

Natural transformations

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$,

$\mathbf{K}:$	$\mathbf{K}':$	
A	$\mathbf{F}(A)$	$\mathbf{G}(A)$
B	$\mathbf{F}(B)$	$\mathbf{G}(B)$

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$, a *natural transformation* from \mathbf{F} to \mathbf{G}

$$\tau: \mathbf{F} \rightarrow \mathbf{G}$$

\mathbf{K} :

A

B

\mathbf{K}' :

$\mathbf{F}(A)$

$\mathbf{F}(B)$

$\mathbf{G}(A)$

$\mathbf{G}(B)$

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$, a *natural transformation* from \mathbf{F} to \mathbf{G}

$$\tau: \mathbf{F} \rightarrow \mathbf{G}$$

is a family $\tau = \langle \tau_A: \mathbf{F}(A) \rightarrow \mathbf{G}(A) \rangle_{A \in |\mathbf{K}|}$ of \mathbf{K}' -morphisms

$$\begin{array}{ccc} \mathbf{K}: & & \mathbf{K}': \\ & A & \mathbf{F}(A) \xrightarrow{\tau_A} \mathbf{G}(A) \\ & & \\ & B & \mathbf{F}(B) \xrightarrow{\tau_B} \mathbf{G}(B) \end{array}$$

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$, a *natural transformation* from \mathbf{F} to \mathbf{G}

$$\tau: \mathbf{F} \rightarrow \mathbf{G}$$

is a family $\tau = \langle \tau_A: \mathbf{F}(A) \rightarrow \mathbf{G}(A) \rangle_{A \in |\mathbf{K}|}$ of \mathbf{K}' -morphisms such that for all $f: A \rightarrow B$ in \mathbf{K} (with $A, B \in |\mathbf{K}|$),

$$\begin{array}{ccc} \mathbf{K}: & & \mathbf{K}': \\ A & & \mathbf{F}(A) \xrightarrow{\tau_A} \mathbf{G}(A) \\ \downarrow f & & \\ B & & \mathbf{F}(B) \xrightarrow{\tau_B} \mathbf{G}(B) \end{array}$$

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$, a *natural transformation* from \mathbf{F} to \mathbf{G}

$$\tau: \mathbf{F} \rightarrow \mathbf{G}$$

is a family $\tau = \langle \tau_A: \mathbf{F}(A) \rightarrow \mathbf{G}(A) \rangle_{A \in |\mathbf{K}|}$ of \mathbf{K}' -morphisms such that for all $f: A \rightarrow B$ in \mathbf{K} (with $A, B \in |\mathbf{K}|$), $\tau_A; \mathbf{G}(f) = \mathbf{F}(f); \tau_B$

$$\begin{array}{ccc} \mathbf{K}: & & \mathbf{K}': \\ A & & \mathbf{F}(A) \xrightarrow{\tau_A} \mathbf{G}(A) \\ \downarrow f & & \downarrow \mathbf{F}(f) \quad \downarrow \mathbf{G}(f) \\ B & & \mathbf{F}(B) \xrightarrow{\tau_B} \mathbf{G}(B) \end{array}$$

Natural transformations

Given two parallel functors $\mathbf{F}, \mathbf{G}: \mathbf{K} \rightarrow \mathbf{K}'$, a *natural transformation* from \mathbf{F} to \mathbf{G}

$$\tau: \mathbf{F} \rightarrow \mathbf{G}$$

is a family $\tau = \langle \tau_A: \mathbf{F}(A) \rightarrow \mathbf{G}(A) \rangle_{A \in |\mathbf{K}|}$ of \mathbf{K}' -morphisms such that for all $f: A \rightarrow B$ in \mathbf{K} (with $A, B \in |\mathbf{K}|$), $\tau_A; \mathbf{G}(f) = \mathbf{F}(f); \tau_B$

Then, τ is a *natural isomorphism* if for all $A \in |\mathbf{K}|$, τ_A is an isomorphism.

$$\begin{array}{ccc}
 \mathbf{K}: & & \mathbf{K}': \\
 A & & \mathbf{F}(A) \xrightarrow{\tau_A} \mathbf{G}(A) \\
 \downarrow f & & \downarrow \mathbf{F}(f) \quad \downarrow \mathbf{G}(f) \\
 B & & \mathbf{F}(B) \xrightarrow{\tau_B} \mathbf{G}(B)
 \end{array}$$

Examples

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.

$$X \quad \mathbf{Id}_{\mathbf{Set}}(X) \xrightarrow{sing_X} \mathbf{P}(X)$$

$$Y \quad \mathbf{Id}_{\mathbf{Set}}(Y) \xrightarrow{sing_Y} \mathbf{P}(Y)$$

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.

For all $f: X \rightarrow Y$,

$$sing_X; \mathbf{P}(f) = \mathbf{Id}_{\mathbf{Set}}(f); sing_Y,$$

$$\begin{array}{ccccc}
 X & \mathbf{Id}_{\mathbf{Set}}(X) & \xrightarrow{sing_X} & \mathbf{P}(X) & \\
 \downarrow f & \downarrow \mathbf{Id}_{\mathbf{Set}}(f) & & \downarrow \mathbf{P}(f) & \\
 Y & \mathbf{Id}_{\mathbf{Set}}(Y) & \xrightarrow{sing_Y} & \mathbf{P}(Y) &
 \end{array}$$

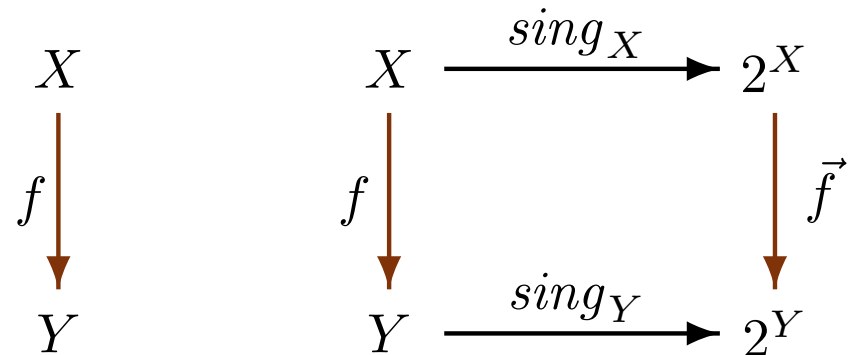
Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.

For all $f: X \rightarrow Y$,

$$sing_X; \mathbf{P}(f) = \mathbf{Id}_{\mathbf{Set}}(f); sing_Y,$$

i.e. $sing_X; \vec{f} = f; sing_Y$,



Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.

For all $f: X \rightarrow Y$,

$$sing_X; \mathbf{P}(f) = \mathbf{Id}_{\mathbf{Set}}(f); sing_Y,$$

$$\text{i.e. } sing_X; \vec{f} = f; sing_Y,$$

$$\text{i.e. for } x \in X, \vec{f}(\{x\}) = \{f(x)\}.$$

$$\begin{array}{ccc}
 X & \xrightarrow{sing_X} & 2^X \\
 f \downarrow & & \downarrow \vec{f} \\
 Y & \xrightarrow{sing_Y} & 2^Y
 \end{array}$$

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.
- *singleton-list functions*: $sing^{\mathbf{List}}: \mathbf{Id}_{\mathbf{Set}} \rightarrow |\mathbf{List}| (: \mathbf{Set} \rightarrow \mathbf{Set})$, where $|\mathbf{List}| = \mathbf{List}; |-|: \mathbf{Set}(\rightarrow \mathbf{Monoid}) \rightarrow \mathbf{Set}$, and for all $X \in |\mathbf{Set}|$, $sing_X^{\mathbf{List}}: X \rightarrow X^*$ is a function defined by $sing_X^{\mathbf{List}}(x) = \langle x \rangle$ for $x \in X$

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.
- *singleton-list functions*: $sing^{\mathbf{List}}: \mathbf{Id}_{\mathbf{Set}} \rightarrow |\mathbf{List}| (: \mathbf{Set} \rightarrow \mathbf{Set})$, where $|\mathbf{List}| = \mathbf{List}; |-|: \mathbf{Set}(\rightarrow \mathbf{Monoid}) \rightarrow \mathbf{Set}$, and for all $X \in |\mathbf{Set}|$, $sing_X^{\mathbf{List}}: X \rightarrow X^*$ is a function defined by $sing_X^{\mathbf{List}}(x) = \langle x \rangle$ for $x \in X$
- *append functions*: $append: |\mathbf{List}|; \mathbf{CP} \rightarrow |\mathbf{List}| (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $append_X: (X^* \times X^*) \rightarrow X^*$ is the usual append function (list concatenation) polymorphic functions between algebraic types

Examples

- *identity transformations*: $id_{\mathbf{F}}: \mathbf{F} \rightarrow \mathbf{F}$, where $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$, for all objects $A \in |\mathbf{K}|$, $(id_{\mathbf{F}})_A = id_{\mathbf{F}(A)}: \mathbf{F}(A) \rightarrow \mathbf{F}(A)$
- *singleton functions*: $sing: \mathbf{Id}_{\mathbf{Set}} \rightarrow \mathbf{P} (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $sing_X: X \rightarrow \mathbf{P}(X)$ is a function defined by $sing_X(x) = \{x\}$ for $x \in X$.
- *singleton-list functions*: $sing^{\mathbf{List}}: \mathbf{Id}_{\mathbf{Set}} \rightarrow |\mathbf{List}| (: \mathbf{Set} \rightarrow \mathbf{Set})$, where $|\mathbf{List}| = \mathbf{List}; |-|: \mathbf{Set}(\rightarrow \mathbf{Monoid}) \rightarrow \mathbf{Set}$, and for all $X \in |\mathbf{Set}|$, $sing_X^{\mathbf{List}}: X \rightarrow X^*$ is a function defined by $sing_X^{\mathbf{List}}(x) = \langle x \rangle$ for $x \in X$
- *append functions*: $append: |\mathbf{List}|; \mathbf{CP} \rightarrow |\mathbf{List}| (: \mathbf{Set} \rightarrow \mathbf{Set})$, where for all $X \in |\mathbf{Set}|$, $append_X: (X^* \times X^*) \rightarrow X^*$ is the usual append function (list concatenation) polymorphic functions between algebraic types

$$\begin{array}{ccccc}
 X & & X^* \times X^* & \xrightarrow{append_X} & X^* \\
 \downarrow f & & \downarrow f^* \times f^* & & \downarrow f^* \\
 Y & & Y^* \times Y^* & \xrightarrow{append_Y} & Y^*
 \end{array}$$

Polymorphic functions

Polymorphic functions

Work out the following generalisation of the last two examples:

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$
 - $\llbracket \alpha_i \rrbracket (X_1, \dots, X_n) = X_i$
 - $\llbracket \mathbf{int} \rrbracket (X_1, \dots, X_n) = \{\dots, -2, -1, 0, 1, 2, \dots\}$
 - $\llbracket T_1 \times T_2 \rrbracket (X_1, \dots, X_n) = \llbracket T_1 \rrbracket (X_1, \dots, X_n) \times \llbracket T_2 \rrbracket (X_1, \dots, X_n)$
 - $\llbracket T_1 + T_2 \rrbracket (X_1, \dots, X_n) = \llbracket T_1 \rrbracket (X_1, \dots, X_n) + \llbracket T_2 \rrbracket (X_1, \dots, X_n)$
 - \dots
 - \dots recursive type definitions work as well \dots

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$
- argue that in a representative subset of **Standard ML**, for each polymorphic expression $E : \forall \alpha_1 \dots \alpha_n \cdot T \rightarrow T'$ its semantics is a natural transformation $\llbracket E \rrbracket : \llbracket T \rrbracket \rightarrow \llbracket T' \rrbracket$

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$
- argue that in a representative subset of **Standard ML**, for each polymorphic expression $E : \forall \alpha_1 \dots \alpha_n \cdot T \rightarrow T'$ its semantics is a natural transformation $\llbracket E \rrbracket : \llbracket T \rrbracket \rightarrow \llbracket T' \rrbracket$
 - by induction on the structure of well-typed expressions

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket: \mathbf{Set}^n \rightarrow \mathbf{Set}$
- argue that in a representative subset of **Standard ML**, for each polymorphic expression $E: \forall \alpha_1 \dots \alpha_n \cdot T \rightarrow T'$ its semantics is a natural transformation $\llbracket E \rrbracket: \llbracket T \rrbracket \rightarrow \llbracket T' \rrbracket$
- Then for $f_1: X_1 \rightarrow Y_1, \dots, f_n: X_n \rightarrow Y_n$:

$$\llbracket T \rrbracket(f_1, \dots, f_n); \llbracket E \rrbracket_{\langle Y_1, \dots, Y_n \rangle} = \llbracket E \rrbracket_{\langle X_1, \dots, X_n \rangle}; \llbracket T' \rrbracket(f_1, \dots, f_n)$$

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket: \mathbf{Set}^n \rightarrow \mathbf{Set}$
- argue that in a representative subset of **Standard ML**, for each polymorphic expression $E: \forall \alpha_1 \dots \alpha_n \cdot T \rightarrow T'$ its semantics is a natural transformation $\llbracket E \rrbracket: \llbracket T \rrbracket \rightarrow \llbracket T' \rrbracket$
- Then for $f_1: X_1 \rightarrow Y_1, \dots, f_n: X_n \rightarrow Y_n$:

$$\llbracket T \rrbracket(f_1, \dots, f_n); \llbracket E \rrbracket_{\langle Y_1, \dots, Y_n \rangle} = \llbracket E \rrbracket_{\langle X_1, \dots, X_n \rangle}; \llbracket T' \rrbracket(f_1, \dots, f_n)$$

For instance, for $rev: \alpha \text{ list} \rightarrow \alpha \text{ list}$,

$even: \text{int} \rightarrow \text{bool}$ and $l: \text{int list}$:

$$rev(even^*(l)) = even^*(rev(l))$$

Polymorphic functions

Work out the following generalisation of the last two examples:

- for each algebraic type scheme $\forall \alpha_1 \dots \alpha_n \cdot T$, built in **Standard ML** using at least products and algebraic data types (no function types though), define the corresponding functor $\llbracket T \rrbracket: \mathbf{Set}^n \rightarrow \mathbf{Set}$
- argue that in a representative subset of **Standard ML**, for each polymorphic expression $E: \forall \alpha_1 \dots \alpha_n \cdot T \rightarrow T'$ its semantics is a natural transformation $\llbracket E \rrbracket: \llbracket T \rrbracket \rightarrow \llbracket T' \rrbracket$
- Then for $f_1: X_1 \rightarrow Y_1, \dots, f_n: X_n \rightarrow Y_n$:

$$\llbracket T \rrbracket(f_1, \dots, f_n); \llbracket E \rrbracket_{\langle Y_1, \dots, Y_n \rangle} = \llbracket E \rrbracket_{\langle X_1, \dots, X_n \rangle}; \llbracket T' \rrbracket(f_1, \dots, f_n)$$

For instance, for $rev: \alpha \text{ list} \rightarrow \alpha \text{ list}$,

$even: \text{int} \rightarrow \text{bool}$ and $l: \text{int list}$:

$$rev(even^*(l)) = even^*(rev(l))$$

Theorems for free!
(see Wadler 89)

Yoneda lemma

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(A, -), \mathbf{F}) \cong \mathbf{F}(A)$$

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(A, -), \mathbf{F}) \cong \mathbf{F}(A)$$

natural transformations from $\mathbf{Hom}_{\mathbf{K}}(A, -)$ to \mathbf{F} , between functors from \mathbf{K} to \mathbf{Set} , are given exactly by the elements of the set $\mathbf{F}(A)$

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(A, -), \mathbf{F}) \cong \mathbf{F}(A)$$

natural transformations from $\mathbf{Hom}_{\mathbf{K}}(A, -)$ to \mathbf{F} , between functors from \mathbf{K} to \mathbf{Set} , are given exactly by the elements of the set $\mathbf{F}(A)$

EXERCISES:

- Dualise: for $\mathbf{G}: \mathbf{K}^{op} \rightarrow \mathbf{Set}$,

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(A, -), \mathbf{F}) \cong \mathbf{F}(A)$$

natural transformations from $\mathbf{Hom}_{\mathbf{K}}(A, -)$ to \mathbf{F} , between functors from \mathbf{K} to \mathbf{Set} , are given exactly by the elements of the set $\mathbf{F}(A)$

EXERCISES:

- Dualise: for $\mathbf{G}: \mathbf{K}^{op} \rightarrow \mathbf{Set}$,

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(-, A), \mathbf{G}) \cong \mathbf{G}(A)$$

Yoneda lemma

Given a locally small category \mathbf{K} , functor $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{Set}$ and object $A \in |\mathbf{K}|$:

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(A, -), \mathbf{F}) \cong \mathbf{F}(A)$$

natural transformations from $\mathbf{Hom}_{\mathbf{K}}(A, -)$ to \mathbf{F} , between functors from \mathbf{K} to \mathbf{Set} , are given exactly by the elements of the set $\mathbf{F}(A)$

EXERCISES:

- Dualise: for $\mathbf{G}: \mathbf{K}^{op} \rightarrow \mathbf{Set}$,

$$\text{Nat}(\mathbf{Hom}_{\mathbf{K}}(-, A), \mathbf{G}) \cong \mathbf{G}(A)$$

- Characterise all natural transformations from $\mathbf{Hom}_{\mathbf{K}}(A, -)$ to $\mathbf{Hom}_{\mathbf{K}}(B, -)$, for all objects $A, B \in |\mathbf{K}|$.

Proof

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$,

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} .
Note: $\mathbf{F}(f): \mathbf{F}(A) \rightarrow \mathbf{F}(B)$ in \mathbf{Set} , so $\mathbf{F}(f)(a) \in \mathbf{F}(B)$.

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$

$$\begin{array}{ccc}
 \mathbf{K}: & & \mathbf{Set}: \\
 B & & \mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B) \\
 \downarrow g & & \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g) \\
 C & & \mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)
 \end{array}$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{array}{ccc}
 \mathbf{K}: & & \mathbf{Set}: \\
 B & & \mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B) \\
 \downarrow g & & \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g) \\
 C & & \mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)
 \end{array}$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$, $\mathbf{F}(g)(\tau_B^a(f))$

$$\begin{array}{ccc}
 \mathbf{K}: & & \mathbf{Set}: \\
 B & & \mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B) \\
 \downarrow g & & \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g) \\
 C & & \mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)
 \end{array}$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$, $\mathbf{F}(g)(\tau_B^a(f)) = \mathbf{F}(g)(\mathbf{F}(f)(a))$

$$\begin{array}{ccc}
 \mathbf{K}: & & \mathbf{Set}: \\
 B & & \mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B) \\
 \downarrow g & & \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g) \\
 C & & \mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)
 \end{array}$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{aligned} \mathbf{F}(g)(\tau_B^a(f)) &= \mathbf{F}(g)(\mathbf{F}(f)(a)) \\ &= \mathbf{F}(f;g)(a) = \tau_C^a(f;g) \end{aligned}$$

K:	Set:
B $\downarrow g$ C	$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$ $\downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g)$ $\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{aligned} \mathbf{F}(g)(\tau_B^a(f)) &= \mathbf{F}(g)(\mathbf{F}(f)(a)) \\ &= \mathbf{F}(f;g)(a) = \tau_C^a(f;g) \\ &= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f)) \end{aligned}$$

$\mathbf{K}:$ B $\downarrow g$ C	$\mathbf{Set}:$ $\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$ $\downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g)$ $\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$
---	--

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\mathbf{F}(g)(\tau_B^a(f)) = \mathbf{F}(g)(\mathbf{F}(f)(a))$$

$$= \mathbf{F}(f;g)(a) = \tau_C^a(f;g)$$

$$= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f))$$

$$\text{Then } \tau_A^a(id_A) = a,$$

K:

B

$g \downarrow$

C

Set:

$$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$$

$$\begin{array}{ccc} \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \end{array}$$

$$\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\mathbf{F}(g)(\tau_B^a(f)) = \mathbf{F}(g)(\mathbf{F}(f)(a))$$

$$= \mathbf{F}(f;g)(a) = \tau_C^a(f;g)$$

$$= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f))$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

K:

B

$g \downarrow$

C

Set:

$$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$$

$$\begin{array}{ccc} \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \end{array}$$

$$\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\mathbf{F}(g)(\tau_B^a(f)) = \mathbf{F}(g)(\mathbf{F}(f)(a))$$

$$= \mathbf{F}(f;g)(a) = \tau_C^a(f;g)$$

$$= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f))$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

K:

B

$g \downarrow$

C

Set:

$$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$$

$$\begin{array}{ccc} \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \end{array}$$

$$\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\mathbf{F}(g)(\tau_B^a(f)) = \mathbf{F}(g)(\mathbf{F}(f)(a))$$

$$= \mathbf{F}(f;g)(a) = \tau_C^a(f;g)$$

$$= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f))$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

\mathbf{K} :

B

$g \downarrow$

C

Set:

$$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$$

$$\begin{array}{ccc} \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \\ \mathbf{K}(A, C) & \xrightarrow{\tau_C^a} & \mathbf{F}(C) \end{array}$$

- If $\tau: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$ is a natural transformation

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{aligned} \mathbf{F}(g)(\tau_B^a(f)) &= \mathbf{F}(g)(\mathbf{F}(f)(a)) \\ &= \mathbf{F}(f;g)(a) = \tau_C^a(f;g) \\ &= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f)) \end{aligned}$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

\mathbf{K} :

$$\begin{array}{c} B \\ \downarrow g \\ C \end{array}$$

Set:

$$\begin{array}{ccc} \mathbf{K}(A, B) & \xrightarrow{\tau_B^a} & \mathbf{F}(B) \\ \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \\ \mathbf{K}(A, C) & \xrightarrow{\tau_C^a} & \mathbf{F}(C) \end{array}$$

- If $\tau: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$ is a natural transformation then $\tau = \tau^a$, where we put $a = \tau_A(id_A)$,

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{aligned} \mathbf{F}(g)(\tau_B^a(f)) &= \mathbf{F}(g)(\mathbf{F}(f)(a)) \\ &= \mathbf{F}(f;g)(a) = \tau_C^a(f;g) \\ &= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f)) \end{aligned}$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

$\mathbf{K}:$	$\mathbf{Set}:$
B $\downarrow g$ C	$\mathbf{K}(A, B) \xrightarrow{\tau_B^a} \mathbf{F}(B)$ $\downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) \quad \downarrow \mathbf{F}(g)$ $\mathbf{K}(A, C) \xrightarrow{\tau_C^a} \mathbf{F}(C)$

- If $\tau: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$ is a natural transformation then $\tau = \tau^a$, where we put $a = \tau_A(id_A)$, since for $B \in |\mathbf{K}|$ and $f: A \rightarrow B$, $\tau_B(f) = \mathbf{F}(f)(\tau_A(id_A))$

Proof

- For $a \in \mathbf{F}(A)$, define $\tau^a: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$, as the family of functions $\tau_B^a: \mathbf{K}(A, B) \rightarrow \mathbf{F}(B)$, $B \in |\mathbf{K}|$, given by $\tau_B^a(f) = \mathbf{F}(f)(a)$ for $f: A \rightarrow B$ in \mathbf{K} . This is a natural transformation, since for $g: B \rightarrow C$ and then $f: A \rightarrow B$,

$$\begin{aligned} \mathbf{F}(g)(\tau_B^a(f)) &= \mathbf{F}(g)(\mathbf{F}(f)(a)) \\ &= \mathbf{F}(f;g)(a) = \tau_C^a(f;g) \\ &= \tau_C^a(\mathbf{Hom}_{\mathbf{K}}(A, g)(f)) \end{aligned}$$

Then $\tau_A^a(id_A) = a$, and so for distinct $a, a' \in \mathbf{F}(A)$, τ^a and $\tau^{a'}$ differ.

\mathbf{K} :

$$\begin{array}{c} B \\ \downarrow g \\ C \end{array}$$

\mathbf{Set} :

$$\begin{array}{ccc} \mathbf{K}(A, B) & \xrightarrow{\tau_B^a} & \mathbf{F}(B) \\ \downarrow (-);g = \mathbf{Hom}_{\mathbf{K}}(A, g) & & \downarrow \mathbf{F}(g) \\ \mathbf{K}(A, C) & \xrightarrow{\tau_C^a} & \mathbf{F}(C) \end{array}$$

- If $\tau: \mathbf{Hom}_{\mathbf{K}}(A, -) \rightarrow \mathbf{F}$ is a natural transformation then $\tau = \tau^a$, where we put $a = \tau_A(id_A)$, since for $B \in |\mathbf{K}|$ and $f: A \rightarrow B$, $\tau_B(f) = \mathbf{F}(f)(\tau_A(id_A))$ by naturality of τ :

$$\begin{array}{c} A \\ \downarrow f \\ B \end{array}$$

$$\begin{array}{ccc} \mathbf{K}(A, A) & \xrightarrow{\tau_A} & \mathbf{F}(A) \\ \downarrow (-);f = \mathbf{Hom}_{\mathbf{K}}(A, f) & & \downarrow \mathbf{F}(f) \\ \mathbf{K}(A, B) & \xrightarrow{\tau_B} & \mathbf{F}(B) \end{array}$$

Compositions

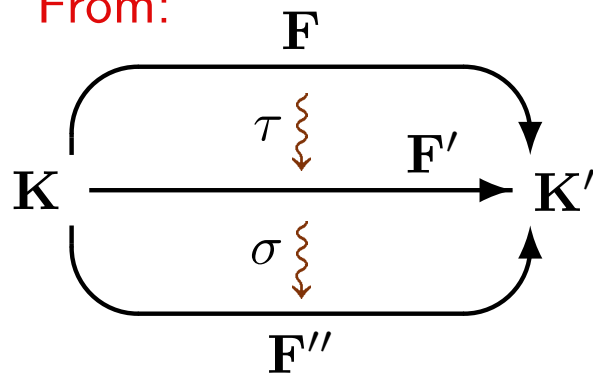
Compositions

vertical composition:

Compositions

vertical composition:

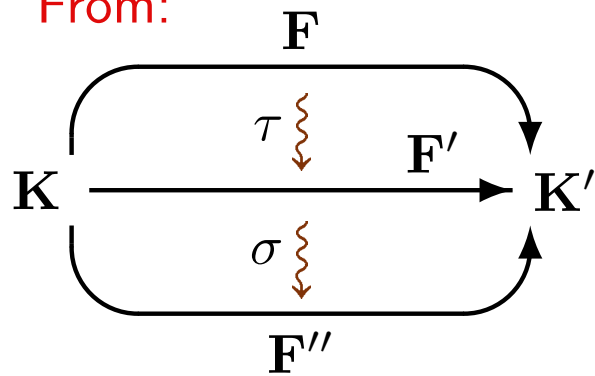
From:



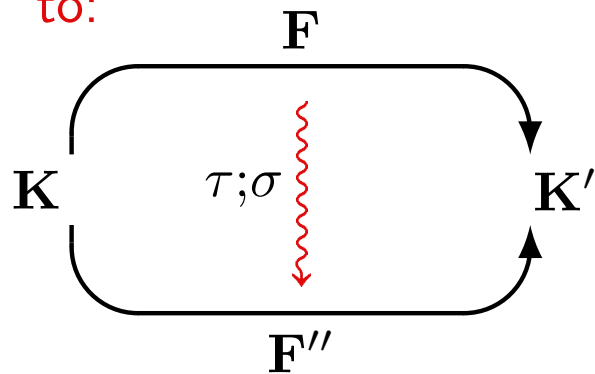
Compositions

vertical composition:

From:



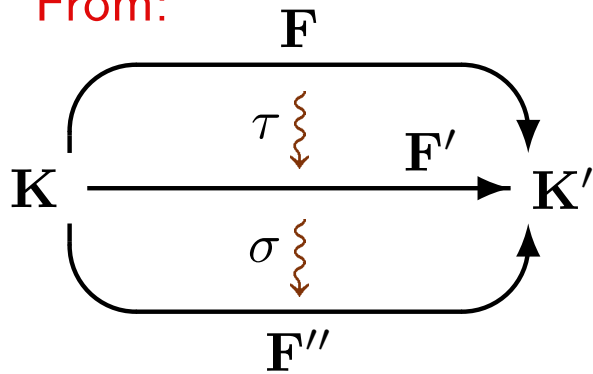
to:



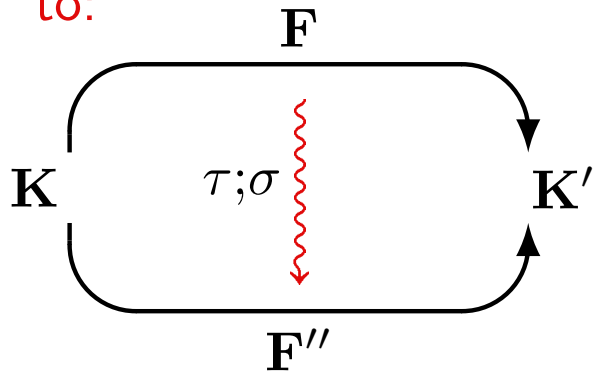
Compositions

vertical composition:

From:



to:

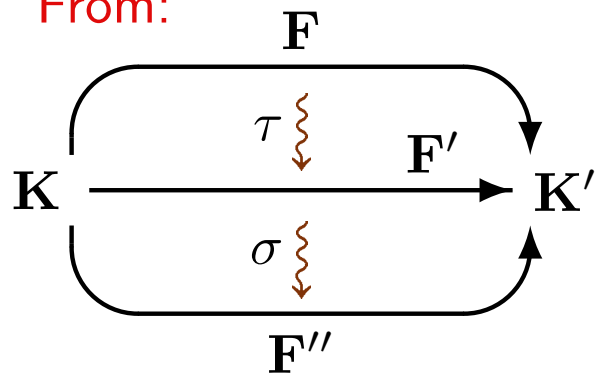


horizontal composition:

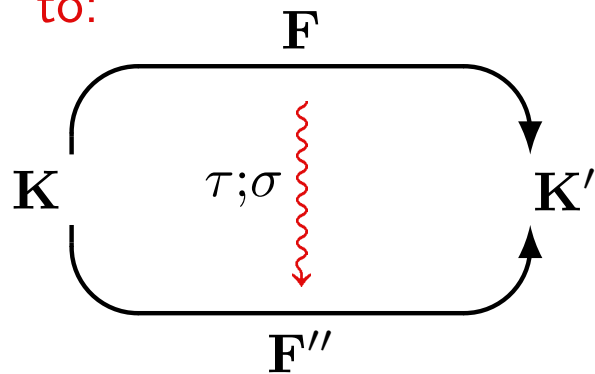
Compositions

vertical composition:

From:

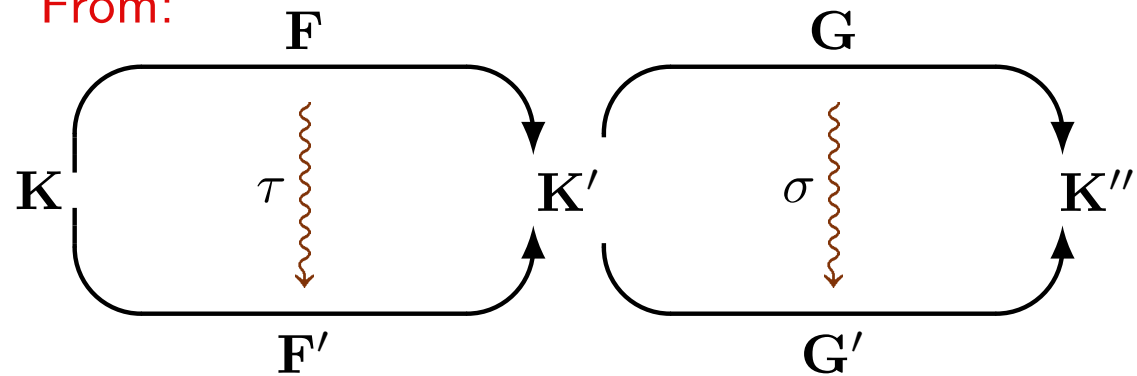


to:



horizontal composition:

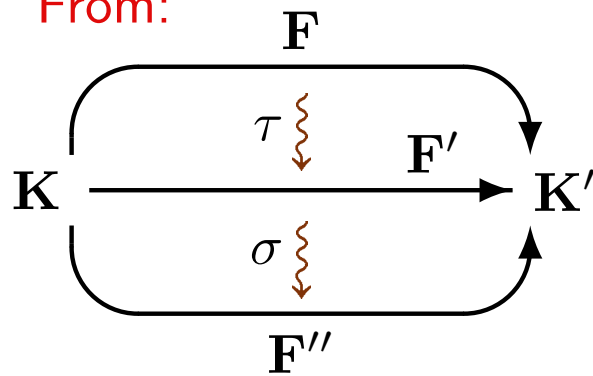
From:



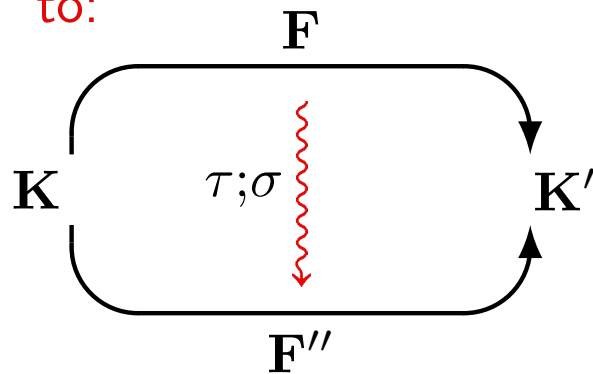
Compositions

vertical composition:

From:

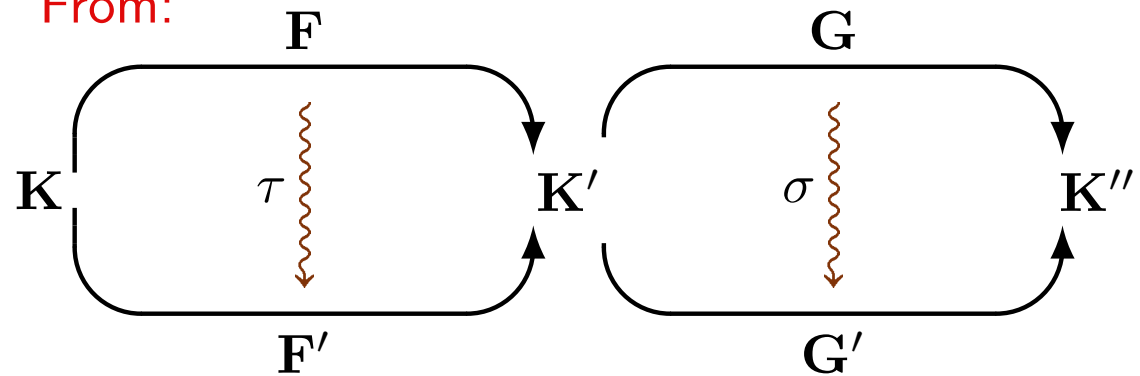


to:

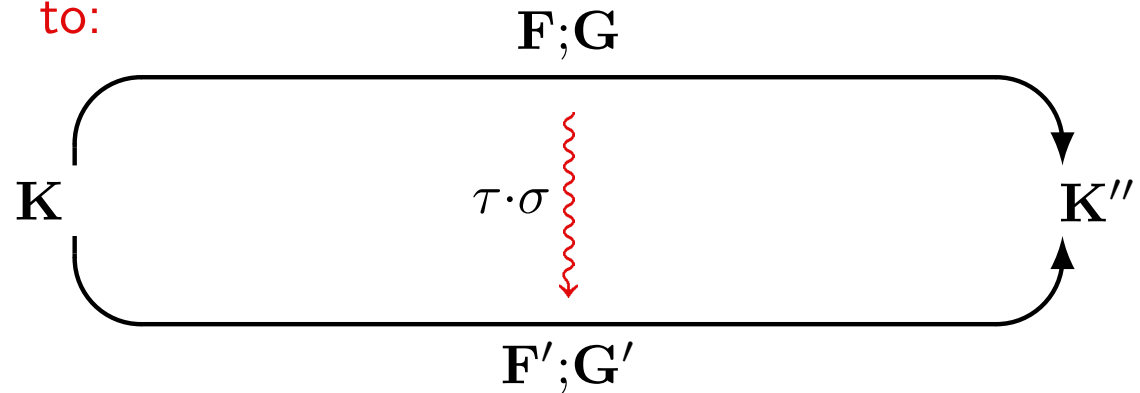


horizontal composition:

From:

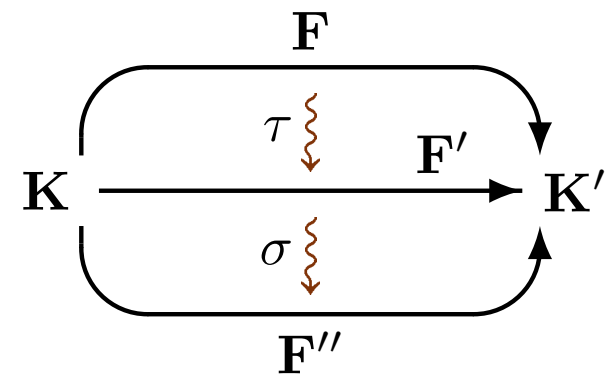


to:



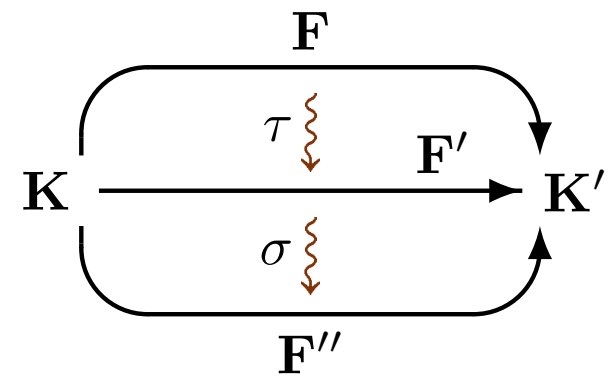
Vertical composition

Vertical composition



The *vertical composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{F}' \rightarrow \mathbf{F}''$ between parallel functors $\mathbf{F}, \mathbf{F}', \mathbf{F}'': \mathbf{K} \rightarrow \mathbf{K}'$

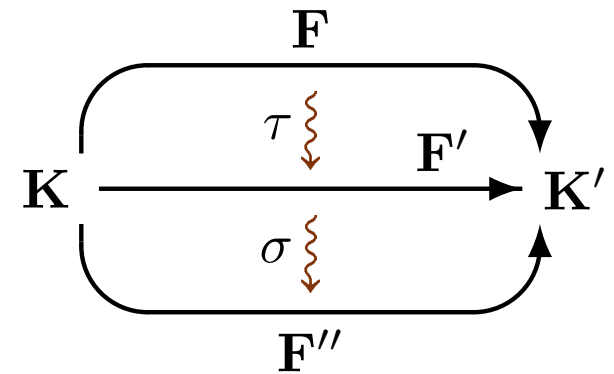
Vertical composition



The *vertical composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{F}' \rightarrow \mathbf{F}''$ between parallel functors $\mathbf{F}, \mathbf{F}', \mathbf{F}'': \mathbf{K} \rightarrow \mathbf{K}'$

$$\tau; \sigma: \mathbf{F} \rightarrow \mathbf{F}''$$

Vertical composition

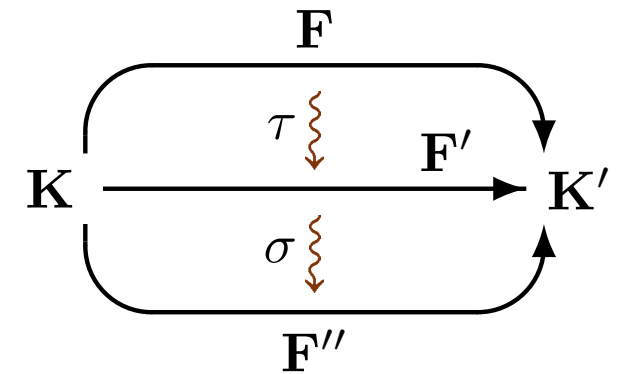


The *vertical composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{F}' \rightarrow \mathbf{F}''$ between parallel functors $\mathbf{F}, \mathbf{F}', \mathbf{F}'': \mathbf{K} \rightarrow \mathbf{K}'$

$$\tau; \sigma: \mathbf{F} \rightarrow \mathbf{F}''$$

is a natural transformation given by $(\tau; \sigma)_A = \tau_A; \sigma_A$ for all $A \in |\mathbf{K}|$.

Vertical composition



The *vertical composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{F}' \rightarrow \mathbf{F}''$ between parallel functors $\mathbf{F}, \mathbf{F}', \mathbf{F}'': \mathbf{K} \rightarrow \mathbf{K}'$

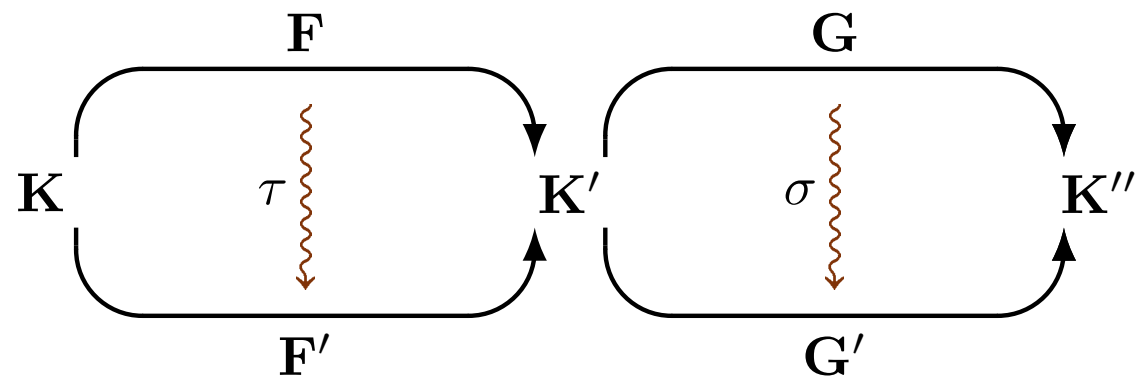
$$\tau; \sigma: \mathbf{F} \rightarrow \mathbf{F}''$$

is a natural transformation given by $(\tau; \sigma)_A = \tau_A; \sigma_A$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccccc}
 \mathbf{K}: & & \mathbf{K}': & & \\
 A & & \mathbf{F}(A) & \xrightarrow{\tau_A} & \mathbf{F}'(A) & \xrightarrow{\sigma_A} & \mathbf{F}''(A) \\
 \downarrow f & & \downarrow \mathbf{F}(f) & & \downarrow \mathbf{F}'(f) & & \downarrow \mathbf{F}''(f) \\
 B & & \mathbf{F}(B) & \xrightarrow{\tau_B} & \mathbf{F}'(B) & \xrightarrow{\sigma_B} & \mathbf{F}''(B)
 \end{array}$$

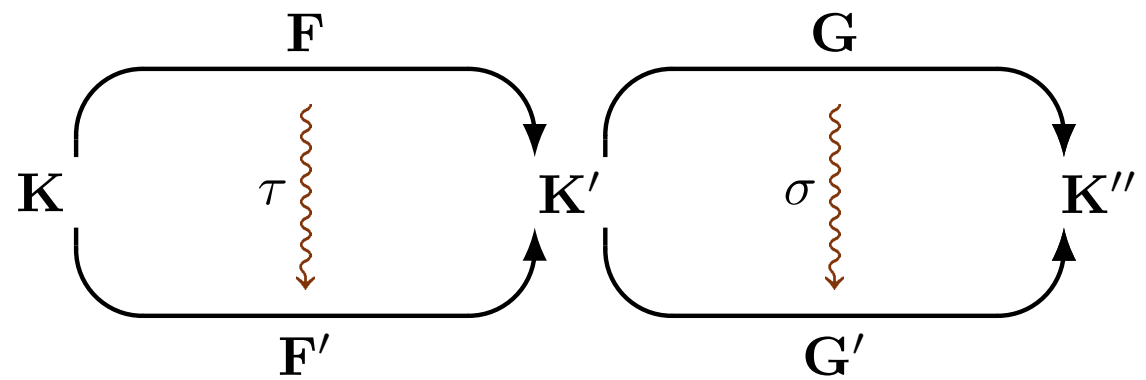
Horizontal composition

Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

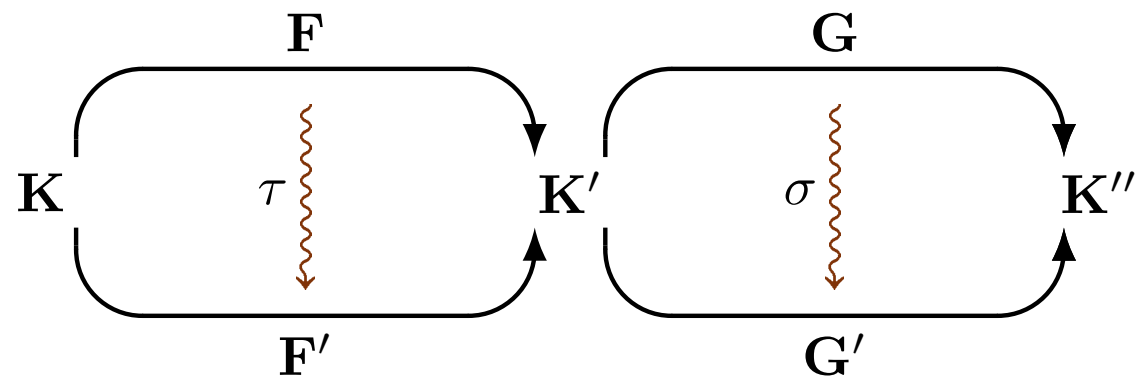
Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

Horizontal composition



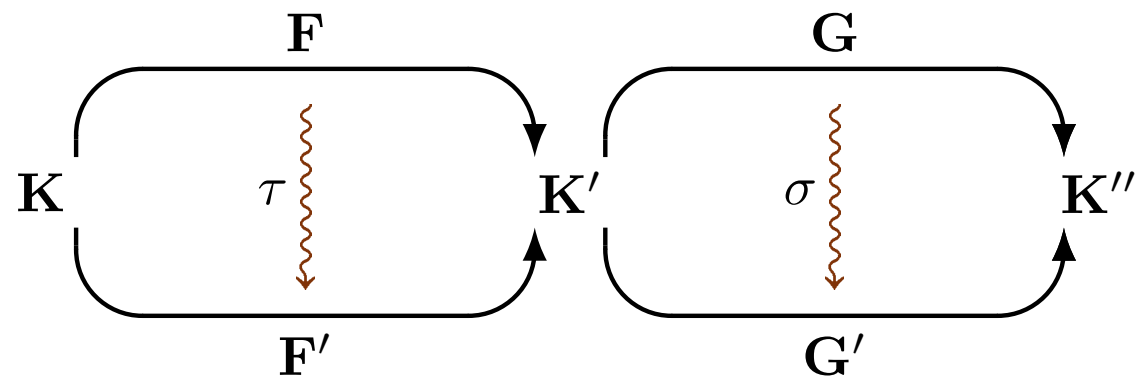
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc} \mathbf{K}'': & & \\ \mathbf{G}(\mathbf{F}(A)) & \xrightarrow{(\tau \cdot \sigma)_A} & \mathbf{G}'(\mathbf{F}'(A)) \end{array}$$

Horizontal composition



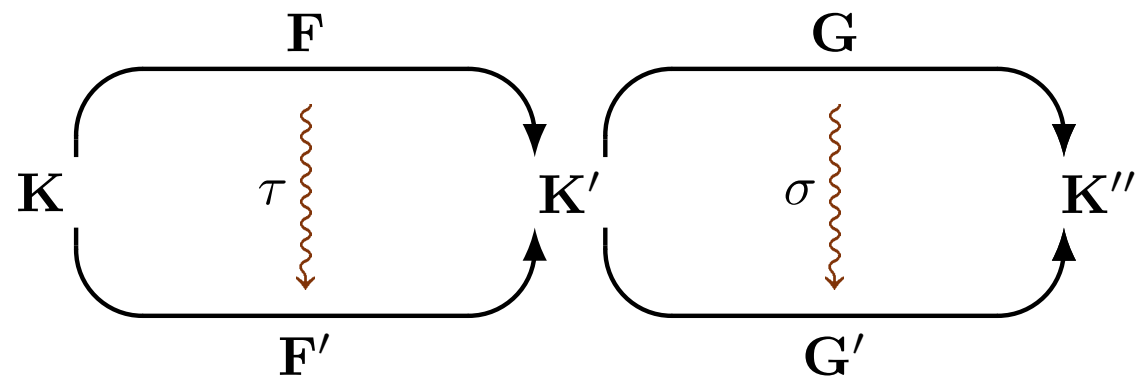
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc} \mathbf{K}'': & & \\ \mathbf{G}(\mathbf{F}(A)) & \xrightarrow{\sigma_{\mathbf{F}(A)}} & \mathbf{G}'(\mathbf{F}(A)) \\ & \searrow (\tau \cdot \sigma)_A & \\ & & \mathbf{G}'(\mathbf{F}'(A)) \end{array}$$

Horizontal composition



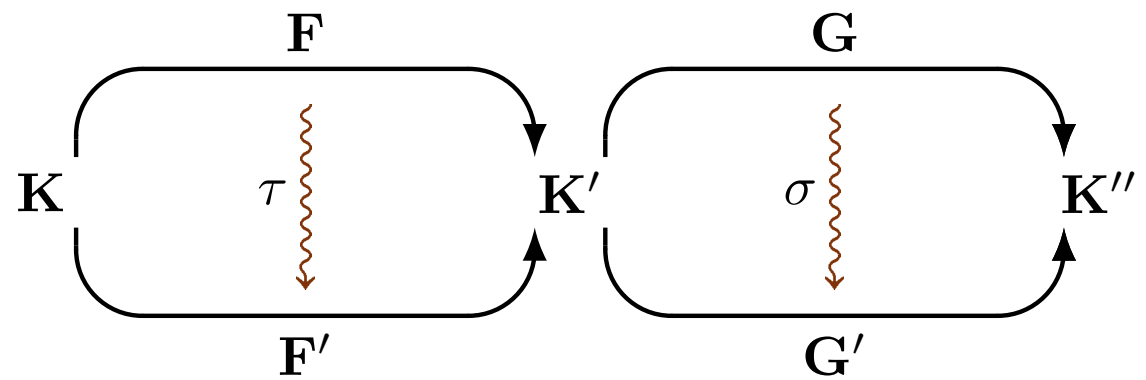
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc}
 \mathbf{K}': & & \mathbf{K}'': \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \searrow (\tau \cdot \sigma)_A \\
 \mathbf{F}'(A) & & \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Horizontal composition



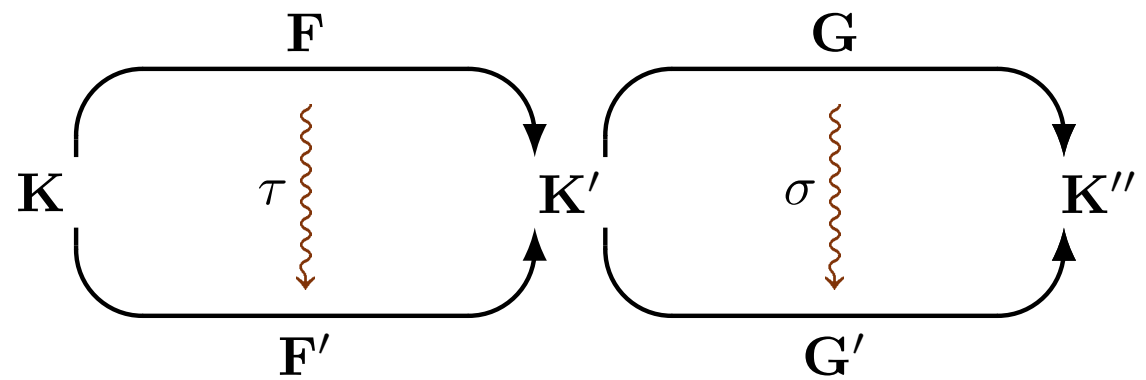
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc}
 \mathbf{K}': & & \mathbf{K}'': \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \searrow (\tau \cdot \sigma)_A \quad \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Horizontal composition



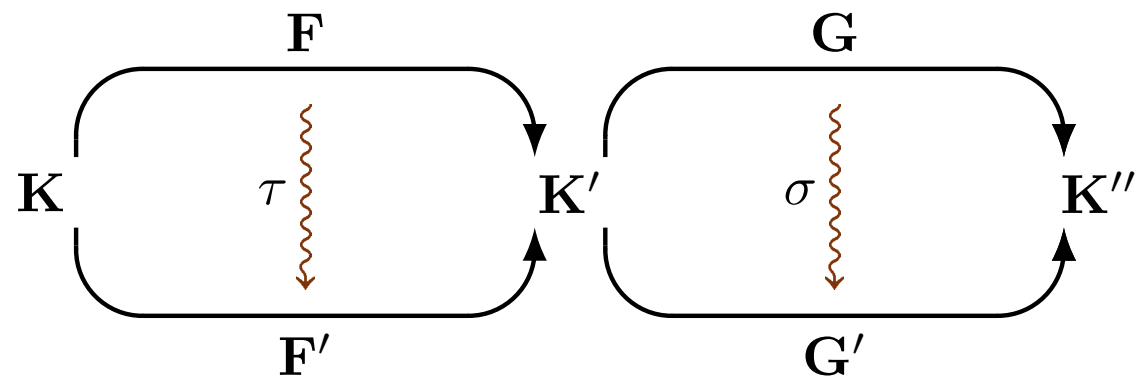
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc}
 \mathbf{K}': & & \mathbf{K}'': \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) \quad \searrow (\tau \cdot \sigma)_A \quad \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) \xrightarrow{\sigma_{\mathbf{F}'(A)}} \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc}
 \mathbf{K}': & & \mathbf{K}'': \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) \quad \searrow (\tau \cdot \sigma)_A \quad \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) \xrightarrow{\sigma_{\mathbf{F}'(A)}} \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

K:

A

K'':

$\mathbf{G}(\mathbf{F}(A))$

$(\tau \cdot \sigma)_A$

\longrightarrow

$\mathbf{G}'(\mathbf{F}'(A))$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

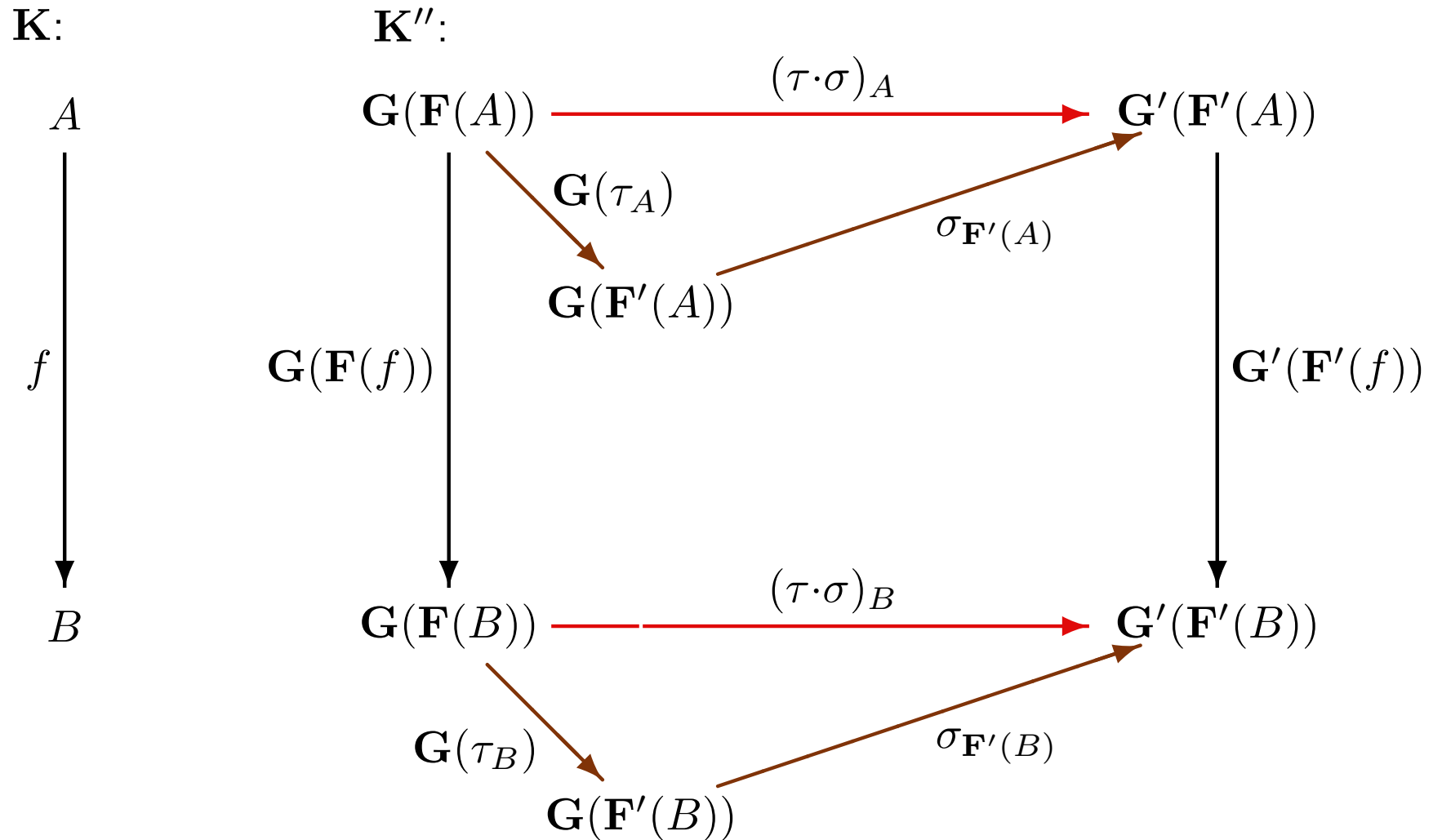
$$\begin{array}{ccc} \mathbf{K}: & \mathbf{K}'': & \\ A & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{(\tau \cdot \sigma)_A} & \mathbf{G}'(\mathbf{F}'(A)) \end{array}$$

$$\begin{array}{ccc} & & \\ B & \mathbf{G}(\mathbf{F}(B)) \xrightarrow{(\tau \cdot \sigma)_B} & \mathbf{G}'(\mathbf{F}'(B)) \end{array}$$

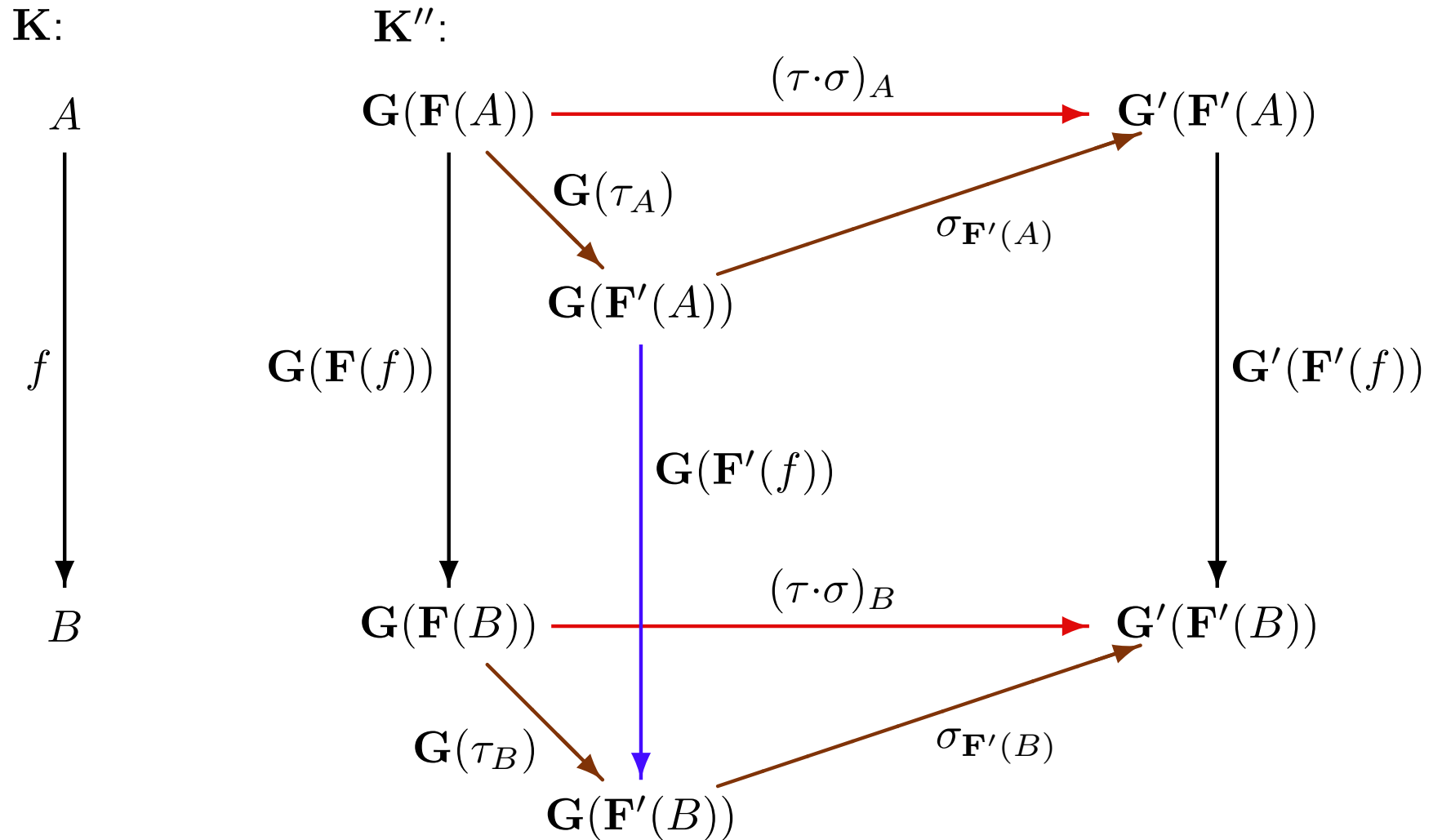
Show that indeed, $\tau \cdot \sigma$ is a natural transformation

$$\begin{array}{ccccc}
 \mathbf{K}: & & \mathbf{K}'': & & \\
 A & & \mathbf{G}(\mathbf{F}(A)) & \xrightarrow{(\tau \cdot \sigma)_A} & \mathbf{G}'(\mathbf{F}'(A)) \\
 \downarrow f & & \downarrow \mathbf{G}(\mathbf{F}(f)) & & \downarrow \mathbf{G}'(\mathbf{F}'(f)) \\
 B & & \mathbf{G}(\mathbf{F}(B)) & \xrightarrow{(\tau \cdot \sigma)_B} & \mathbf{G}'(\mathbf{F}'(B))
 \end{array}$$

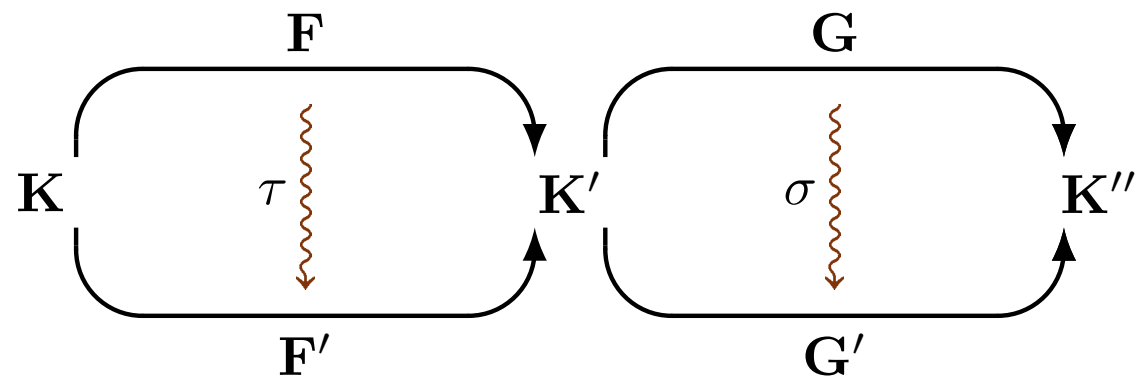
Show that indeed, $\tau \cdot \sigma$ is a natural transformation



Show that indeed, $\tau \cdot \sigma$ is a natural transformation



Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

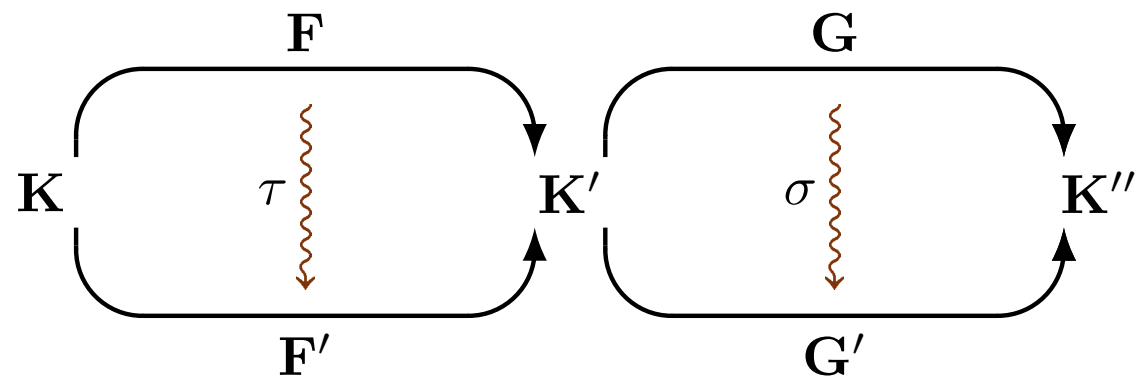
$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

$$\begin{array}{ccc}
 \mathbf{K}': & & \mathbf{K}'': \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) \quad \searrow (\tau \cdot \sigma)_A \quad \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) \xrightarrow{\sigma_{\mathbf{F}'(A)}} \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

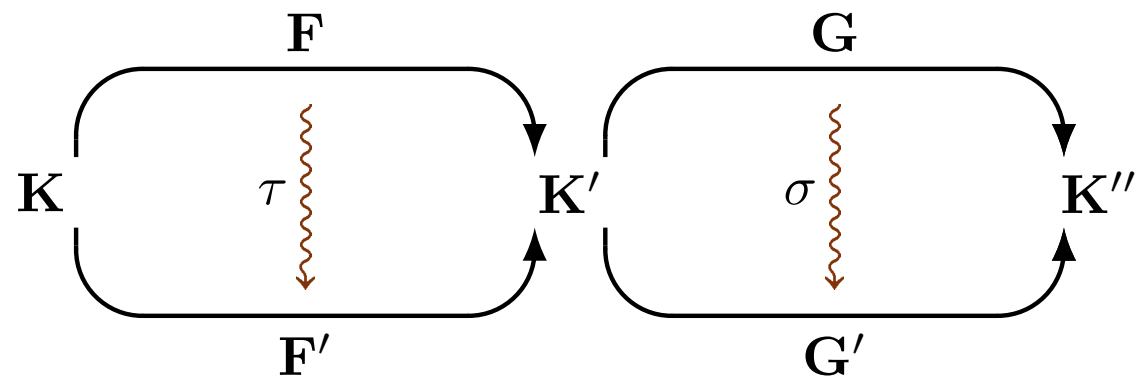
is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

Multiplication by functor:

$$\begin{array}{ccccc}
 \mathbf{K}': & & \mathbf{K}'': & & \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) & \xrightarrow{\sigma_{\mathbf{F}(A)}} & \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) & \searrow (\tau \cdot \sigma)_A & \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) & \xrightarrow{\sigma_{\mathbf{F}'(A)}} & \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

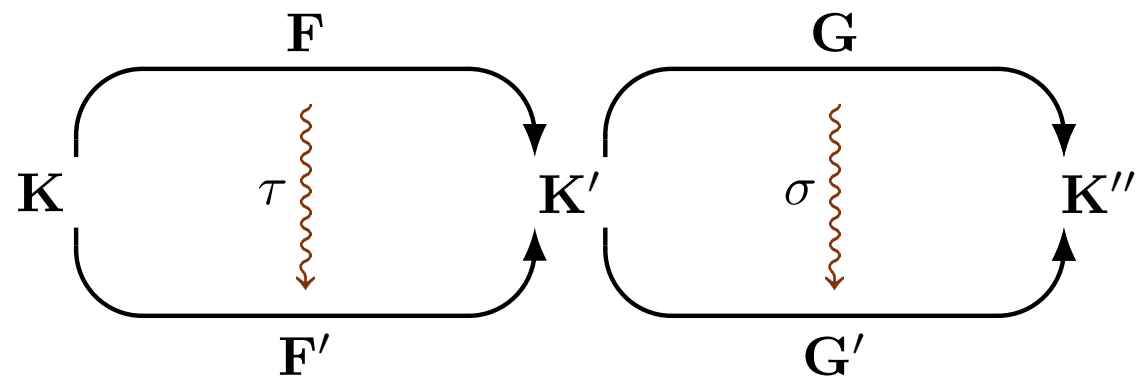
Multiplication by functor:

$$- \quad \tau \cdot \mathbf{G} = \tau \cdot id_{\mathbf{G}}: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G},$$

$$\begin{array}{ccccc}
 \mathbf{K}': & & \mathbf{K}'': & & \\
 \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) & \xrightarrow{\sigma_{\mathbf{F}(A)}} & \mathbf{G}'(\mathbf{F}(A)) \\
 \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) & \searrow (\tau \cdot \sigma)_A & \downarrow \mathbf{G}'(\tau_A) \\
 \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) & \xrightarrow{\sigma_{\mathbf{F}'(A)}} & \mathbf{G}'(\mathbf{F}'(A))
 \end{array}$$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Horizontal composition



The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

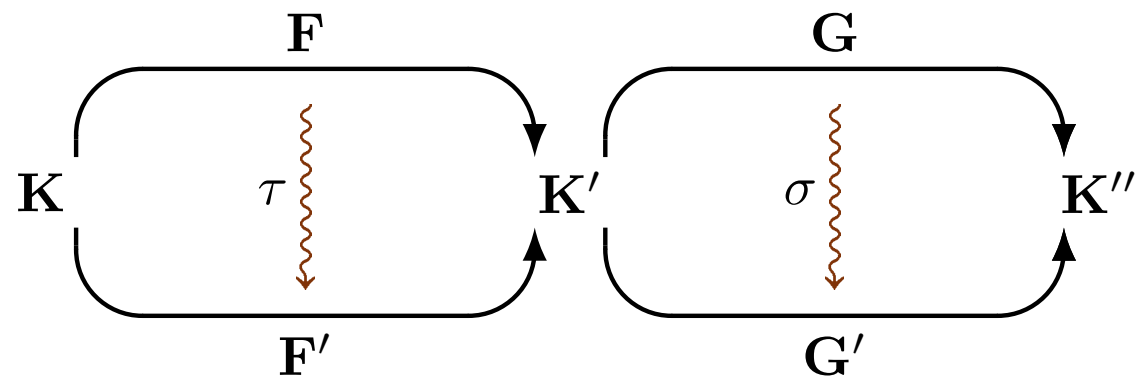
Multiplication by functor:

$$\begin{aligned} - \quad \tau \cdot \mathbf{G} &= \tau \cdot id_{\mathbf{G}}: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}, \\ \text{i.e., } (\tau \cdot \mathbf{G})_A &= \mathbf{G}(\tau_A) \end{aligned}$$

$$\begin{array}{ccc} \mathbf{K}': & & \mathbf{K}'': \\ \mathbf{F}(A) & & \mathbf{G}(\mathbf{F}(A)) \xrightarrow{\sigma_{\mathbf{F}(A)}} \mathbf{G}'(\mathbf{F}(A)) \\ \downarrow \tau_A & & \downarrow \mathbf{G}(\tau_A) \quad \searrow (\tau \cdot \sigma)_A \quad \downarrow \mathbf{G}'(\tau_A) \\ \mathbf{F}'(A) & & \mathbf{G}(\mathbf{F}'(A)) \xrightarrow{\sigma_{\mathbf{F}'(A)}} \mathbf{G}'(\mathbf{F}'(A)) \end{array}$$

Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Horizontal composition



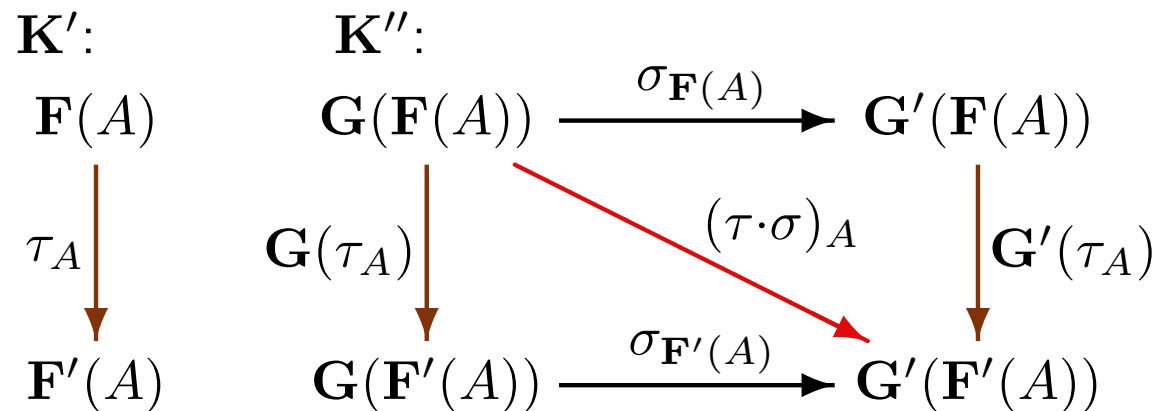
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}'$, $\mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

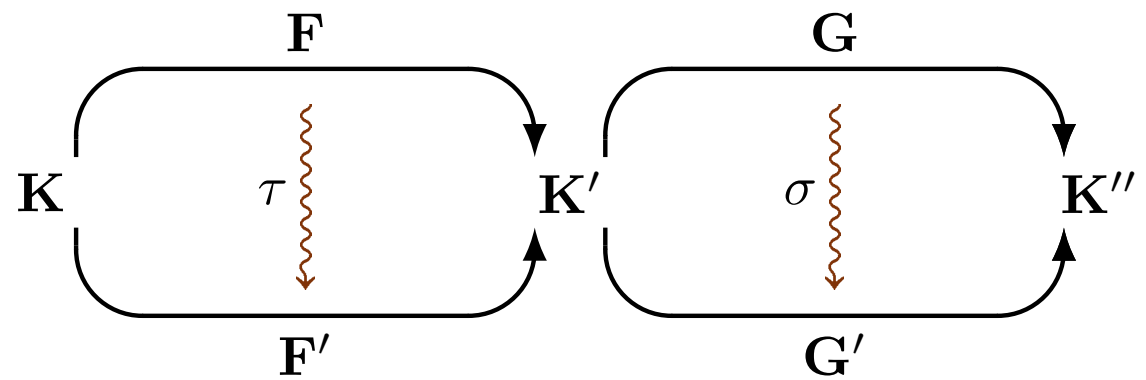
Multiplication by functor:

- $\tau \cdot \mathbf{G} = \tau \cdot id_{\mathbf{G}}: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}$,
i.e., $(\tau \cdot \mathbf{G})_A = \mathbf{G}(\tau_A)$
- $\mathbf{F} \cdot \sigma = id_{\mathbf{F}} \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}; \mathbf{G}'$,



Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Horizontal composition



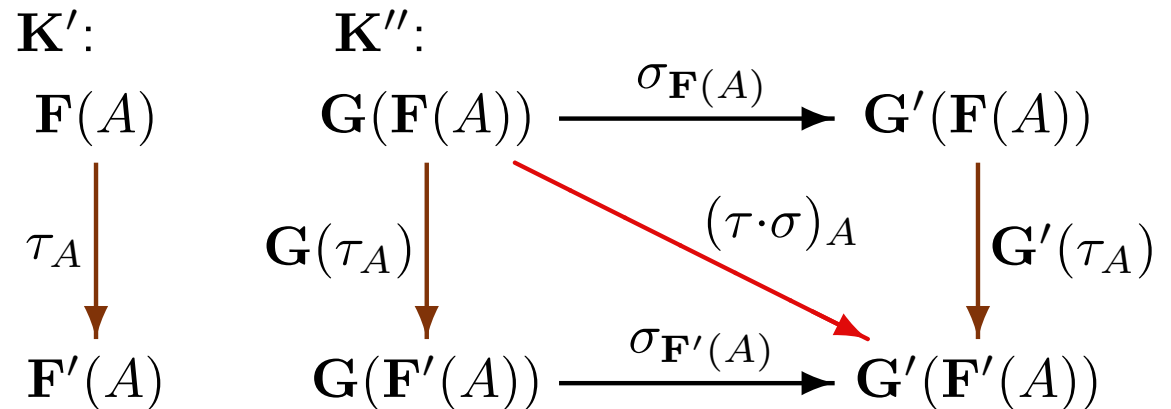
The *horizontal composition* of natural transformations $\tau: \mathbf{F} \rightarrow \mathbf{F}'$ and $\sigma: \mathbf{G} \rightarrow \mathbf{G}'$ between composable pairs of parallel functors $\mathbf{F}, \mathbf{F}': \mathbf{K} \rightarrow \mathbf{K}', \mathbf{G}, \mathbf{G}': \mathbf{K}' \rightarrow \mathbf{K}''$

$$\tau \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}'$$

is a natural transformation given by $(\tau \cdot \sigma)_A = \mathbf{G}(\tau_A); \sigma_{\mathbf{F}'(A)} = \sigma_{\mathbf{F}(A)}; \mathbf{G}'(\tau_A)$ for all $A \in |\mathbf{K}|$.

Multiplication by functor:

- $\tau \cdot \mathbf{G} = \tau \cdot id_{\mathbf{G}}: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}'; \mathbf{G}$,
i.e., $(\tau \cdot \mathbf{G})_A = \mathbf{G}(\tau_A)$
- $\mathbf{F} \cdot \sigma = id_{\mathbf{F}} \cdot \sigma: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}; \mathbf{G}'$,
i.e., $(\mathbf{F} \cdot \sigma)_A = \sigma_{\mathbf{F}(A)}$



Show that indeed, $\tau \cdot \sigma$ is a natural transformation

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of S -sorted sets, \mathbf{Set}^S , as a functor category.

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of S -sorted sets, \mathbf{Set}^S , as a functor category.
- Check whether $\mathbf{K}^{\mathbf{K}'}$ is (finitely) (co)complete whenever \mathbf{K} is so.

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $F, G: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $F \times G: \mathbf{K}' \rightarrow \mathbf{K}$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$

— for $A' \in |\mathbf{K}'|$, $(\mathbf{F} \times \mathbf{G})(A') = \mathbf{F}(A') \times \mathbf{G}(A')$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$

- for $A' \in |\mathbf{K}'|$, $(\mathbf{F} \times \mathbf{G})(A') = \mathbf{F}(A') \times \mathbf{G}(A')$
- for $f: A' \rightarrow B'$,
 $(\mathbf{F} \times \mathbf{G})(f) = \langle \pi_{\mathbf{F}(A')}; \mathbf{F}(f), \pi_{\mathbf{G}(A')}; \mathbf{G}(f) \rangle: (\mathbf{F} \times \mathbf{G})(A') \rightarrow (\mathbf{F} \times \mathbf{G})(B')$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$

– for $A' \in |\mathbf{K}'|$, $(\mathbf{F} \times \mathbf{G})(A') = \mathbf{F}(A') \times \mathbf{G}(A')$

– for $f: A' \rightarrow B'$,

$(\mathbf{F} \times \mathbf{G})(f) = \langle \pi_{\mathbf{F}(A')}; \mathbf{F}(f), \pi_{\mathbf{G}(A')}; \mathbf{G}(f) \rangle: (\mathbf{F} \times \mathbf{G})(A') \rightarrow (\mathbf{F} \times \mathbf{G})(B')$

$$\begin{array}{ccc}
 & \mathbf{F}(A') & \xrightarrow{\mathbf{F}(f)} & \mathbf{F}(B') \\
 & \nearrow \pi_{\mathbf{F}(A')} & & \nearrow \pi_{\mathbf{F}(B')} \\
 \mathbf{F}(A') \times \mathbf{G}(A') & \xrightarrow{\quad} & \mathbf{F}(B') \times \mathbf{G}(B') \\
 & \searrow \pi_{\mathbf{G}(A')} & & \searrow \pi_{\mathbf{G}(B')} \\
 & \mathbf{G}(A') & \xrightarrow{\mathbf{G}(f)} & \mathbf{G}(B')
 \end{array}$$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$

– for $A' \in |\mathbf{K}'|$, $(\mathbf{F} \times \mathbf{G})(A') = \mathbf{F}(A') \times \mathbf{G}(A')$

– for $f: A' \rightarrow B'$,

$$(\mathbf{F} \times \mathbf{G})(f) = \langle \pi_{\mathbf{F}(A')}; \mathbf{F}(f), \pi_{\mathbf{G}(A')}; \mathbf{G}(f) \rangle: (\mathbf{F} \times \mathbf{G})(A') \rightarrow (\mathbf{F} \times \mathbf{G})(B')$$

$$\begin{array}{ccc}
 & \mathbf{F}(A') & \xrightarrow{\mathbf{F}(f)} & \mathbf{F}(B') \\
 \nearrow \pi_{\mathbf{F}(A')} & & & \nearrow \pi_{\mathbf{F}(B')} \\
 \mathbf{F}(A') \times \mathbf{G}(A') & \xrightarrow{\quad} & \mathbf{F}(B') \times \mathbf{G}(B') \\
 \searrow \pi_{\mathbf{G}(A')} & & & \searrow \pi_{\mathbf{G}(B')} \\
 & \mathbf{G}(A') & \xrightarrow{\mathbf{G}(f)} & \mathbf{G}(B')
 \end{array}$$

To be checked:

$$(\mathbf{F} \times \mathbf{G})(id_{A'}) = id_{(\mathbf{F} \times \mathbf{G})(A')}, \text{ and } \\
 (\mathbf{F} \times \mathbf{G})(f;g) = (\mathbf{F} \times \mathbf{G})(f);(\mathbf{F} \times \mathbf{G})(g)$$

Theorem: If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

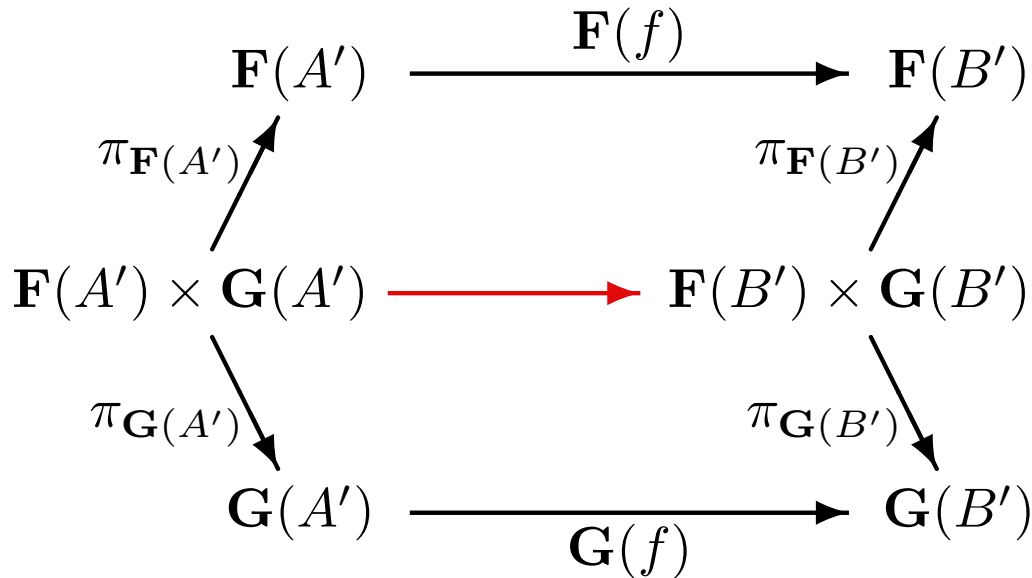
Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$

– for $A' \in |\mathbf{K}'|$, $(\mathbf{F} \times \mathbf{G})(A') = \mathbf{F}(A') \times \mathbf{G}(A')$

– for $f: A' \rightarrow B'$,

$$(\mathbf{F} \times \mathbf{G})(f) = \langle \pi_{\mathbf{F}(A')}; \mathbf{F}(f), \pi_{\mathbf{G}(A')}; \mathbf{G}(f) \rangle: (\mathbf{F} \times \mathbf{G})(A') \rightarrow (\mathbf{F} \times \mathbf{G})(B')$$



To be checked:

$$(\mathbf{F} \times \mathbf{G})(id_{A'}) = id_{(\mathbf{F} \times \mathbf{G})(A')}, \text{ and } (\mathbf{F} \times \mathbf{G})(f;g) = (\mathbf{F} \times \mathbf{G})(f);(\mathbf{F} \times \mathbf{G})(g)$$

This yields natural transformations:

$$\pi_{\mathbf{F}} = \langle \pi_{\mathbf{F}(A')} \rangle_{A' \in |\mathbf{K}'|}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$$

$$\pi_{\mathbf{G}} = \langle \pi_{\mathbf{G}(A')} \rangle_{A' \in |\mathbf{K}'|}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$
with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $C_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$
with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers:

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$
with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations
 $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$
with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations
 $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:
— for $A' \in |\mathbf{K}'|$, $\delta_{A'}: \mathbf{H}(A') \rightarrow \mathbf{F}(A')$ is equaliser of $\tau_{A'}, \sigma_{A'}: \mathbf{F}(A') \rightarrow \mathbf{G}(A')$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$
with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations
 $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:

- for $A' \in |\mathbf{K}'|$, $\delta_{A'}: \mathbf{H}(A') \rightarrow \mathbf{F}(A')$ is equaliser of $\tau_{A'}, \sigma_{A'}: \mathbf{F}(A') \rightarrow \mathbf{G}(A')$
- for $f: A' \rightarrow B'$, $\mathbf{H}(f): \mathbf{H}(A') \rightarrow \mathbf{H}(B')$ is s. t. $\delta_{A'}; \mathbf{H}(f) = \mathbf{G}(f); \delta_{B'}$.

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:

$$\begin{array}{ccc}
 \mathbf{G}(A') & \xrightarrow{\mathbf{G}(f)} & \mathbf{G}(B') \\
 \uparrow \tau_{A'} \quad \uparrow \sigma_{A'} & & \uparrow \tau_{B'} \quad \uparrow \sigma_{B'} \\
 \mathbf{F}(A') & \xrightarrow{\mathbf{F}(f)} & \mathbf{F}(B') \\
 \uparrow \delta_{A'} & & \uparrow \delta_{B'} \\
 \mathbf{H}(A') & \xrightarrow{\mathbf{H}(f)} & \mathbf{H}(B')
 \end{array}$$

Theorem: *If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.*

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:

$$\begin{array}{ccc}
 \mathbf{G}(A') & \xrightarrow{\mathbf{G}(f)} & \mathbf{G}(B') \\
 \uparrow \tau_{A'} \quad \uparrow \sigma_{A'} & & \uparrow \tau_{B'} \quad \uparrow \sigma_{B'} \\
 \mathbf{F}(A') & \xrightarrow{\mathbf{F}(f)} & \mathbf{F}(B') \\
 \uparrow \delta_{A'} & & \uparrow \delta_{B'} \\
 \mathbf{H}(A') & \xrightarrow{\mathbf{H}(f)} & \mathbf{H}(B')
 \end{array}$$

To be checked:

$$\mathbf{H}(id_{A'}) = id_{\mathbf{H}(A')}, \text{ and}$$

$$\mathbf{H}(f;g) = \mathbf{H}(f); \mathbf{H}(g)$$

Theorem: If \mathbf{K} is finitely complete then $\mathbf{K}^{\mathbf{K}'}$ is finitely complete as well.

Proof (idea): Define a terminal object, binary products and equalisers in $\mathbf{K}^{\mathbf{K}'}$:

Terminal object: $\mathbf{C}_T: \mathbf{K}' \rightarrow \mathbf{K}$ is terminal in $\mathbf{K}^{\mathbf{K}'}$, where T is terminal in \mathbf{K} .

Products: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, their product in $\mathbf{K}^{\mathbf{K}'}$ is $\mathbf{F} \times \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ with product projections $\pi_{\mathbf{F}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{F}$ and $\pi_{\mathbf{G}}: (\mathbf{F} \times \mathbf{G}) \rightarrow \mathbf{G}$.

Equalisers: Given $\mathbf{F}, \mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, equaliser of two natural transformations $\tau, \sigma: \mathbf{F} \rightarrow \mathbf{G}$ is $\delta: \mathbf{H} \rightarrow \mathbf{F}$, where:

$$\begin{array}{ccc}
 \mathbf{G}(A') & \xrightarrow{\mathbf{G}(f)} & \mathbf{G}(B') \\
 \uparrow \tau_{A'} \quad \uparrow \sigma_{A'} & & \uparrow \tau_{B'} \quad \uparrow \sigma_{B'} \\
 \mathbf{F}(A') & \xrightarrow{\mathbf{F}(f)} & \mathbf{F}(B') \\
 \uparrow \delta_{A'} & & \uparrow \delta_{B'} \\
 \mathbf{H}(A') & \xrightarrow{\mathbf{H}(f)} & \mathbf{H}(B')
 \end{array}$$

To be checked:

$$\begin{aligned}
 \mathbf{H}(id_{A'}) &= id_{\mathbf{H}(A')}, \text{ and} \\
 \mathbf{H}(f;g) &= \mathbf{H}(f); \mathbf{H}(g)
 \end{aligned}$$

This yields a natural transformation:

$$\delta = \langle \delta_{A'} \rangle_{A' \in |\mathbf{K}'|}: \mathbf{H} \rightarrow \mathbf{F}$$

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Define (arbitrary) products and equalisers in $\mathbf{K}^{\mathbf{K}'}$, as for the finite case.

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:
 - for $A' \in |\mathbf{K}'|$, let $\alpha^{A'}: \mathbf{X}(A') \rightarrow \mathbf{D}(A')$ be the limit of $\mathbf{D}(A')$ in \mathbf{K}

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:
 - for $A' \in |\mathbf{K}'|$, let $\alpha^{A'}: \mathbf{X}(A') \rightarrow \mathbf{D}(A')$ be the limit of $\mathbf{D}(A')$ in \mathbf{K}
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , let $\mathbf{X}(f): \mathbf{X}(A') \rightarrow \mathbf{X}(B')$ be unique such that $\alpha^{A'}; \mathbf{D}(f) = \mathbf{X}(f); \alpha^{B'}$ (given by the limit property of $\alpha^{B'}$)

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:
 - for $A' \in |\mathbf{K}'|$, let $\alpha^{A'}: \mathbf{X}(A') \rightarrow \mathbf{D}(A')$ be the limit of $\mathbf{D}(A')$ in \mathbf{K}
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , let $\mathbf{X}(f): \mathbf{X}(A') \rightarrow \mathbf{X}(B')$ be unique such that $\alpha^{A'}; \mathbf{D}(f) = \mathbf{X}(f); \alpha^{B'}$ (given by the limit property of $\alpha^{B'}$)
 - define $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ by $(\alpha_n)_{A'} = (\alpha^{A'})_n$, for $A' \in |\mathbf{K}'|$

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:
 - for $A' \in |\mathbf{K}'|$, let $\alpha^{A'}: \mathbf{X}(A') \rightarrow \mathbf{D}(A')$ be the limit of $\mathbf{D}(A')$ in \mathbf{K}
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , let $\mathbf{X}(f): \mathbf{X}(A') \rightarrow \mathbf{X}(B')$ be unique such that $\alpha^{A'}; \mathbf{D}(f) = \mathbf{X}(f); \alpha^{B'}$ (given by the limit property of $\alpha^{B'}$)
 - define $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ by $(\alpha_n)_{A'} = (\alpha^{A'})_n$, for $A' \in |\mathbf{K}'|$
 - check that $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ is a functor, and $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ are natural transformations

Theorem: *If \mathbf{K} is complete then $\mathbf{K}^{\mathbf{K}'}$ is complete as well.*

Proof (idea): Or proceed with limit construction for an arbitrary diagram:

- Let \mathbf{D} be a diagram in $\mathbf{K}^{\mathbf{K}'}$ with nodes $n \in N$ and edges $e \in E$.
 - for $A' \in |\mathbf{K}'|$, define $\mathbf{D}(A')$ to be a diagram in \mathbf{K} with $\mathbf{D}(A')_n = \mathbf{D}_n(A')$ and $\mathbf{D}(A')_e = (\mathbf{D}_e)_{A'}$
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , define a diagram morphism $\mathbf{D}(f): \mathbf{D}(A') \rightarrow \mathbf{D}(B')$ by $\mathbf{D}(f)_n = \mathbf{D}_n(f)$
- Define a functor $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural transformations $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ as follows:
 - for $A' \in |\mathbf{K}'|$, let $\alpha^{A'}: \mathbf{X}(A') \rightarrow \mathbf{D}(A')$ be the limit of $\mathbf{D}(A')$ in \mathbf{K}
 - for $f: A' \rightarrow B'$ in \mathbf{K}' , let $\mathbf{X}(f): \mathbf{X}(A') \rightarrow \mathbf{X}(B')$ be unique such that $\alpha^{A'}; \mathbf{D}(f) = \mathbf{X}(f); \alpha^{B'}$ (given by the limit property of $\alpha^{B'}$)
 - define $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ by $(\alpha_n)_{A'} = (\alpha^{A'})_n$, for $A' \in |\mathbf{K}'|$
 - check that $\mathbf{X}: \mathbf{K}' \rightarrow \mathbf{K}$ is a functor, and $\alpha_n: \mathbf{X} \rightarrow \mathbf{D}_n$ are natural transformations
- Prove that $\alpha: \mathbf{X} \rightarrow \mathbf{D}$ is a limit of \mathbf{D} in $\mathbf{K}^{\mathbf{K}'}$.

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of S -sorted sets, \mathbf{Set}^S , as a functor category.
- Check whether $\mathbf{K}^{\mathbf{K}'}$ is (finitely) (co)complete whenever \mathbf{K} is so.
- Show how any functor $\mathbf{F}: \mathbf{K}'' \rightarrow \mathbf{K}'$ induces a functor $(\mathbf{F}; -): \mathbf{K}^{\mathbf{K}'} \rightarrow \mathbf{K}^{\mathbf{K}''}$,

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of S -sorted sets, \mathbf{Set}^S , as a functor category.
- Check whether $\mathbf{K}^{\mathbf{K}'}$ is (finitely) (co)complete whenever \mathbf{K} is so.
- Show how any functor $\mathbf{F}: \mathbf{K}'' \rightarrow \mathbf{K}'$ induces a functor $(\mathbf{F}; -): \mathbf{K}^{\mathbf{K}'} \rightarrow \mathbf{K}^{\mathbf{K}''}$, where for $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, $(\mathbf{F}; -)(\mathbf{G}) = \mathbf{F}; \mathbf{G}: \mathbf{K}'' \rightarrow \mathbf{K}$,

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of S -sorted sets, \mathbf{Set}^S , as a functor category.
- Check whether $\mathbf{K}^{\mathbf{K}'}$ is (finitely) (co)complete whenever \mathbf{K} is so.
- Show how any functor $\mathbf{F}: \mathbf{K}'' \rightarrow \mathbf{K}'$ induces a functor $(\mathbf{F};_-): \mathbf{K}^{\mathbf{K}'} \rightarrow \mathbf{K}^{\mathbf{K}''}$, where for $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, $(\mathbf{F};_-)(\mathbf{G}) = \mathbf{F};\mathbf{G}: \mathbf{K}'' \rightarrow \mathbf{K}$, and for $\tau: \mathbf{G} \rightarrow \mathbf{G}' (: \mathbf{K}' \rightarrow \mathbf{K})$, $(\mathbf{F};_-)(\tau) = \mathbf{F} \cdot \tau: \mathbf{F};\mathbf{G} \rightarrow \mathbf{F};\mathbf{G}'$.

Functor categories

Given two categories \mathbf{K}, \mathbf{K}' , define the *category of functors from \mathbf{K}' to \mathbf{K}* , $\mathbf{K}^{\mathbf{K}'}$, as follows:

- objects: functors from \mathbf{K}' to \mathbf{K}
- morphisms: natural transformations between them
- composition: vertical composition of the natural transformations

Exercises:

- View the category of \mathcal{S} -sorted sets, $\mathbf{Set}^{\mathcal{S}}$, as a functor category.
- Check whether $\mathbf{K}^{\mathbf{K}'}$ is (finitely) (co)complete whenever \mathbf{K} is so.
- Show how any functor $\mathbf{F}: \mathbf{K}'' \rightarrow \mathbf{K}'$ induces a functor $(\mathbf{F}; -): \mathbf{K}^{\mathbf{K}'} \rightarrow \mathbf{K}^{\mathbf{K}''}$, where for $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$, $(\mathbf{F}; -)(\mathbf{G}) = \mathbf{F}; \mathbf{G}: \mathbf{K}'' \rightarrow \mathbf{K}$, and for $\tau: \mathbf{G} \rightarrow \mathbf{G}' (: \mathbf{K}' \rightarrow \mathbf{K})$, $(\mathbf{F}; -)(\tau) = \mathbf{F} \cdot \tau: \mathbf{F}; \mathbf{G} \rightarrow \mathbf{F}; \mathbf{G}'$.
- Check if $(\mathbf{F}; -): \mathbf{K}^{\mathbf{K}'} \rightarrow \mathbf{K}^{\mathbf{K}''}$ is (finitely) (co)continuous, for any $\mathbf{F}: \mathbf{K}'' \rightarrow \mathbf{K}'$.

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

$$\begin{array}{ccc} X & & \mathcal{Y}(A)(X) \\ \downarrow h & & \uparrow \mathcal{Y}(A)(h) \\ Y & & \mathcal{Y}(A)(Y) \end{array}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$
 - for $h: X \rightarrow Y$ in \mathbf{K} , $\mathcal{Y}(A)(h) = (h; -): \mathbf{Hom}_{\mathbf{K}}(Y, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, A)$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

$$\begin{array}{ccc} X & & \mathcal{Y}(A)(X) \\ \downarrow h & & \uparrow \mathcal{Y}(A)(h) \\ Y & & \mathcal{Y}(A)(Y) \end{array}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$
 - for $h: X \rightarrow Y$ in \mathbf{K} , $\mathcal{Y}(A)(h) = (h; -): \mathbf{Hom}_{\mathbf{K}}(Y, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, A)$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.

Yoneda embedding

$$A \xrightarrow{f} B$$

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

$$\begin{array}{ccccc} X & & \mathcal{Y}(A)(X) & & \mathcal{Y}(B)(X) \\ & \downarrow h & \uparrow \mathcal{Y}(A)(h) & & \uparrow \mathcal{Y}(B)(h) \\ Y & & \mathcal{Y}(A)(Y) & & \mathcal{Y}(B)(Y) \end{array}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$
 - for $h: X \rightarrow Y$ in \mathbf{K} , $\mathcal{Y}(A)(h) = (h; -): \mathbf{Hom}_{\mathbf{K}}(Y, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, A)$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.
 - for $f: A \rightarrow B$, $\mathcal{Y}(f): \mathcal{Y}(A) \rightarrow \mathcal{Y}(B) (: \mathbf{K}^{op} \rightarrow \mathbf{Set})$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

$$\begin{array}{ccccc}
 & & A & \xrightarrow{f} & B \\
 & & \mathcal{Y}(A)(X) & \xrightarrow{\mathcal{Y}(f)_X} & \mathcal{Y}(B)(X) \\
 & \uparrow \mathcal{Y}(A)(h) & & & \uparrow \mathcal{Y}(B)(h) \\
 X & \xrightarrow{h} & Y & \xrightarrow{\mathcal{Y}(f)_Y} & \mathcal{Y}(B)(Y)
 \end{array}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$
 - for $h: X \rightarrow Y$ in \mathbf{K} , $\mathcal{Y}(A)(h) = (h; -): \mathbf{Hom}_{\mathbf{K}}(Y, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, A)$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.
 - for $f: A \rightarrow B$, $\mathcal{Y}(f): \mathcal{Y}(A) \rightarrow \mathcal{Y}(B) (: \mathbf{K}^{op} \rightarrow \mathbf{Set})$
 - for $X \in |\mathbf{K}^{op}|$, $\mathcal{Y}(f)_X: \mathcal{Y}(A)(X) \rightarrow \mathcal{Y}(B)(X)$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

$$\begin{array}{ccccc}
 & & A & \xrightarrow{f} & B \\
 & & \mathcal{Y}(A)(X) & \xrightarrow{\mathcal{Y}(f)_X} & \mathcal{Y}(B)(X) \\
 & \uparrow \mathcal{Y}(A)(h) & & & \uparrow \mathcal{Y}(B)(h) \\
 X & \xrightarrow{h} & Y & \xrightarrow{\mathcal{Y}(f)_Y} & \mathcal{Y}(B)(Y)
 \end{array}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
 - for $X \in |\mathbf{K}|$, $\mathcal{Y}(A)(X) = \mathbf{Hom}_{\mathbf{K}}(X, A)$
 - for $h: X \rightarrow Y$ in \mathbf{K} , $\mathcal{Y}(A)(h) = (h; -): \mathbf{Hom}_{\mathbf{K}}(Y, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, A)$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.
 - for $f: A \rightarrow B$, $\mathcal{Y}(f): \mathcal{Y}(A) \rightarrow \mathcal{Y}(B) (: \mathbf{K}^{op} \rightarrow \mathbf{Set})$
 - for $X \in |\mathbf{K}^{op}|$, $\mathcal{Y}(f)_X: \mathcal{Y}(A)(X) \rightarrow \mathcal{Y}(B)(X)$
 - naturality of $\mathcal{Y}(f): \mathcal{Y}(A) \rightarrow \mathcal{Y}(B)$: for $h: X \rightarrow Y$ in \mathbf{K} ,
 $\mathcal{Y}(A)(h); \mathcal{Y}(f)_X = (h; -); (-; f) = h; -; f = (-; f); (h; -) = \mathcal{Y}(f)_Y; \mathcal{Y}(B)(h)$

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.

Theorem: *The category of presheaves $\mathbf{Set}^{\mathbf{K}^{op}}$ is complete and cocomplete.*

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.

Theorem: *The category of presheaves $\mathbf{Set}^{\mathbf{K}^{op}}$ is complete and cocomplete.*

Theorem: *$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$ is full and faithful.*

Yoneda embedding

Given a locally small category \mathbf{K} , define

$$\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$$

- $\mathcal{Y}(A) = \mathbf{Hom}_{\mathbf{K}}(-, A): \mathbf{K}^{op} \rightarrow \mathbf{Set}$, for $A \in |\mathbf{K}|$
- $\mathcal{Y}(f)_X = (-; f): \mathbf{Hom}_{\mathbf{K}}(X, A) \rightarrow \mathbf{Hom}_{\mathbf{K}}(X, B)$, for $f: A \rightarrow B$ in \mathbf{K} , $X \in |\mathbf{K}^{op}|$.

Theorem: *The category of presheaves $\mathbf{Set}^{\mathbf{K}^{op}}$ is complete and cocomplete.*

Theorem: $\mathcal{Y}: \mathbf{K} \rightarrow \mathbf{Set}^{\mathbf{K}^{op}}$ *is full and faithful.*

$\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ *is full and faithful*

if for all $A, B \in |\mathbf{K}|$,

$\mathbf{F}: \mathbf{K}(A, B) \rightarrow \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$ *is a bijection*

Diagrams as functors

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

- for distinct diagrams D and D' of shape G , \mathbf{F}_D and $\mathbf{F}_{D'}$ are different

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

- for distinct diagrams D and D' of shape G , \mathbf{F}_D and $\mathbf{F}_{D'}$ are different
- all functors from $\mathbf{Path}(G)$ to \mathbf{K} are given by diagrams over G

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D: \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

- for distinct diagrams D and D' of shape G , \mathbf{F}_D and $\mathbf{F}_{D'}$ are different
- all functors from $\mathbf{Path}(G)$ to \mathbf{K} are given by diagrams over G

Diagram morphisms $\mu: D \rightarrow D'$ between diagrams of the same shape G are exactly natural transformations $\mu: \mathbf{F}_D \rightarrow \mathbf{F}_{D'}$.

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D: \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

- for distinct diagrams D and D' of shape G , \mathbf{F}_D and $\mathbf{F}_{D'}$ are different
- all functors from $\mathbf{Path}(G)$ to \mathbf{K} are given by diagrams over G

Diagram morphisms $\mu: D \rightarrow D'$ between diagrams of the same shape G are exactly natural transformations $\mu: \mathbf{F}_D \rightarrow \mathbf{F}_{D'}$.

$$\mathbf{Diag}_{\mathbf{K}}^G \cong \mathbf{K}^{\mathbf{Path}(G)}$$

Diagrams as functors

Each diagram D over graph G in category \mathbf{K} yields a functor $\mathbf{F}_D : \mathbf{Path}(G) \rightarrow \mathbf{K}$ given by:

- $\mathbf{F}_D(n) = D_n$, for all nodes $n \in |G|_{nodes}$
- $\mathbf{F}_D(n_0 e_1 n_1 \dots n_{k-1} e_k n_k) = D_{e_1}; \dots; D_{e_k}$, for paths $n_0 e_1 n_1 \dots n_{k-1} e_k n_k$ in G

Moreover:

- for distinct diagrams D and D' of shape G , \mathbf{F}_D and $\mathbf{F}_{D'}$ are different
- all functors from $\mathbf{Path}(G)$ to \mathbf{K} are given by diagrams over G

Diagram morphisms $\mu : D \rightarrow D'$ between diagrams of the same shape G are exactly natural transformations $\mu : \mathbf{F}_D \rightarrow \mathbf{F}_{D'}$.

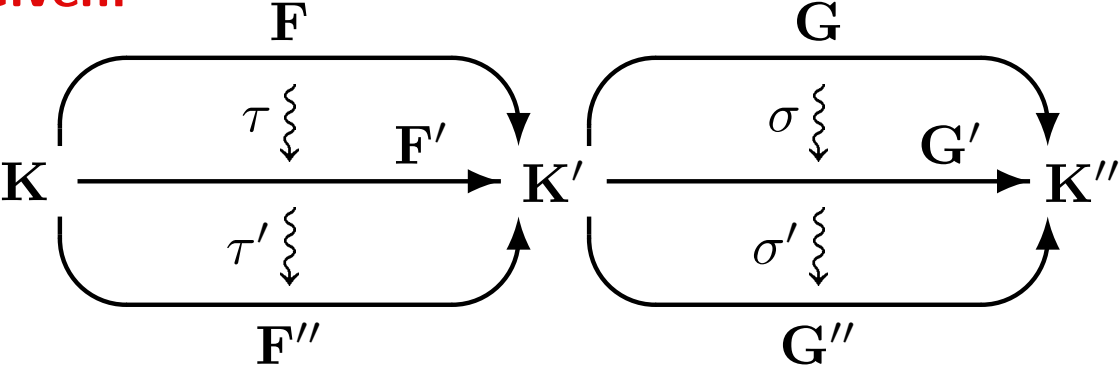
$$\mathbf{Diag}_{\mathbf{K}}^G \cong \mathbf{K}^{\mathbf{Path}(G)}$$

Diagrams are functors from small (shape) categories

Double law

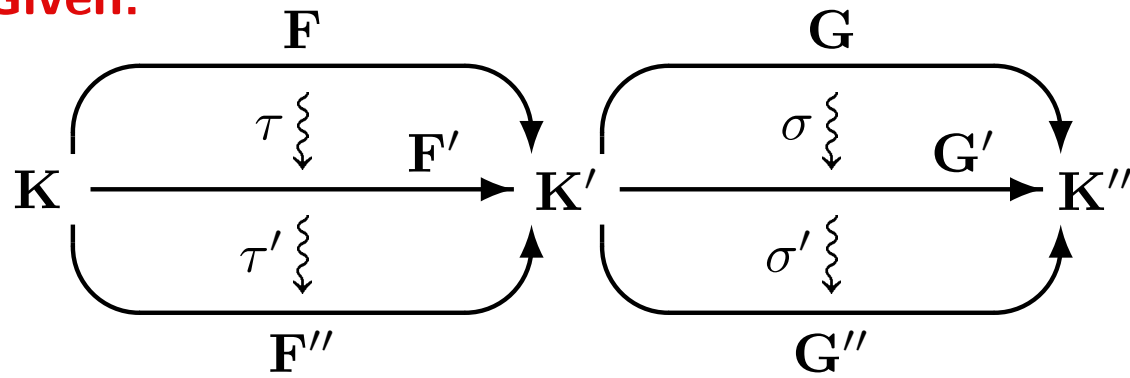
Double law

Given:

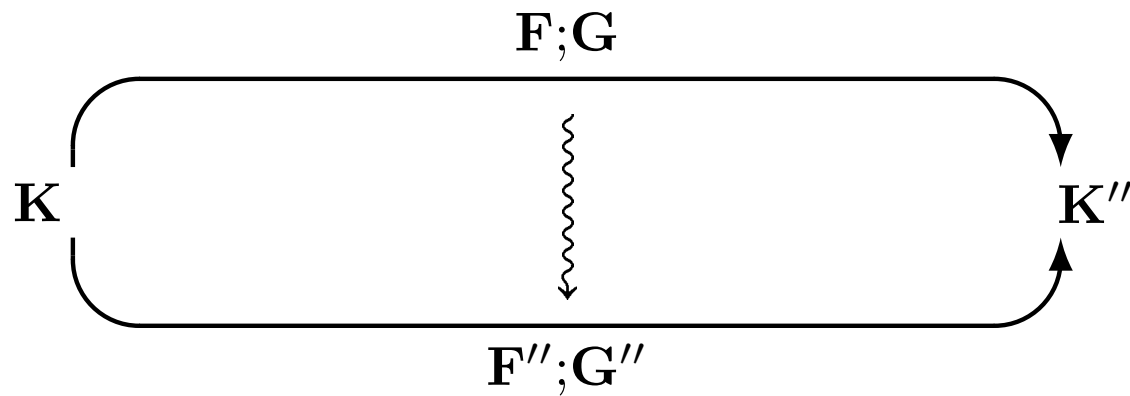


Double law

Given:

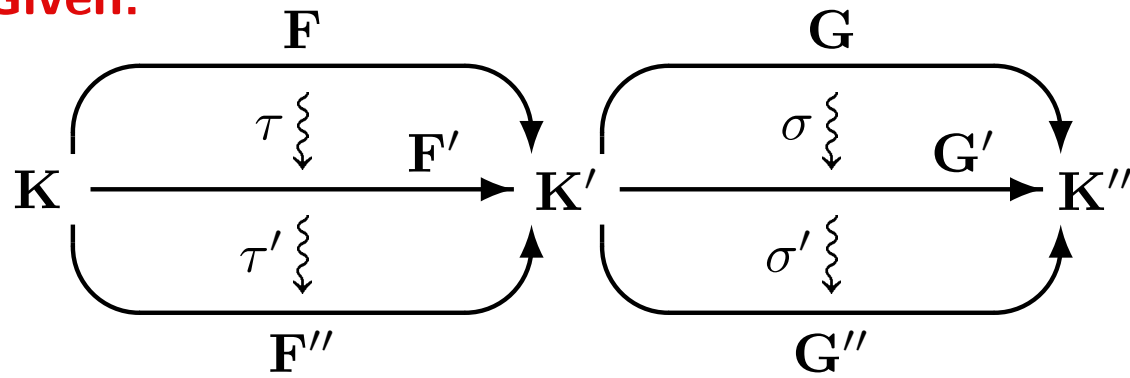


then:

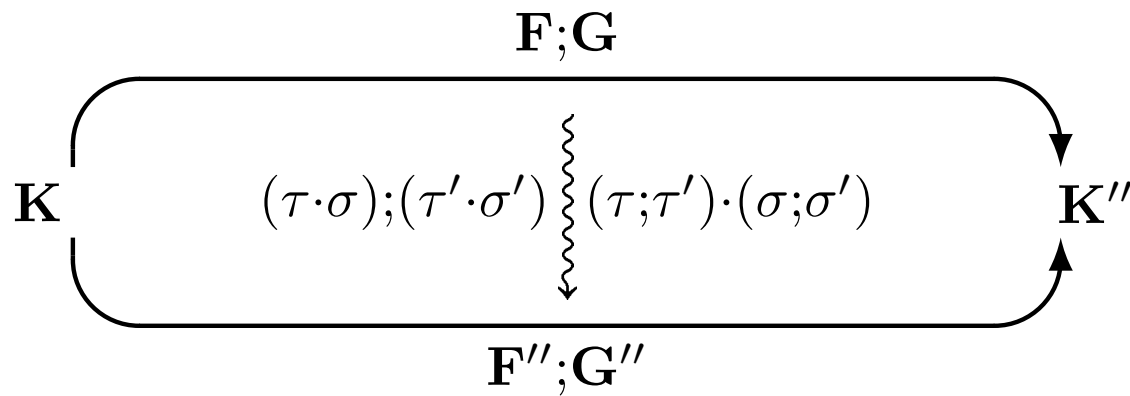


Double law

Given:

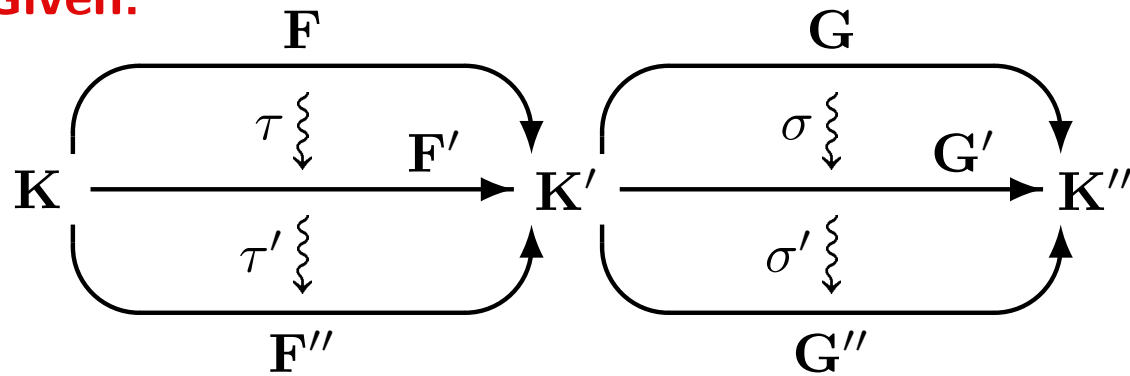


then:



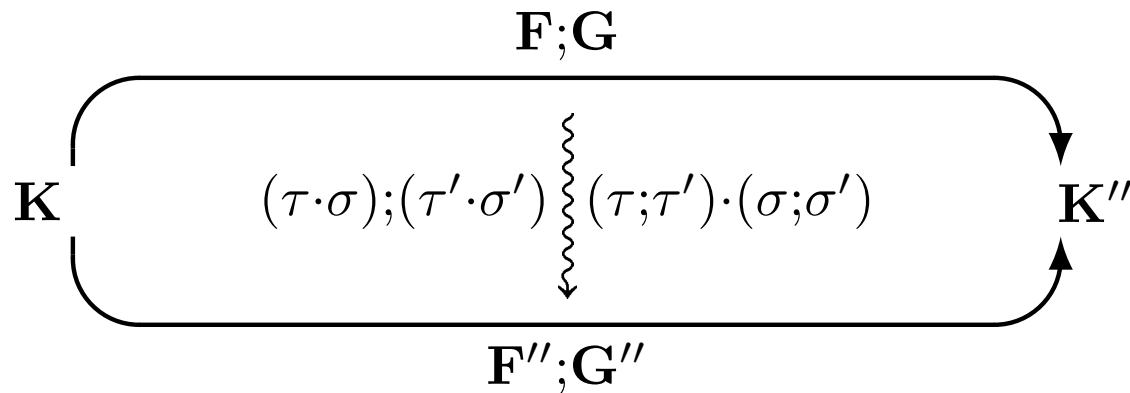
Double law

Given:



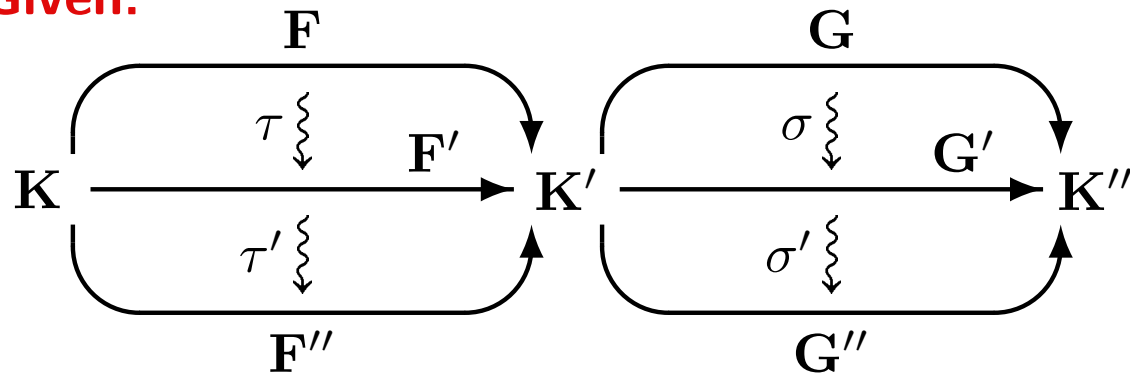
then:

$$(\tau \cdot \sigma); (\tau' \cdot \sigma') = (\tau; \tau') \cdot (\sigma; \sigma')$$



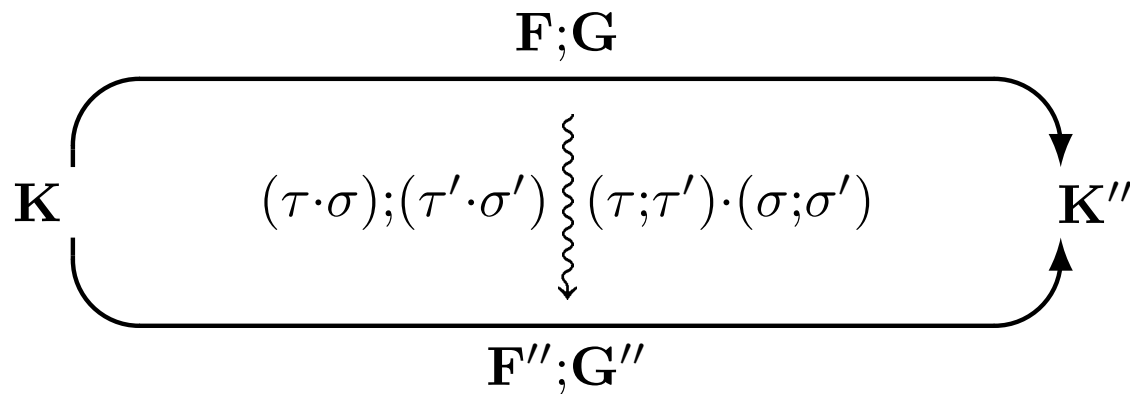
Double law

Given:



then:

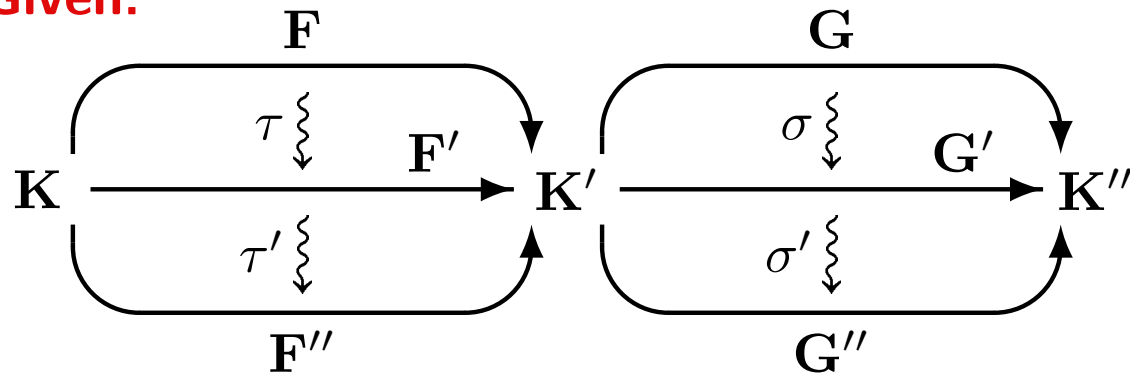
$$(\tau \cdot \sigma); (\tau' \cdot \sigma') = (\tau; \tau') \cdot (\sigma; \sigma')$$



This holds in **Cat**, which is a paradigmatic example of a two-category.

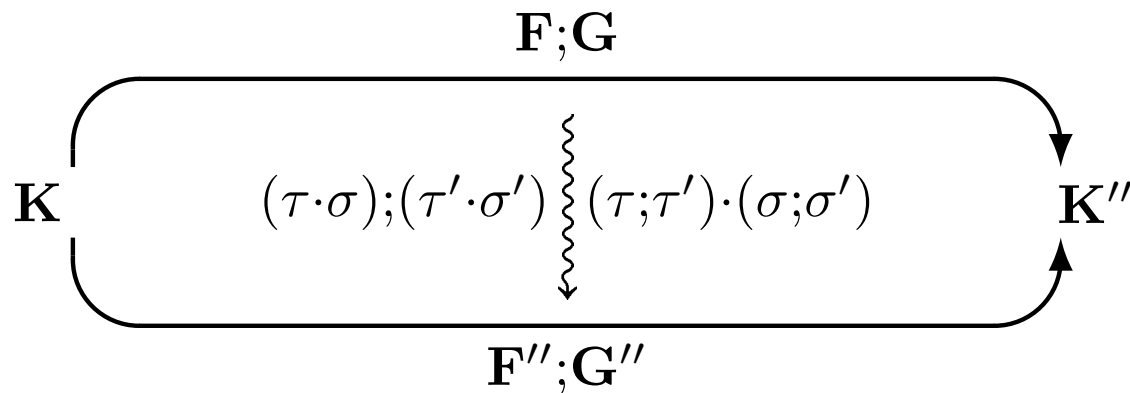
Double law

Given:



then:

$$(\tau \cdot \sigma); (\tau' \cdot \sigma') = (\tau; \tau') \cdot (\sigma; \sigma')$$

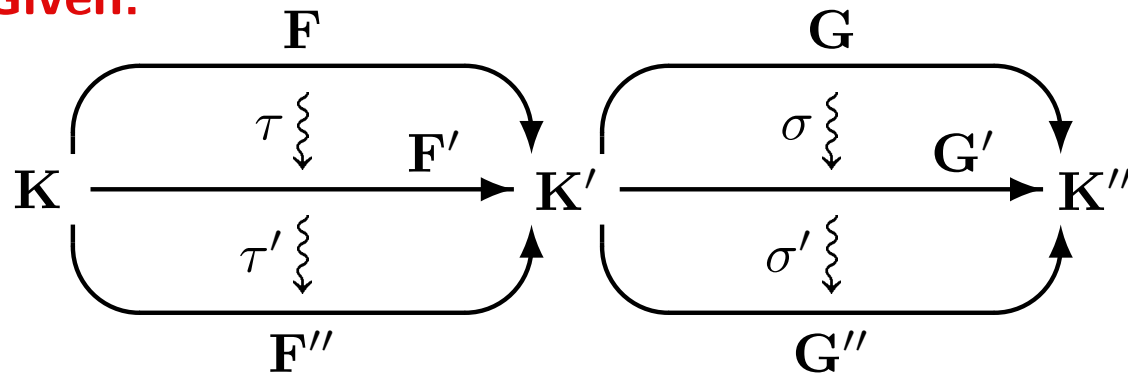


This holds in **Cat**, which is a paradigmatic example of a two-category.

A category **K** is a *two-category* when for all objects $A, B \in |\mathbf{K}|$, $\mathbf{K}(A, B)$ is again a category, with *1-morphisms* (the usual **K**-morphisms) as objects and *2-morphisms* between them.

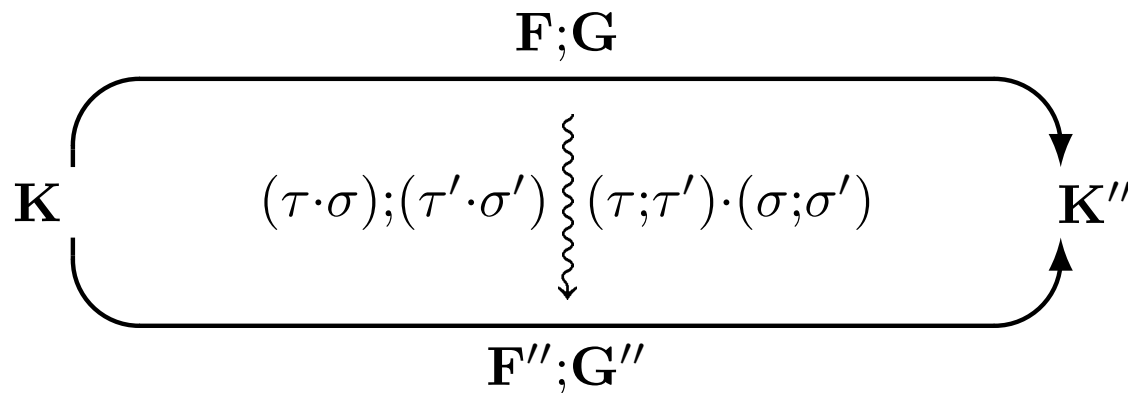
Double law

Given:



then:

$$(\tau \cdot \sigma); (\tau' \cdot \sigma') = (\tau; \tau') \cdot (\sigma; \sigma')$$



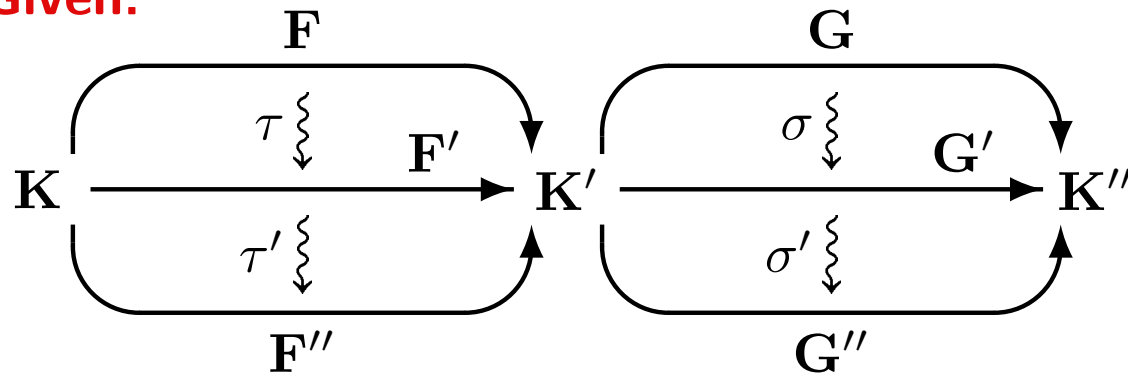
This holds in **Cat**, which is a paradigmatic example of a two-category.

A category K is a *two-category* when for all objects $A, B \in |K|$, $K(A, B)$ is again a category, with *1-morphisms* (the usual K -morphisms) as objects and *2-morphisms* between them.

Those 2-morphisms compose vertically (in the categories $K(A, B)$) and horizontally, subject to the double law as stated here.

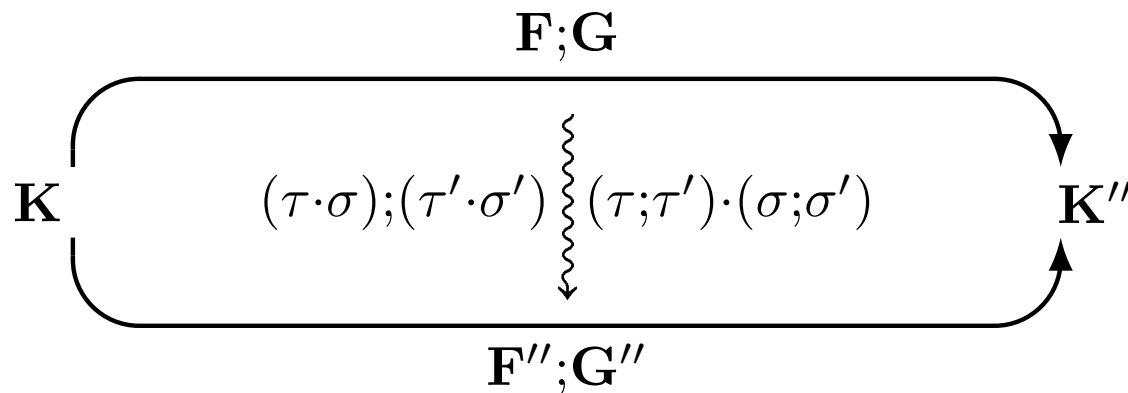
Double law

Given:



then:

$$(\tau \cdot \sigma); (\tau' \cdot \sigma') = (\tau; \tau') \cdot (\sigma; \sigma')$$



This holds in **Cat**, which is a paradigmatic example of a two-category.

A category K is a *two-category* when for all objects $A, B \in |K|$, $K(A, B)$ is again a category, with *1-morphisms* (the usual K -morphisms) as objects and *2-morphisms* between them.

Those 2-morphisms compose vertically (in the categories $K(A, B)$) and horizontally, subject to the double law as stated here.

In two-category **Cat**, we have $\mathbf{Cat}(K', K) = K^{K'}$.

Equivalence of categories

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.
- Two categories \mathbf{K} and \mathbf{K}' are *equivalent* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural isomorphisms $\eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$ and $\epsilon: \mathbf{G};\mathbf{F} \rightarrow \text{Id}_{\mathbf{K}'}$.

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.
- Two categories \mathbf{K} and \mathbf{K}' are *equivalent* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural isomorphisms $\eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$ and $\epsilon: \mathbf{G};\mathbf{F} \rightarrow \text{Id}_{\mathbf{K}'}$.
- A category is *skeletal* if any two isomorphic objects are identical.

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.
- Two categories \mathbf{K} and \mathbf{K}' are *equivalent* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural isomorphisms $\eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$ and $\epsilon: \mathbf{G};\mathbf{F} \rightarrow \text{Id}_{\mathbf{K}'}$.
- A category is *skeletal* if any two isomorphic objects are identical.
- A *skeleton* of a category is any of its maximal skeletal subcategory.

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.
- Two categories \mathbf{K} and \mathbf{K}' are *equivalent* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural isomorphisms $\eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$ and $\epsilon: \mathbf{G};\mathbf{F} \rightarrow \text{Id}_{\mathbf{K}'}$.
- A category is *skeletal* if any two isomorphic objects are identical.
- A *skeleton* of a category is any of its maximal skeletal subcategory.

Theorem: *Two categories are equivalent iff they have isomorphic skeletons.*

Equivalence of categories

- Two categories \mathbf{K} and \mathbf{K}' are *isomorphic* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ such that $\mathbf{F};\mathbf{G} = \text{Id}_{\mathbf{K}}$ and $\mathbf{G};\mathbf{F} = \text{Id}_{\mathbf{K}'}$.
- Two categories \mathbf{K} and \mathbf{K}' are *equivalent* if there are functors $\mathbf{F}: \mathbf{K} \rightarrow \mathbf{K}'$ and $\mathbf{G}: \mathbf{K}' \rightarrow \mathbf{K}$ and natural isomorphisms $\eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$ and $\epsilon: \mathbf{G};\mathbf{F} \rightarrow \text{Id}_{\mathbf{K}'}$.
- A category is *skeletal* if any two isomorphic objects are identical.
- A *skeleton* of a category is any of its maximal skeletal subcategory.

Theorem: *Two categories are equivalent iff they have isomorphic skeletons.*

All “categorical” properties are preserved under equivalence of categories