsymbol{C}$, $F(f)\colon F(A) \to F(B) \in \boldsymbol{D}$



- $F(1_A) = 1_{F(A)}$
- $F(g \circ f) = F(g) \circ F(f)$

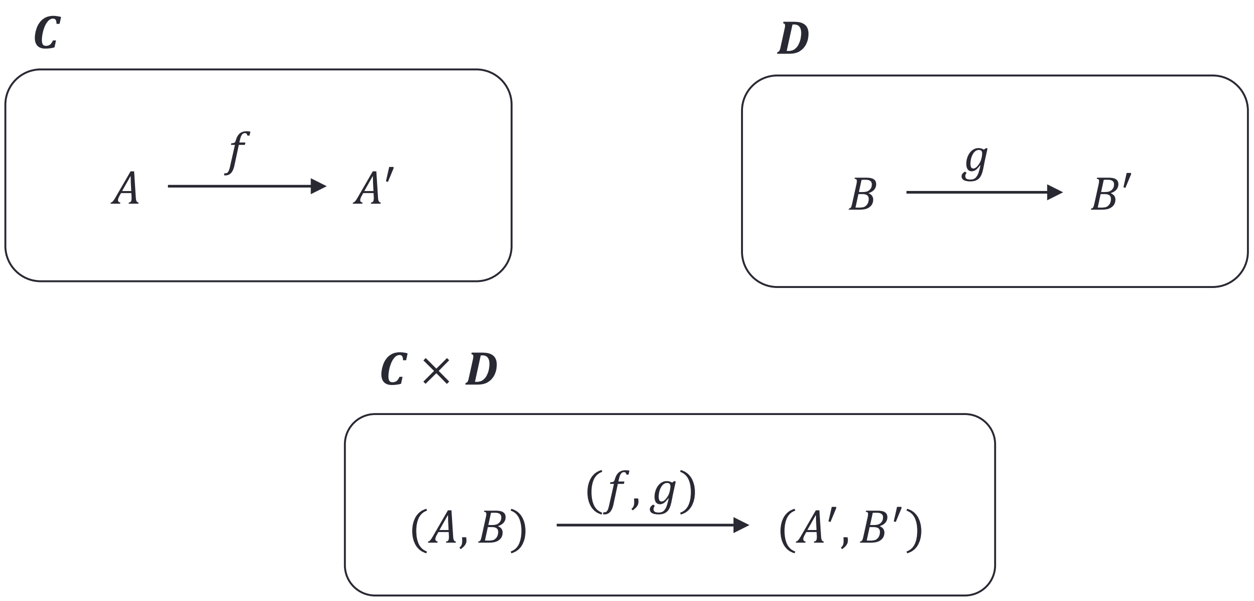
- contravariant functor
  - For any object $A \in \boldsymbol{C}$, $F(A) \in \boldsymbol{D}$
  - For any arrow $f\colon A \to B \in \boldsymbol{C}$, $F(f)\colon F(B) \to F(A) \in \boldsymbol{D}$

- $F(1_A) = 1_{F(A)}$
- $F(g \circ f) = F(f) \circ F(g)$



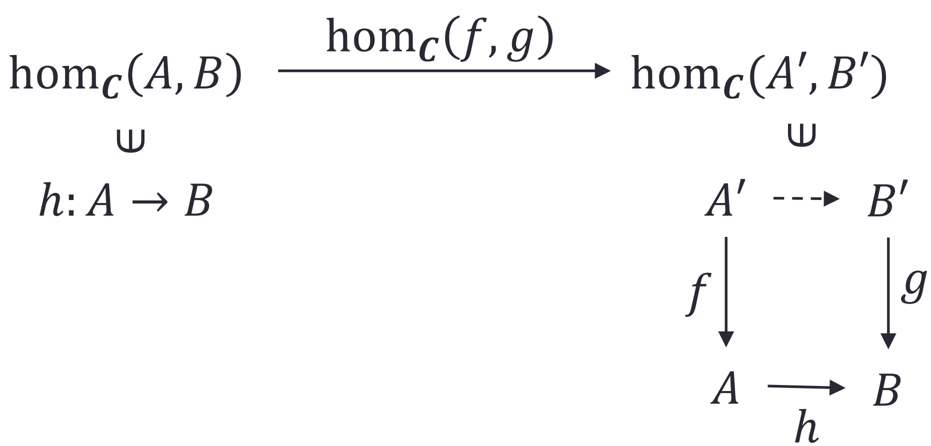
Product of Categories

- For category $\boldsymbol{C}$ and $\boldsymbol{D}$, the product category $\boldsymbol{C} \times \boldsymbol{D}$ is
  - object: for any objects $A \in \boldsymbol{C}$ and $B \in \boldsymbol{D}$, $(A, B) \in \boldsymbol{C} \times \boldsymbol{D}$
  - arrow: for any arrow $f: A \to A' \in \boldsymbol{C}$ and $g: B \to B' \in \boldsymbol{D}$, $(f, g): (A, B) \to (A', B') \in \boldsymbol{C} \times \boldsymbol{D}$



Hom Functor

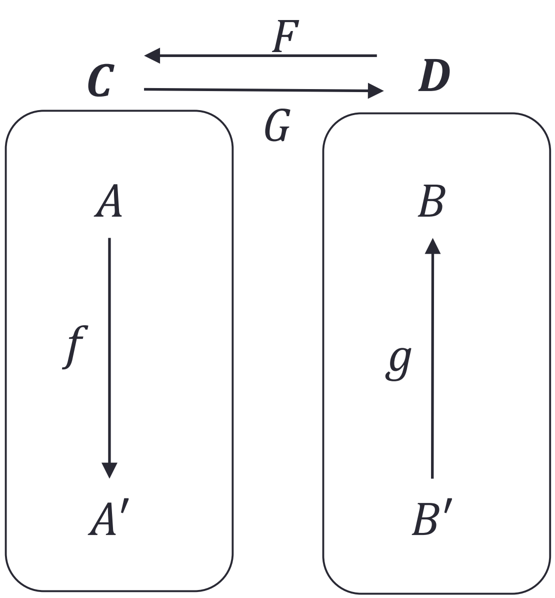
- For a category $\boldsymbol{C}$, its hom functor is the functor which maps arrows to sets.
- $\hom_{\boldsymbol{C}}(-,-): \boldsymbol{C}^{op} \times \boldsymbol{C} \to \boldsymbol{Set}$
  - $\hom_{\boldsymbol{C}}(A,B)$: the set of arrows from A to B
  - For $f: A' \to A$ and $g: B \to B'$, $\hom_{\boldsymbol{C}}(f,g): \hom_{\boldsymbol{C}}(A,B) \to \hom_{\boldsymbol{C}}(A',B')$ is defined as $\hom_{\boldsymbol{C}}(f,g)(h) = g \circ h \circ f$.

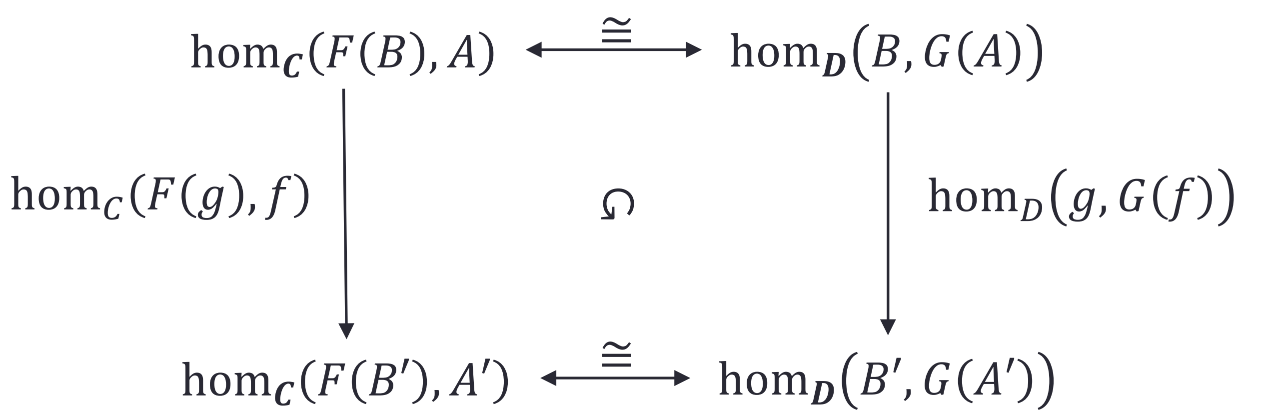


Adjunction

- An adjunction from category $\boldsymbol{C}$ to $\boldsymbol{D}$ is:
  - left adjoint functor $F: \boldsymbol{D} \rightarrow \boldsymbol{C}$
  - right adjoint functor $G: \boldsymbol{C} \rightarrow \boldsymbol{D}$
  - For $A \in \boldsymbol{C}$ and $B \in \boldsymbol{D}$, $\hom_{\boldsymbol{C}}(F(B), A) \cong \hom_{\boldsymbol{D}}(B, G(A))$ is naturally isomorphic.
  - Adjunction $F \dashv G$

$$\hom_{\boldsymbol{C}}(F(-), -) \simeq \hom_{\boldsymbol{D}}(-, G(-)) : \boldsymbol{D}^{op} \times \boldsymbol{C} \rightarrow \boldsymbol{Set}$$

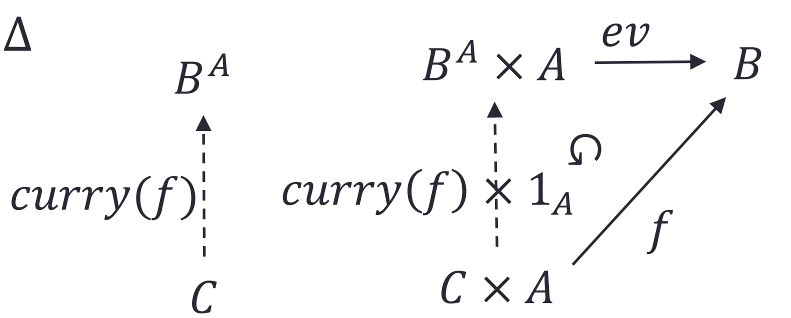




Everything is Adjunction

- Let $\Delta: \boldsymbol{C} \to \boldsymbol{C} \times \boldsymbol{C}$ be the diagonal functor $\Delta(A) = (A, A)$
  - $+ \dashv \Delta \dashv \times$
  - Co-product is the left adjoint functor of Δ
  - Product is the right adjoint functor of Δ
- Let $! : \boldsymbol{C} \to \cdot$ be the functor $!\,(A) = \cdot$

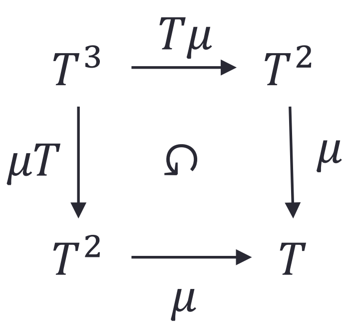
  where $\cdot$ is the category which has only one object and one identity arrow.
  - Initial object is the left adjoint functor of !
  - Final object is the right adjoint of !
- Let $\boldsymbol{D}$ be a subcategory of $\boldsymbol{C}$, and $\Delta: \boldsymbol{C} \to \boldsymbol{C}^{\boldsymbol{D}}$ be $\Delta(A)(B) = (A)$, $\Delta(A)(f) = 1_A$
  - Co-limit $\varinjlim \boldsymbol{D}$ is the left adjoint functor of Δ
  - Limit $\varprojlim \boldsymbol{D}$ is the right adjoint functor of Δ
- The right adjoint functor of $(-) \times A: \boldsymbol{C} \to \boldsymbol{C}$ is the function space $(-)^A: \boldsymbol{C} \to \boldsymbol{C}$
  - $\hom_{\boldsymbol{C}}(C \times A, B) \simeq \hom_{\boldsymbol{C}}(C, B^A)$

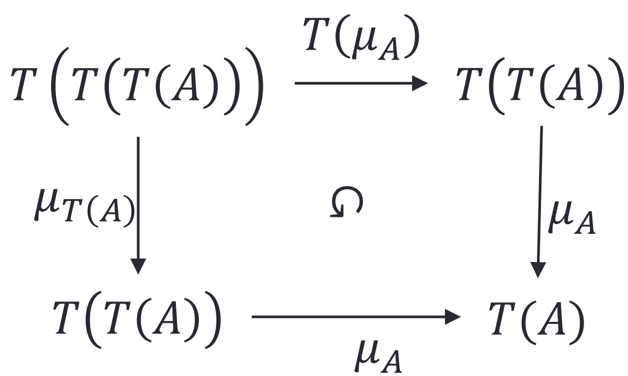


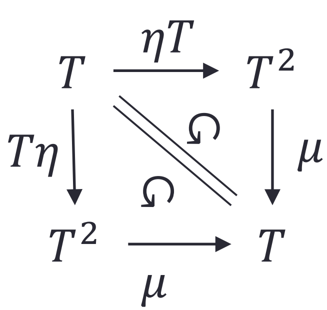
Monad

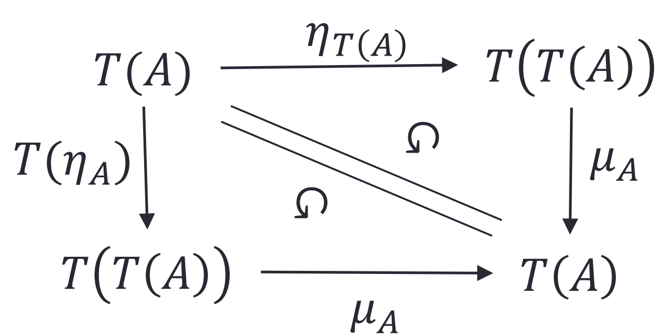
- For a category $\boldsymbol{C}$, a monad consists of a functor $T: \boldsymbol{C} \to \boldsymbol{C}$ and two natural transformations $\eta: 1_{\boldsymbol{C}} \dot{\to} T$ and $\mu: T^2 \dot{\to} T$ which satisfies the followings:
  - $\mu \circ T\mu = \mu \circ \mu T$
  - $\mu \circ T\eta = \mu \circ \eta T = 1_T$

associative law for monoid









unit law for monoid

- When $F \dashv G$, $T = G \circ F$ is a monad.

Haskell Monad

```
class Monad m where
   (>>=)   :: m a -> (a -> m b) -> m b
   return :: a -> m a
```

- Instances of Monad class are monad.
  - Need to implement two functions.
  - (>>=) is called bind

- The two functions need to satisfy the following equations.
  - Monad laws

```
1. (return x) >>= f    = f x
2. m >>= return        = m
3. (m >>= f) >>= g     = m >>= (\x -> f x >>= g)
```

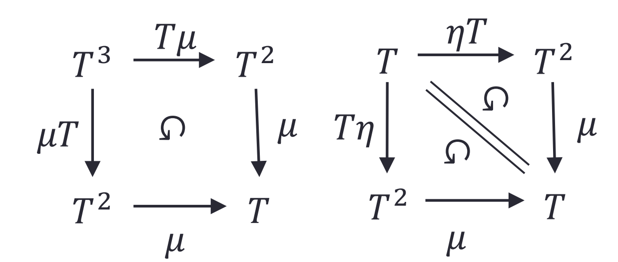
Haskell Monad and Category Monad

```
class Monad m where
   (>>=)   :: m a -> (a -> m b) -> m b
   return :: a -> m a
```

```
1. (return x) >>= f    = f x
2. m >>= return        = m
3. (m >>= f) >>= g     = m >>= (\x -> f x >>= g)
```

For a category $\boldsymbol{C}$, a monad consists of a functor $T{:}\,\boldsymbol{C}\to\boldsymbol{C}$ and two natural transformations $\eta{:}\,1_{\boldsymbol{C}}\dot{\to}T$ and $\mu{:}\,T^2\dot{\to}T$ which satisfies the followings:

- $\mu\circ T\mu=\mu\circ\mu T$
- $\mu\circ T\eta=\mu\circ\eta T=1_T$

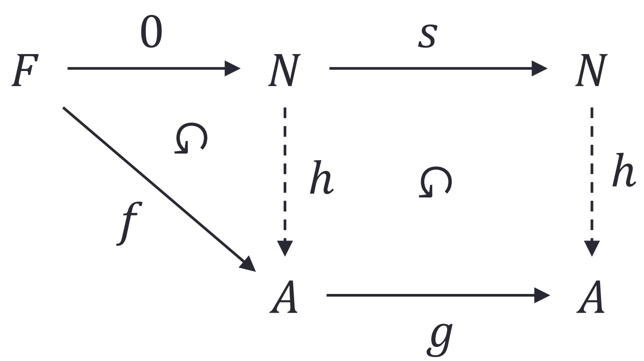


- `return::a -> m a` is
  - $\eta_A{:}\,A\to T(A)$
- `(>>=)::m a -> (a -> m b) -> m b` is
  - Given $f{:}\,A\to T(B)$, $T(f)$ is $T(A)\to T\big(T(B)\big)$, so combine this with $\mu_B{:}\,T\big(T(B)\big)\to T(B)$ gives $\mu_B\circ T(f){:}\,T(A)\to T(B)$.
  - $\big(A\to T(B)\big)\to\big(T(A)\to T(B)\big)\equiv T(A)\to\big(A\to T(B)\big)\to T(B)$

Natural Number Object

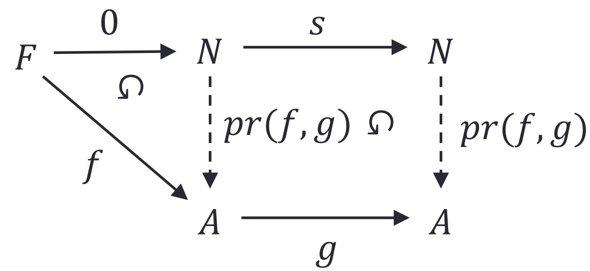
- N is a Natural Number Object when
  - There are two arrows.
    - $0: F \to N$
    - $s: N \to N$

    where F is the final object.
  - For $f: F \to A$ and $g: A \to A$, there exists a unique arrow $h: N \to A$ such that
    - $h \circ 0 = f$
    - $h \circ s = g \circ h$

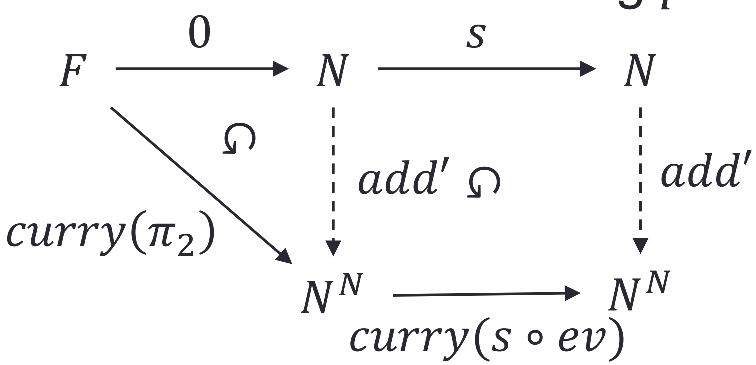


- Let us write $pr(f, g)$ for h.

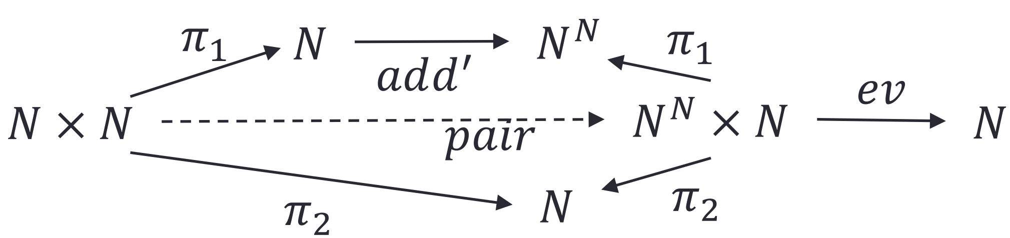
Addition Arrow

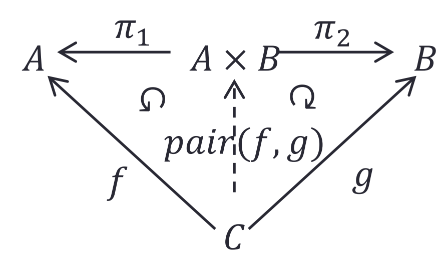


- Definition of $add: N \times N \to N$
  - Define curried $add': N \to N^N$ using pr.



  - $\pi_2: F \times N \to N$ is curried to $curry(\pi_2): F \to N^N$
  - $s \circ ev: N^N \times N \to N$ is curried to $curry(s \circ ev): N^N \to N^N$
  - $add' = pr\big(curry(\pi_2), curry(s \circ ev)\big)$

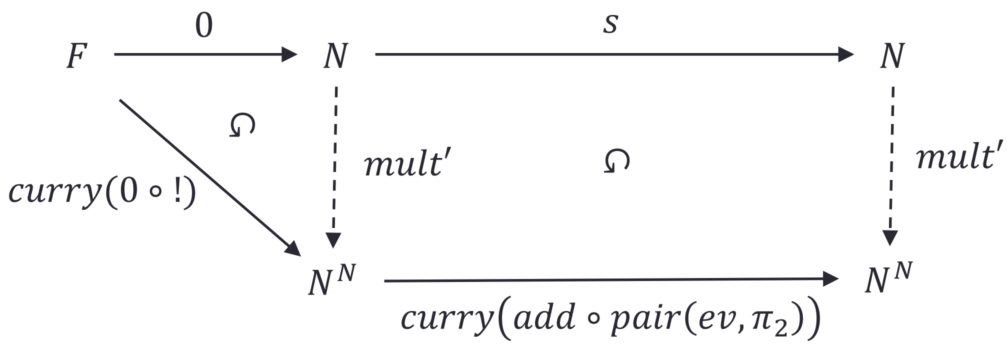




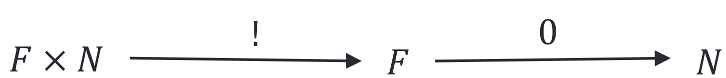
  - $add = ev \circ pair\big(pr\big(curry(\pi_2), curry(s \circ ev)\big) \circ \pi_1, \pi_2\big)$

Multiplication Arrow

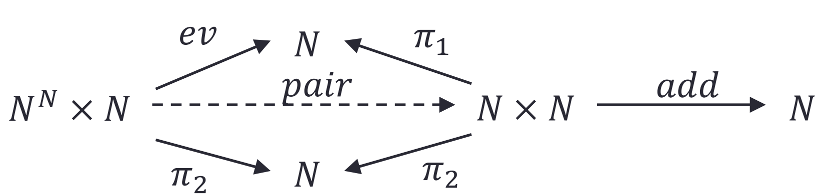
- Definition of $mult: N \times N \to N$
  - Define curried $mult': N \to N^N$ using pr and add.



  - $0 \circ !: F \times N \to N$ curried to $curry(0 \circ !): F \to N^N$



  - $add \circ pair(ev, \pi_2): N^N \times N \to N$ curried to $curry\big(add \circ pair(ev, \pi_2)\big): N^N \to N^N$



  - $mult' = pr\left(curry(0 \circ !), curry\big(add \circ pair(ev, \pi_2)\big)\right)$
  - $mult = ev \circ pair\left(pr\left(curry(0 \circ !), curry\big(add \circ pair(ev, \pi_2)\big)\right) \circ \pi_1, \pi_2\right)$

Summary

- Functor
- Adjunction
  - Right adjoint functor and left adjoint functor
  - Diagonal functor and product and co-product
  - Limit and Adjoint
  - Monad
- Natural Number Object