Functors

and

natural transformations

Functors

and

natural transformations

functors	$\rightsquigarrow$	*category morphisms*
natural transformations	$\rightsquigarrow$	*functor morphisms*

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and
- for all $A, B \in |\mathbf{K}|$, a function $\mathbf{F}\colon \mathbf{K}(A,B) \to \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and
- for all $A, B \in |\mathbf{K}|$, a function $\mathbf{F}\colon \mathbf{K}(A, B) \to \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$

We really should differentiate between various components of F

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and
- for all $A, B \in |\mathbf{K}|$, a function $\mathbf{F}\colon \mathbf{K}(A, B) \to \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$

such that:

Make explicit categories in which we work at various places here

We really should differentiate between various components of F

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and
- for all $A, B \in |\mathbf{K}|$, a function $\mathbf{F}\colon \mathbf{K}(A,B) \to \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$

such that: Make explicit categories in which we work at various places here

- $\mathbf{F}$ preserves identities, i.e.,

$$\mathbf{F}(id_A) = id_{\mathbf{F}(A)}$$

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

We really should differentiate between various components of F

Functors

A *functor* $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ from a category $\mathbf{K}$ to a category $\mathbf{K}'$ consists of:

- a function $\mathbf{F}\colon |\mathbf{K}| \to |\mathbf{K}'|$, and
- for all $A, B \in |\mathbf{K}|$, a function $\mathbf{F}\colon \mathbf{K}(A,B) \to \mathbf{K}'(\mathbf{F}(A), \mathbf{F}(B))$

such that:

Make explicit categories in which we work at various places here

- $\mathbf{F}$ preserves identities, i.e.,

$$\mathbf{F}(id_A) = id_{\mathbf{F}(A)}$$

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

- $\mathbf{F}$ preserves composition, i.e.,

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

  for all $f\colon A \to B$ and $g\colon B \to C$ in $\mathbf{K}$.

We really should differentiate between various components of F

Examples

Examples

- *identity functors*: $\mathbf{Id_K} : \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$

Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I_{K \hookrightarrow K'}}\colon \mathbf{K} \to \mathbf{K'}$, for any subcategory $\mathbf{K}$ of $\mathbf{K'}$

Examples

- *identity functors*: $\mathbf{Id}_{\mathbf{K}} : \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'} : \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A : \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$

Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by

$$\mathbf{Set} \xrightarrow{\mathbf{P}} \mathbf{Set}$$

Examples

- *identity functors*: $\mathbf{Id_K} \colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'} \colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A \colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P} \colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$

$$\begin{array}{ccc} \mathbf{Set} & \xrightarrow{\mathbf{P}} & \mathbf{Set} \\ X & \longmapsto & 2^X \end{array}$$

Examples

- *identity functors*: $\mathbf{Id}_\mathbf{K} \colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'} \colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A \colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P} \colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$

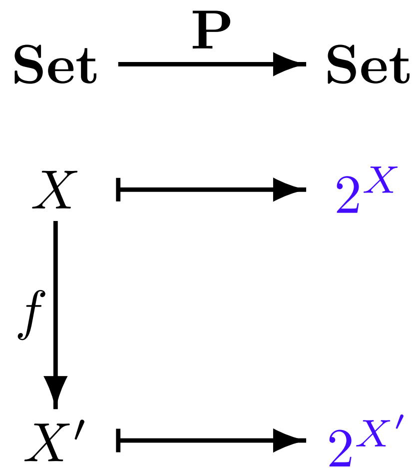
$$\mathbf{Set} \xrightarrow{\mathbf{P}} \mathbf{Set}$$

$$X \longmapsto 2^X$$

$$X' \longmapsto 2^{X'}$$

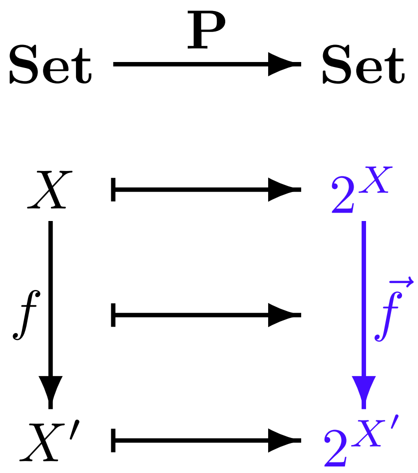
Examples

- *identity functors*: $\mathbf{Id}_\mathbf{K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$,



Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$



Examples

- *identity functors*: $\mathbf{Id}_{\mathbf{K}} \colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'} \colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A \colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P} \colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f) \colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f \colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$

$$\mathbf{Set}^{op} \xrightarrow{\mathbf{P}_{-1}} \mathbf{Set}$$

- *contravariant powerset functor*: $\mathbf{P}_{-1} \colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by

Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1}\colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$

$$\mathbf{Set}^{op} \xrightarrow{\mathbf{P}_{-1}} \mathbf{Set} \qquad X \longmapsto 2^X$$

Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I_{K \hookrightarrow K'}}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1}\colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$

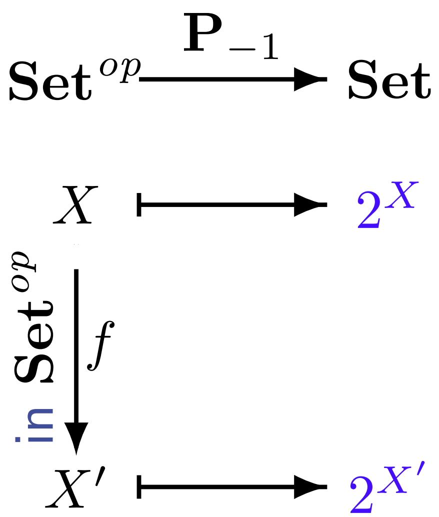
$$\mathbf{Set}^{op} \xrightarrow{\mathbf{P}_{-1}} \mathbf{Set}$$

$$X \longmapsto 2^X$$

$$X' \longmapsto 2^{X'}$$

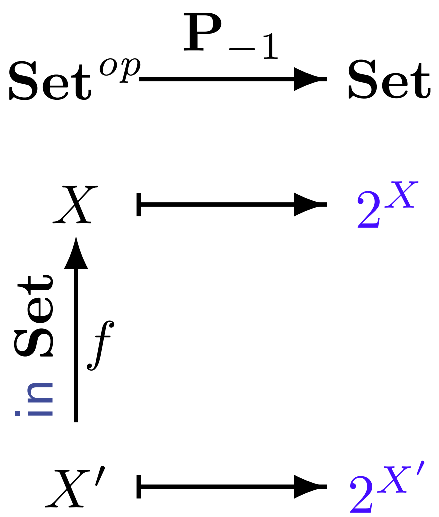
Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1}\colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$



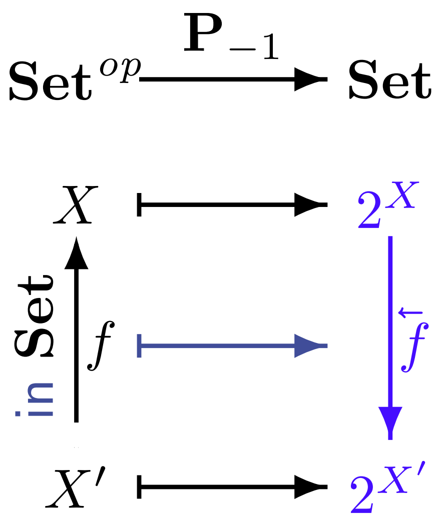
Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1}\colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}_{-1}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X' \to X$ in $\mathbf{Set}$,



Examples

- *identity functors*: $\mathbf{Id}_{\mathbf{K}} \colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K} \hookrightarrow \mathbf{K}'} \colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A \colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P} \colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f) \colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f \colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1} \colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}_{-1}(f) \colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f \colon X' \to X$ in $\mathbf{Set}$, $\mathbf{P}_{-1}(f)(Y) = \{x' \in X' \mid f(x') \in Y\}$ for all $Y \subseteq X$



Examples

- *identity functors*: $\mathbf{Id_K}\colon \mathbf{K} \to \mathbf{K}$, for any category $\mathbf{K}$
- *inclusions*: $\mathbf{I}_{\mathbf{K}\hookrightarrow\mathbf{K}'}\colon \mathbf{K} \to \mathbf{K}'$, for any subcategory $\mathbf{K}$ of $\mathbf{K}'$
- *constant functors*: $\mathbf{C}_A\colon \mathbf{K} \to \mathbf{K}'$, for any categories $\mathbf{K}, \mathbf{K}'$ and $A \in |\mathbf{K}'|$, with $\mathbf{C}_A(f) = id_A$ for all morphisms f in $\mathbf{K}$
- *powerset functor*: $\mathbf{P}\colon \mathbf{Set} \to \mathbf{Set}$ given by
  - $\mathbf{P}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X \to X'$ in $\mathbf{Set}$, $\mathbf{P}(f)(Y) = \{f(y) \mid y \in Y\}$ for all $Y \subseteq X$
- *contravariant powerset functor*: $\mathbf{P}_{-1}\colon \mathbf{Set}^{op} \to \mathbf{Set}$ given by
  - $\mathbf{P}_{-1}(X) = \{Y \mid Y \subseteq X\}$, for all $X \in |\mathbf{Set}|$
  - $\mathbf{P}_{-1}(f)\colon \mathbf{P}(X) \to \mathbf{P}(X')$ for all $f\colon X' \to X$ in $\mathbf{Set}$, $\mathbf{P}_{-1}(f)(Y) = \{x' \in X' \mid f(x') \in Y\}$ for all $Y \subseteq X$

Examples, cont'd.

- *projection functors*: $\pi_1 : \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 : \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:

$$\mathbf{Set} \xrightarrow{\mathbf{List}} \mathbf{Monoid}$$

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.

$$\mathbf{Set} \xrightarrow{\mathbf{List}} \mathbf{Monoid}$$

$$X \longmapsto \langle X^*, {}^\frown, \epsilon \rangle$$

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.

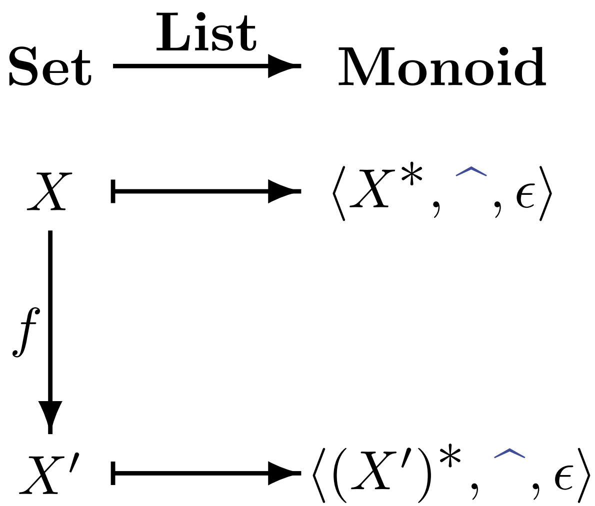
$$\mathbf{Set} \xrightarrow{\mathbf{List}} \mathbf{Monoid}$$

$$X \longmapsto \langle X^*, {}^\frown, \epsilon \rangle$$

$$X' \longmapsto \langle (X')^*, {}^\frown, \epsilon \rangle$$

Examples, cont'd.

- *projection functors*: $\pi_1 : \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 : \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} : \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) : \mathbf{List}(X) \to \mathbf{List}(X')$ for $f : X \to X'$ in $\mathbf{Set}$,



Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, ^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, $^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) \colon \mathbf{List}(X) \to \mathbf{List}(X')$ for $f \colon X \to X'$ in $\mathbf{Set}$, $\mathbf{List}(f)(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$ for all $x_1, \ldots, x_n \in X$

$$\mathbf{Set} \xrightarrow{\mathbf{List}} \mathbf{Monoid}$$

$$\begin{array}{ccc}
X & \longmapsto & \langle X^*, ^\frown, \epsilon \rangle \\
{\scriptstyle f}\downarrow & \longmapsto & \downarrow{\scriptstyle f^*} \\
X' & \longmapsto & \langle (X')^*, ^\frown, \epsilon \rangle
\end{array}$$

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) \colon \mathbf{List}(X) \to \mathbf{List}(X')$ for $f \colon X \to X'$ in $\mathbf{Set}$, $\mathbf{List}(f)(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$ for all $x_1, \ldots, x_n \in X$
- *totalisation functor*: $\mathbf{Tot} \colon \mathbf{Pfn} \to \mathbf{Set}_*$, where $\mathbf{Set}_*$ is the subcategory of $\mathbf{Set}$ of sets with a distinguished element $*$ and $*$-preserving functions

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) \colon \mathbf{List}(X) \to \mathbf{List}(X')$ for $f \colon X \to X'$ in $\mathbf{Set}$,
    $\mathbf{List}(f)(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$ for all $x_1, \ldots, x_n \in X$
- *totalisation functor*: $\mathbf{Tot} \colon \mathbf{Pfn} \to \mathbf{Set}_*$, where $\mathbf{Set}_*$ is the subcategory of $\mathbf{Set}$ of sets with a distinguished element $*$ and $*$-preserving functions

Define $\mathbf{Set}_*$ as the category of algebras

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) \colon \mathbf{List}(X) \to \mathbf{List}(X')$ for $f \colon X \to X'$ in $\mathbf{Set}$,
    $\mathbf{List}(f)(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$ for all $x_1, \ldots, x_n \in X$
- *totalisation functor*: $\mathbf{Tot} \colon \mathbf{Pfn} \to \mathbf{Set}_*$, where $\mathbf{Set}_*$ is the subcategory of $\mathbf{Set}$ of sets with a distinguished element $*$ and $*$-preserving functions
  - $\mathbf{Tot}(X) = X \uplus \{*\}$

Define $\mathbf{Set}_*$ as the category of algebras

Examples, cont'd.

- *projection functors*: $\pi_1 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}$, $\pi_2 \colon \mathbf{K} \times \mathbf{K}' \to \mathbf{K}'$
- *list functor*: $\mathbf{List} \colon \mathbf{Set} \to \mathbf{Monoid}$, where $\mathbf{Monoid}$ is the category of monoids (as objects) with monoid homomorphisms as morphisms:
  - $\mathbf{List}(X) = \langle X^*, {}^\frown, \epsilon \rangle$, for all $X \in |\mathbf{Set}|$, where X^* is the set of all finite lists of elements from X, ${}^\frown$ is the list concatenation, and ϵ is the empty list.
  - $\mathbf{List}(f) \colon \mathbf{List}(X) \to \mathbf{List}(X')$ for $f \colon X \to X'$ in $\mathbf{Set}$,
    $\mathbf{List}(f)(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$ for all $x_1, \ldots, x_n \in X$
- *totalisation functor*: $\mathbf{Tot} \colon \mathbf{Pfn} \to \mathbf{Set}_*$, where $\mathbf{Set}_*$ is the subcategory of $\mathbf{Set}$ of sets with a distinguished element $*$ and $*$-preserving functions
  - $\mathbf{Tot}(X) = X \uplus \{*\}$
  - $\mathbf{Tot}(f)(x) = \begin{cases} f(x) & \text{if it is defined} \\ * & \text{otherwise} \end{cases}$

Define $\mathbf{Set}_*$ as the category of algebras

Examples, cont'd.

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma \colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma \colon \Sigma \to \Sigma'$, as defined earlier

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_\big|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$
  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$
  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$
  - $\mathbf{T}_\Sigma(f) = f^\#\colon T_\Sigma(X) \to T_\Sigma(X')$ for all functions $f\colon X \to X'$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$

  Generalise to many-sorted signatures

  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$
  - $\mathbf{T}_\Sigma(f) = f^\#\colon T_\Sigma(X) \to T_\Sigma(X')$ for all functions $f\colon X \to X'$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$

  Generalise to many-sorted signatures

  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$
  - $\mathbf{T}_\Sigma(f) = f^\#\colon T_\Sigma(X) \to T_\Sigma(X')$ for all functions $f\colon X \to X'$
- *diagonal functors*: $\Delta^G_{\mathbf{K}}\colon \mathbf{K} \to \mathbf{Diag}^G_{\mathbf{K}}$ for any graph G with nodes $N = |G|_{nodes}$ and edges $E = |G|_{edges}$, and category $\mathbf{K}$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega\rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$

  Generalise to many-sorted signatures

  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$
  - $\mathbf{T}_\Sigma(f) = f^\#\colon T_\Sigma(X) \to T_\Sigma(X')$ for all functions $f\colon X \to X'$
- *diagonal functors*: $\Delta^G_{\mathbf{K}}\colon \mathbf{K} \to \mathbf{Diag}^G_{\mathbf{K}}$ for any graph G with nodes $N = |G|_{nodes}$ and edges $E = |G|_{edges}$, and category $\mathbf{K}$
  - $\Delta^G_{\mathbf{K}}(A) = D^A$, where D^A is the "constant" diagram, with $D^A_n = A$ for all $n \in N$ and $D^A_e = id_A$ for all $e \in E$

Examples, cont'd.

- *carrier set functors*: $|\_|\colon \mathbf{Alg}(\Sigma) \to \mathbf{Set}^S$, for any algebraic signature $\Sigma = \langle S, \Omega \rangle$, yielding the algebra carriers and homomorphisms as functions between them
- *reduct functors*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, as defined earlier
- *term algebra functors*: $\mathbf{T}_\Sigma\colon \mathbf{Set} \to \mathbf{Alg}(\Sigma)$ for all (single-sorted) algebraic signatures $\Sigma \in |\mathbf{AlgSig}|$

  Generalise to many-sorted signatures

  - $\mathbf{T}_\Sigma(X) = T_\Sigma(X)$ for all $X \in |\mathbf{Set}|$
  - $\mathbf{T}_\Sigma(f) = f^\#\colon T_\Sigma(X) \to T_\Sigma(X')$ for all functions $f\colon X \to X'$
- *diagonal functors*: $\Delta^G_\mathbf{K}\colon \mathbf{K} \to \mathbf{Diag}^G_\mathbf{K}$ for any graph G with nodes $N = |G|_{nodes}$ and edges $E = |G|_{edges}$, and category $\mathbf{K}$
  - $\Delta^G_\mathbf{K}(A) = D^A$, where D^A is the "constant" diagram, with $D^A_n = A$ for all $n \in N$ and $D^A_e = id_A$ for all $e \in E$
  - $\Delta^G_\mathbf{K}(f) = \mu^f\colon D^A \to D^B$, for all $f\colon A \to B$, where $\mu^f_n = f$ for all $n \in N$

Hom-functors

Hom-functors

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

$$\boxed{\mathbf{Hom}_{\mathbf{K}} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

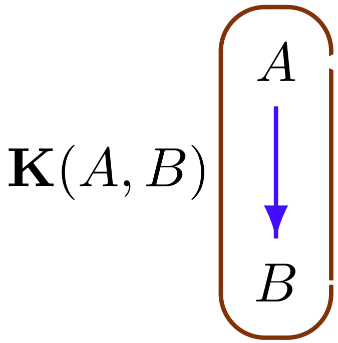
Hom-functors

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

$$\mathbf{Hom_K} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom_K}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$



Hom-functors

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

$$\mathbf{Hom_K} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom_K}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$

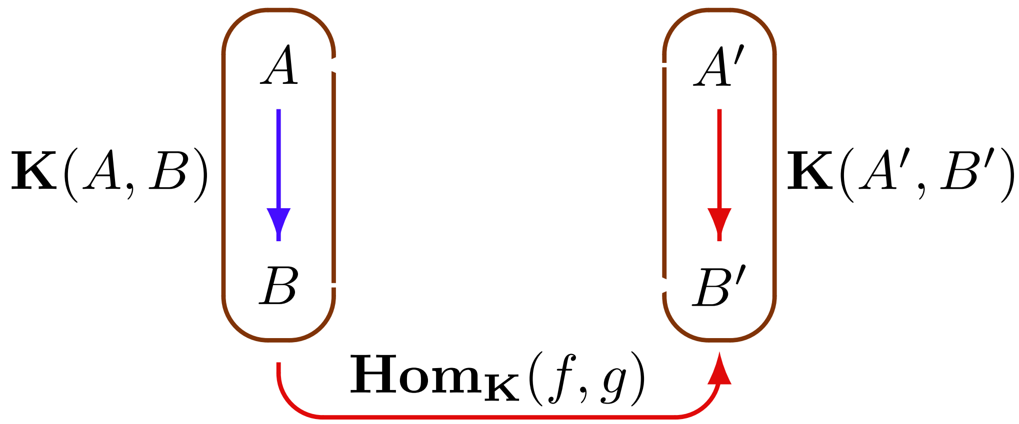
Hom-functors

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

$$\mathbf{Hom}_{\mathbf{K}} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom}_{\mathbf{K}}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$
- $\mathbf{Hom}_{\mathbf{K}}(\langle f, g\rangle) \colon \mathbf{K}(A, B) \to \mathbf{K}(A', B')$, for $\langle f, g\rangle \colon \langle A, B\rangle \to \langle A', B'\rangle$ in $\mathbf{K}^{op} \times \mathbf{K}$,



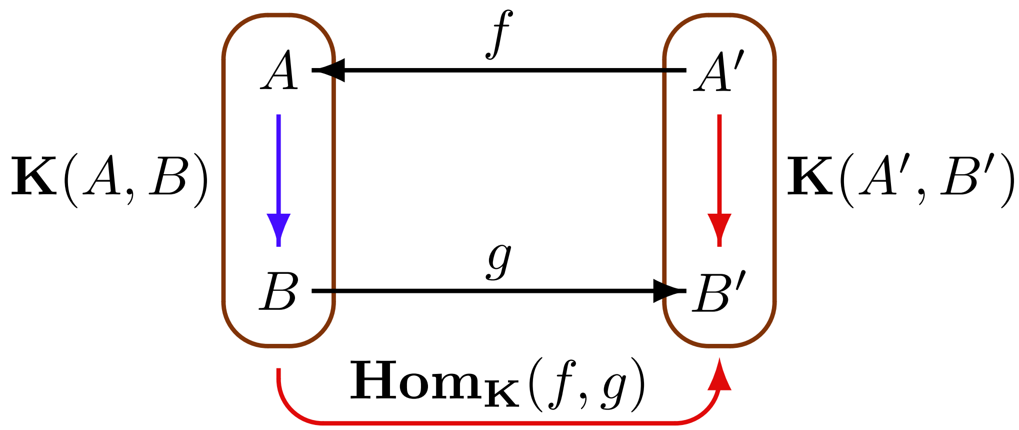
Hom-functors

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

$$\mathbf{Hom_K} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom_K}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$
- $\mathbf{Hom_K}(\langle f, g\rangle) \colon \mathbf{K}(A, B) \to \mathbf{K}(A', B')$, for $\langle f, g\rangle \colon \langle A, B\rangle \to \langle A', B'\rangle$ in $\mathbf{K}^{op} \times \mathbf{K}$, i.e., $f \colon A' \to A$ and $g \colon B \to B'$ in $\mathbf{K}$,



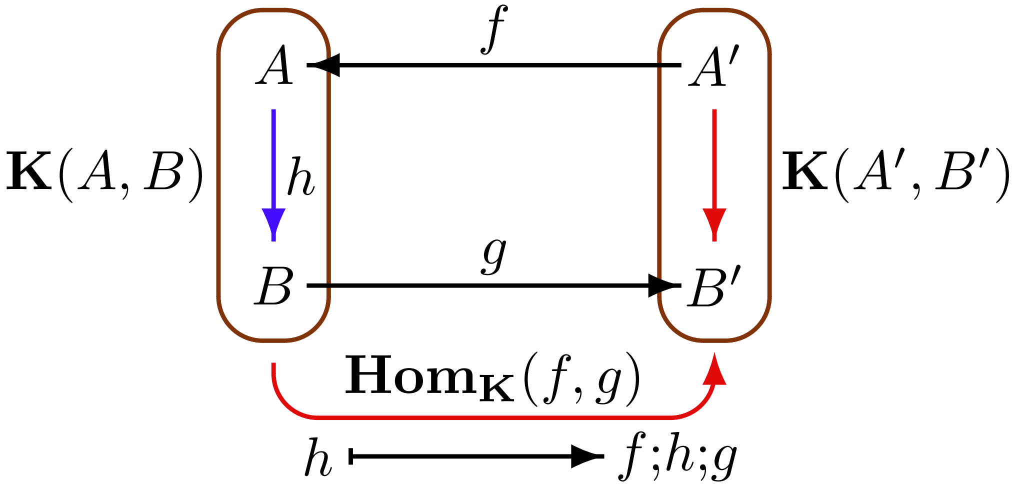
Hom-functors

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

$$\mathbf{Hom}_{\mathbf{K}} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom}_{\mathbf{K}}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$
- $\mathbf{Hom}_{\mathbf{K}}(\langle f, g\rangle) \colon \mathbf{K}(A, B) \to \mathbf{K}(A', B')$, for $\langle f, g\rangle \colon \langle A, B\rangle \to \langle A', B'\rangle$ in $\mathbf{K}^{op} \times \mathbf{K}$, i.e., $f \colon A' \to A$ and $g \colon B \to B'$ in $\mathbf{K}$, as a function given by $\mathbf{Hom}_{\mathbf{K}}(\langle f, g\rangle)(h) = f;h;g$.



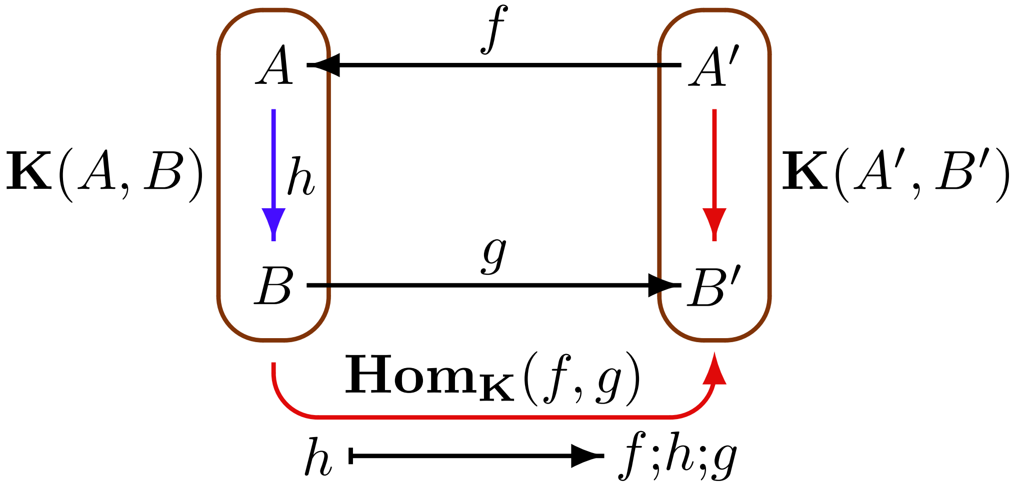
Hom-functors

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

$$\mathbf{Hom_K} \colon \mathbf{K}^{op} \times \mathbf{K} \to \mathbf{Set}$$

a binary *hom-functor*, contravariant on the first argument and covariant on the second argument, as follows:

- $\mathbf{Hom_K}(\langle A, B\rangle) = \mathbf{K}(A, B)$, for all $\langle A, B\rangle \in |\mathbf{K}^{op} \times \mathbf{K}|$, i.e., $A, B \in |\mathbf{K}|$
- $\mathbf{Hom_K}(\langle f, g\rangle) \colon \mathbf{K}(A, B) \to \mathbf{K}(A', B')$, for $\langle f, g\rangle \colon \langle A, B\rangle \to \langle A', B'\rangle$ in $\mathbf{K}^{op} \times \mathbf{K}$, i.e., $f \colon A' \to A$ and $g \colon B \to B'$ in $\mathbf{K}$, as a function given by $\mathbf{Hom_K}(\langle f, g\rangle)(h) = f;h;g$.



Also: $\mathbf{Hom_K}(A, \_) \colon \mathbf{K} \to \mathbf{Set}$
$\mathbf{Hom_K}(\_, B) \colon \mathbf{K}^{op} \to \mathbf{Set}$

Functors preserve...

- Check whether functors preserve:

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms

Functors preserve...

- Check whether functors preserve:
  - monomorphisms

$\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$

If $f\colon A \to B$ is mono in $\mathbf{K}$ then
$\mathbf{F}(f)\colon \mathbf{F}(A) \to \mathbf{F}(B)$ is mono in $\mathbf{K}'$??

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms

$\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$

If $f\colon A \to B$ is epi in $\mathbf{K}$ then
$\mathbf{F}(f)\colon \mathbf{F}(A) \to \mathbf{F}(B)$ is epi in $\mathbf{K}'$??

Functors preserve...

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions

$$\mathbf{F} \colon \mathbf{K} \to \mathbf{K}'$$

If $f \colon A \to B$ is a retraction in $\mathbf{K}$ then $\mathbf{F}(f) \colon \mathbf{F}(A) \to \mathbf{F}(B)$ is a retraction in $\mathbf{K}'$??

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms

$$\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$$

If $f\colon A \to B$ is iso in $\mathbf{K}$ then

$\mathbf{F}(f)\colon \mathbf{F}(A) \to \mathbf{F}(B)$ is iso in $\mathbf{K}'$??

Functors preserve...

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones

$$\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$$

If $\alpha\colon X \to D$ is a cone on diagram D in $\mathbf{K}$ then $\mathbf{F}(\alpha)\colon \mathbf{F}(X) \to \mathbf{F}(D)$ is a cone on diagram $\mathbf{F}(D)$ in $\mathbf{K}'$??

BTW:

- $\mathbf{F}(D)$ has the same shape as D, i.e. $\mathcal{G}(\mathbf{F}(D)) = \mathcal{G}(D)$ (with nodes N and edges E)
  - $(\mathbf{F}(D))_n = \mathbf{F}(D_n)$ for $n \in N$
  - $(\mathbf{F}(D))_e = \mathbf{F}(D_e)$ for $e \in E$
- $\mathbf{F}(\alpha) = \langle \mathbf{F}(\alpha_n)\colon \mathbf{F}(X) \to (\mathbf{F}(D))_n \rangle_{n \in N}$

Functors preserve...

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits

$$\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$$

If $\alpha\colon X \to D$ is a limit of diagram D in $\mathbf{K}$ then $\mathbf{F}(\alpha)\colon \mathbf{F}(X) \to \mathbf{F}(D)$ is a limit of diagram $\mathbf{F}(D)$ in $\mathbf{K}'$??

Functors preserve...

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits
  - ...

$$\mathbf{F}: \mathbf{K} \to \mathbf{K}'$$

...

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits
  - . . .
- A functor is continuous if it preserves all existing limits.

Functors preserve...

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits
  - ...
- A functor is (finitely) continuous if it preserves all existing (finite) limits.

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits
  - . . .
- A functor is (finitely) continuous if it preserves all existing (finite) limits. Which of the above functors are (finitely) continuous?

Functors preserve. . .

- Check whether functors preserve:
  - monomorphisms
  - epimorphisms
  - (co)retractions
  - isomorphisms
  - (co)cones
  - (co)limits
  - . . .
- A functor is (finitely) continuous if it preserves all existing (finite) limits.
  Which of the above functors are (finitely) continuous?

Dualise!

Functors compose. . .

Functors compose. . .

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

Functors compose...

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

Functors compose...

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

$\mathbf{Cat}$, *the category of (sm)all categories*

Functors compose. . .

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

Cat, *the category of (sm)all categories*

– objects: (sm)all categories
– morphisms: functors between them
– composition: as above

Functors compose...

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

Cat, *the category of (sm)all categories*

– objects: (sm)all categories
– morphisms: functors between them
– composition: as above

Characterise isomorphisms in **Cat**

Functors compose...

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

Cat, *the category of (sm)all categories*

– objects: (sm)all categories
– morphisms: functors between them
– composition: as above

Characterise isomorphisms in **Cat**

Define products, terminal objects, equalisers and pullback in **Cat**

Functors compose...

Given two functors $\mathbf{F}\colon \mathbf{K} \to \mathbf{K}'$ and $\mathbf{G}\colon \mathbf{K}' \to \mathbf{K}''$, their *composition* $\mathbf{F};\mathbf{G}\colon \mathbf{K} \to \mathbf{K}''$ is defined as expected:

- $(\mathbf{F};\mathbf{G})(A) = \mathbf{G}(\mathbf{F}(A))$ for all $A \in |\mathbf{K}|$
- $(\mathbf{F};\mathbf{G})(f) = \mathbf{G}(\mathbf{F}(f))$ for all $f\colon A \to B$ in $\mathbf{K}$

Cat, *the category of (sm)all categories*

- objects: (sm)all categories
- morphisms: functors between them
- composition: as above

Characterise isomorphisms in **Cat**

Define products, terminal objects, equalisers and pullback in **Cat**

Try to define their duals

Comma categories

Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

$$\boxed{(\mathbf{F}, \mathbf{G})}$$

Comma categories

Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

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

– objects: triples $\langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$, where $A_1 \in |\mathbf{K1}|$, $A_2 \in |\mathbf{K2}|$, and $f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2)$ in $\mathbf{K}$

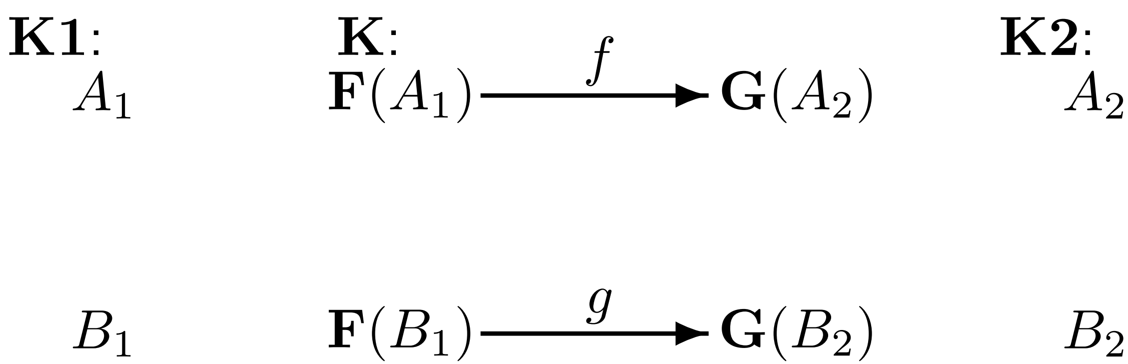
$\mathbf{K1}$: A_1 $\quad$ $\mathbf{K}$: $\mathbf{F}(A_1) \xrightarrow{f} \mathbf{G}(A_2)$ $\quad$ $\mathbf{K2}$: A_2

Comma categories

Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

$$\boxed{(\mathbf{F}, \mathbf{G})}$$

- objects: triples $\langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$, where $A_1 \in |\mathbf{K1}|$, $A_2 \in |\mathbf{K2}|$, and $f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2)$ in $\mathbf{K}$
- morphisms: a morphism in $(\mathbf{F}, \mathbf{G})$ is



Comma categories

Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

$$\boxed{(\mathbf{F}, \mathbf{G})}$$

- objects: triples $\langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$, where $A_1 \in |\mathbf{K1}|$, $A_2 \in |\mathbf{K2}|$, and $f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2)$ in $\mathbf{K}$
- morphisms: a morphism in $(\mathbf{F}, \mathbf{G})$ is any pair $\langle h_1, h_2\rangle\colon \langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle \to \langle B_1, g\colon \mathbf{F}(B_1) \to \mathbf{G}(B_2), B_2\rangle$, where $h_1\colon A_1 \to B_1$ in $\mathbf{K1}$, $h_2\colon A_2 \to B_2$ in $\mathbf{K2}$,

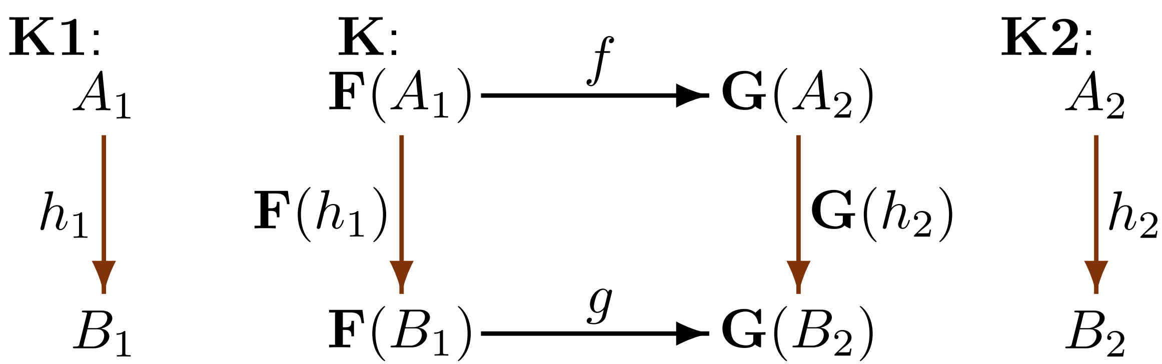
$$\begin{array}{ccccc}
\mathbf{K1}: & \mathbf{K}: & & & \mathbf{K2}: \\
A_1 & \mathbf{F}(A_1) & \xrightarrow{f} & \mathbf{G}(A_2) & A_2 \\
\downarrow h_1 & & & & \downarrow h_2 \\
B_1 & \mathbf{F}(B_1) & \xrightarrow{g} & \mathbf{G}(B_2) & B_2
\end{array}$$

Comma categories

Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

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

- objects: triples $\langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$, where $A_1 \in |\mathbf{K1}|$, $A_2 \in |\mathbf{K2}|$, and $f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2)$ in $\mathbf{K}$
- morphisms: a morphism in $(\mathbf{F}, \mathbf{G})$ is any pair $\langle h_1, h_2\rangle\colon \langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle \to \langle B_1, g\colon \mathbf{F}(B_1) \to \mathbf{G}(B_2), B_2\rangle$, where $h_1\colon A_1 \to B_1$ in $\mathbf{K1}$, $h_2\colon A_2 \to B_2$ in $\mathbf{K2}$, and $\mathbf{F}(h_1);g = f;\mathbf{G}(h_2)$ in $\mathbf{K}$.



Comma categories

– composition:

$$\mathbf{K1}: \qquad \mathbf{K}: \qquad \qquad \mathbf{K2}:$$

$$A_1 \qquad \mathbf{F}(A_1) \xrightarrow{f} \mathbf{G}(A_2) \qquad A_2$$

Comma categories

– composition:

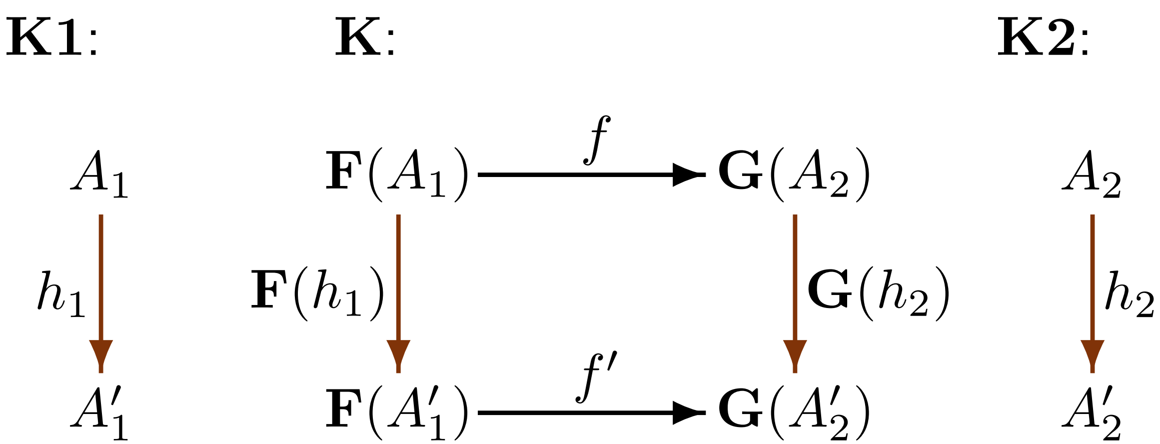
$\mathbf{K1}$: $\mathbf{K}$: $\mathbf{K2}$:

$$A_1 \qquad \mathbf{F}(A_1) \xrightarrow{f} \mathbf{G}(A_2) \qquad A_2$$

$$A_1' \qquad \mathbf{F}(A_1') \xrightarrow{f'} \mathbf{G}(A_2') \qquad A_2'$$

Comma categories

– composition:



Comma categories

– composition:

$$
\begin{array}{ccccc}
\mathbf{K1}: & \mathbf{K}: & & & \mathbf{K2}: \\
A_1 & \mathbf{F}(A_1) & \xrightarrow{\;f\;} & \mathbf{G}(A_2) & A_2 \\
h_1 \downarrow & \mathbf{F}(h_1) \downarrow & & \downarrow \mathbf{G}(h_2) & \downarrow h_2 \\
A_1' & \mathbf{F}(A_1') & \xrightarrow{\;f'\;} & \mathbf{G}(A_2') & A_2' \\
 & & & & \\
A_1'' & \mathbf{F}(A_1'') & \xrightarrow{\;f''\;} & \mathbf{G}(A_2'') & A_2''
\end{array}
$$

Comma categories

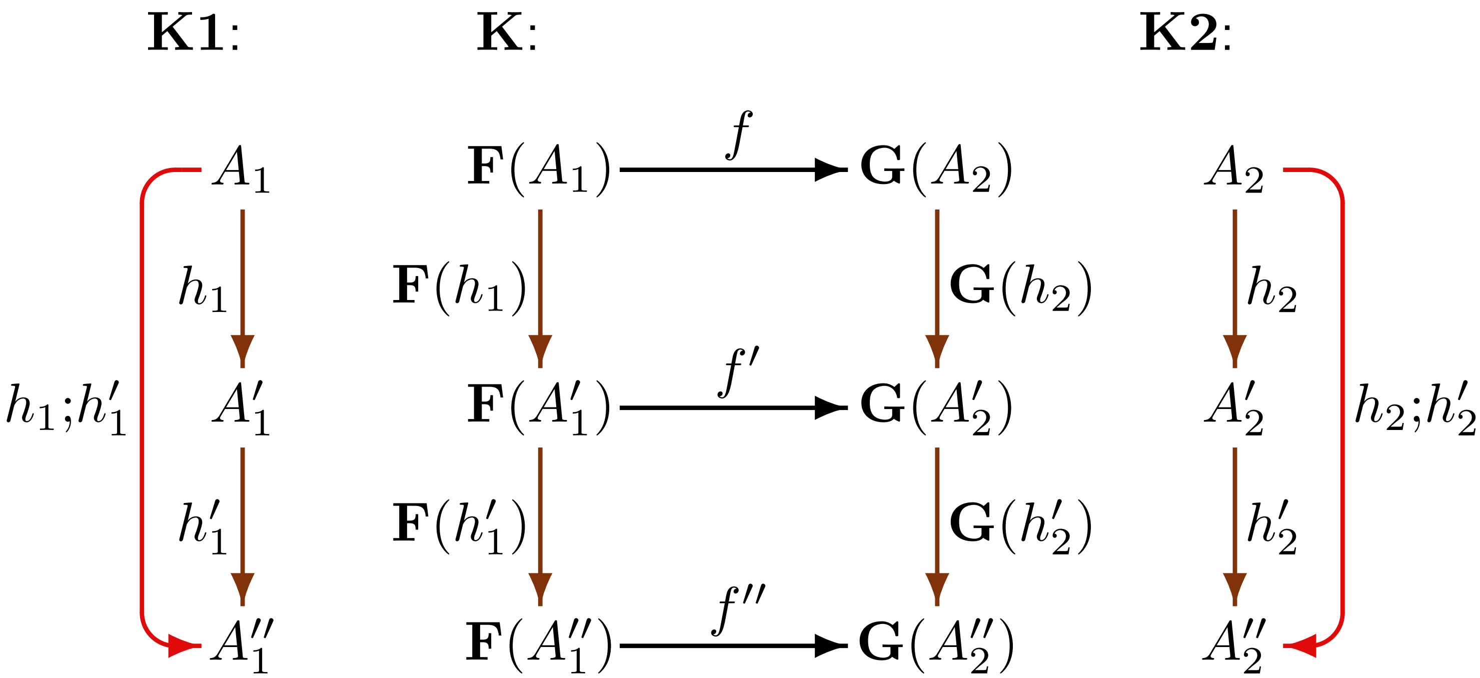
– composition:

$\mathbf{K1}$: $\mathbf{K}$: $\mathbf{K2}$:

$$
\begin{array}{ccccccc}
A_1 & \quad & \mathbf{F}(A_1) & \xrightarrow{f} & \mathbf{G}(A_2) & \quad & A_2 \\
{\scriptstyle h_1}\downarrow & & {\scriptstyle \mathbf{F}(h_1)}\downarrow & & \downarrow{\scriptstyle \mathbf{G}(h_2)} & & \downarrow{\scriptstyle h_2} \\
A_1' & & \mathbf{F}(A_1') & \xrightarrow{f'} & \mathbf{G}(A_2') & & A_2' \\
{\scriptstyle h_1'}\downarrow & & {\scriptstyle \mathbf{F}(h_1')}\downarrow & & \downarrow{\scriptstyle \mathbf{G}(h_2')} & & \downarrow{\scriptstyle h_2'} \\
A_1'' & & \mathbf{F}(A_1'') & \xrightarrow{f''} & \mathbf{G}(A_2'') & & A_2''
\end{array}
$$

Comma categories

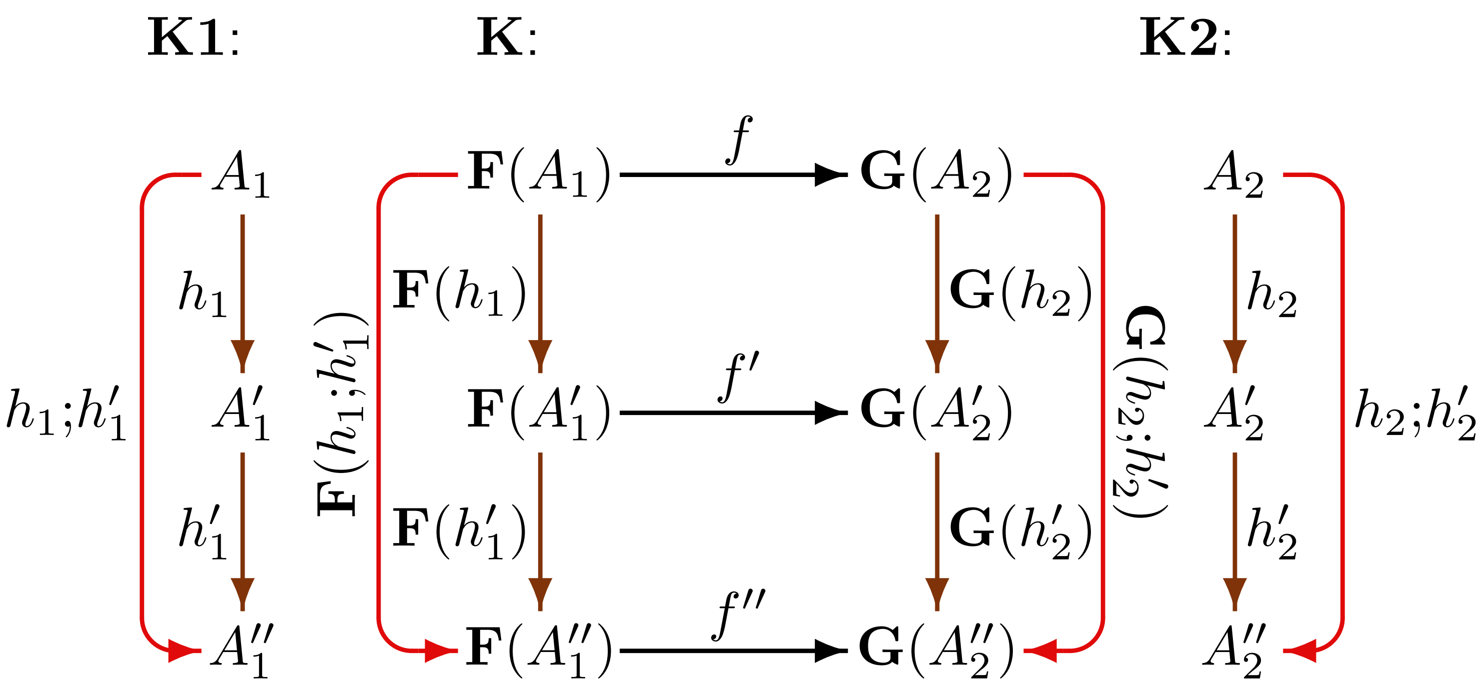
— composition: component-wise



$$\langle h_1, h_2\rangle;\langle h'_1, h'_2\rangle = \langle h_1;h'_1, h_2;h'_2\rangle$$

Comma categories

– composition: component-wise



$$\langle h_1, h_2\rangle;\langle h'_1, h'_2\rangle = \langle h_1;h'_1, h_2;h'_2\rangle$$

$$\mathbf{F}(h_1;h'_1);f'' = \mathbf{F}(h_1);\mathbf{F}(h'_1);f'' = \mathbf{F}(h_1);f';\mathbf{G}(h'_2) = f;\mathbf{G}(h_2);\mathbf{G}(h'_2) = f;\mathbf{G}(h_2;h'_2)$$

Comma categories

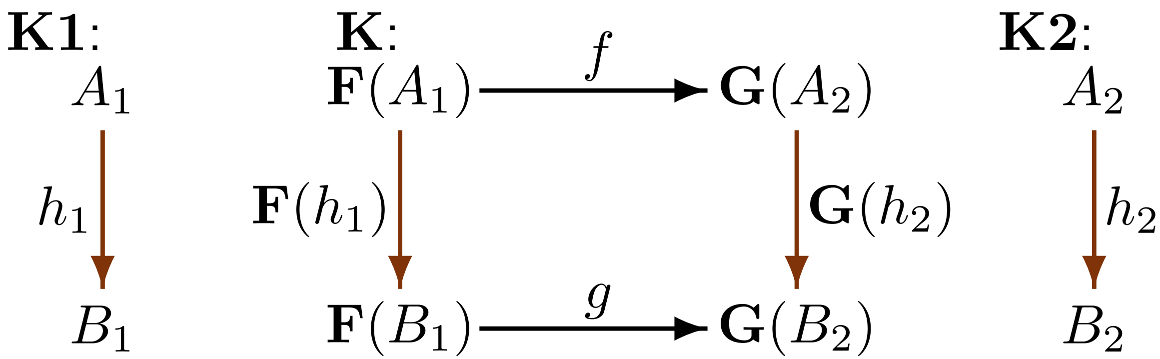
Given two functors with a common target, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$, define their *comma category*

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

– objects: triples $\langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$, where $A_1 \in |\mathbf{K1}|$, $A_2 \in |\mathbf{K2}|$, and $f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2)$ in $\mathbf{K}$

– morphisms: a morphism in $(\mathbf{F}, \mathbf{G})$ is any pair $\langle h_1, h_2\rangle\colon \langle A_1, f\colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle \to \langle B_1, g\colon \mathbf{F}(B_1) \to \mathbf{G}(B_2), B_2\rangle$, where $h_1\colon A_1 \to B_1$ in $\mathbf{K1}$, $h_2\colon A_2 \to B_2$ in $\mathbf{K2}$, and $\mathbf{F}(h_1);g = f;\mathbf{G}(h_2)$ in $\mathbf{K}$.

– composition: component-wise



Examples

Examples

- The category of graphs as a comma category:

$$\boxed{\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})}$$

where $\mathbf{CP} \colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$.

Examples

- The category of graphs as a comma category:

$$\boxed{\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})}$$

  where $\mathbf{CP}\colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$. Hint: write objects of this category as $\langle E, \langle source, target \rangle \colon E \to N \times N, N \rangle$.

Examples

- The category of graphs as a comma category:

$$\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})$$

where $\mathbf{CP}\colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$. Hint: write objects of this category as $\langle E, \langle source, target \rangle \colon E \to N \times N, N \rangle$.

- The category of algebraic signatures as a comma category:

$$\mathbf{AlgSig} = (\mathbf{Id}_{\mathbf{Set}}, (\_)^+)$$

where $(\_)^+ \colon \mathbf{Set} \to \mathbf{Set}$ is the non-empty list functor, i.e. $(X)^+$ is the set of all non-empty lists of elements from X, $(f)^+(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$.

Examples

- The category of graphs as a comma category:

$$\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})$$

  where $\mathbf{CP}\colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$. Hint: write objects of this category as $\langle E, \langle source, target \rangle \colon E \to N \times N, N \rangle$.

- The category of algebraic signatures as a comma category:

$$\mathbf{AlgSig} = (\mathbf{Id}_{\mathbf{Set}}, (\_)^{+})$$

  where $(\_)^{+}\colon \mathbf{Set} \to \mathbf{Set}$ is the non-empty list functor, i.e. $(X)^{+}$ is the set of all non-empty lists of elements from X, $(f)^{+}(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$. Hint: write objects of this category as $\langle \Omega, \langle arity, sort \rangle \colon \Omega \to S^{+}, S \rangle$.

Examples

- The category of graphs as a comma category:

$$\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})$$

where $\mathbf{CP}\colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$. Hint: write objects of this category as $\langle E, \langle source, target \rangle \colon E \to N \times N, N \rangle$.

- The category of algebraic signatures as a comma category:

$$\mathbf{AlgSig} = (\mathbf{Id}_{\mathbf{Set}}, (\_)^+)$$

where $(\_)^+\colon \mathbf{Set} \to \mathbf{Set}$ is the non-empty list functor, i.e. $(X)^+$ is the set of all non-empty lists of elements from X, $(f)^+(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$. Hint: write objects of this category as $\langle \Omega, \langle arity, sort \rangle \colon \Omega \to S^+, S \rangle$.

Define $\mathbf{K}^{\to}$, $\mathbf{K}{\downarrow}A$ as comma categories.

Examples

- The category of graphs as a comma category:

$$\mathbf{Graph} = (\mathbf{Id}_{\mathbf{Set}}, \mathbf{CP})$$

  where $\mathbf{CP}\colon \mathbf{Set} \to \mathbf{Set}$ is the (Cartesian) product functor, i.e. $\mathbf{CP}(X) = X \times X$ and $\mathbf{CP}(f)(\langle x, x' \rangle) = \langle f(x), f(x') \rangle$. Hint: write objects of this category as $\langle E, \langle source, target \rangle \colon E \to N \times N, N \rangle$.

- The category of algebraic signatures as a comma category:

$$\mathbf{AlgSig} = (\mathbf{Id}_{\mathbf{Set}}, (\_)^+)$$

  where $(\_)^+\colon \mathbf{Set} \to \mathbf{Set}$ is the non-empty list functor, i.e. $(X)^+$ is the set of all non-empty lists of elements from X, $(f)^+(\langle x_1, \ldots, x_n \rangle) = \langle f(x_1), \ldots, f(x_n) \rangle$. Hint: write objects of this category as $\langle \Omega, \langle arity, sort \rangle \colon \Omega \to S^+, S \rangle$.

Define $\mathbf{K}^{\to}$, $\mathbf{K}{\downarrow}A$ as comma categories. The same for $\mathbf{Alg}(\Sigma)$.

Cocompleteness of comma categories

Cocompleteness of comma categories

Theorem: *If* $\mathbf{K1}$ *and* $\mathbf{K2}$ *are (finitely) cocomplete categories,* $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ *is a (finitely) cocontinuous functor, and* $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$ *is a functor then the comma category* $(\mathbf{F}, \mathbf{G})$ *is (finitely) cocomplete.*

Cocompleteness of comma categories

Theorem: *If $\mathbf{K1}$ and $\mathbf{K2}$ are (finitely) cocomplete categories, $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ is a (finitely) cocontinuous functor, and $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$ is a functor then the comma category $(\mathbf{F}, \mathbf{G})$ is (finitely) cocomplete.*

Proof (idea):

Construct coproducts and coequalisers in $(\mathbf{F}, \mathbf{G})$, using the corresponding constructions in $\mathbf{K1}$ and $\mathbf{K2}$, and cocontinuity of $\mathbf{F}$.

Cocompleteness of comma categories

Theorem: *If* $\mathbf{K1}$ *and* $\mathbf{K2}$ *are (finitely) cocomplete categories,* $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ *is a (finitely) cocontinuous functor, and* $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$ *is a functor then the comma category* $(\mathbf{F}, \mathbf{G})$ *is (finitely) cocomplete.*

Proof (idea):

Construct coproducts and coequalisers in $(\mathbf{F}, \mathbf{G})$, using the corresponding constructions in $\mathbf{K1}$ and $\mathbf{K2}$, and cocontinuity of $\mathbf{F}$.

State and prove the dual fact,
concerning completeness of comma categories

Cocompleteness of comma categories

Theorem: *If* $\mathbf{K1}$ *and* $\mathbf{K2}$ *are (finitely) cocomplete categories,* $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ *is a (finitely) cocontinuous functor, and* $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$ *is a functor then the comma category* $(\mathbf{F}, \mathbf{G})$ *is (finitely) cocomplete.*

Proof (idea):

Construct coproducts and coequalisers in $(\mathbf{F}, \mathbf{G})$, using the corresponding constructions in $\mathbf{K1}$ and $\mathbf{K2}$, and cocontinuity of $\mathbf{F}$.

> *State and prove the dual fact,*
> *concerning completeness of comma categories*

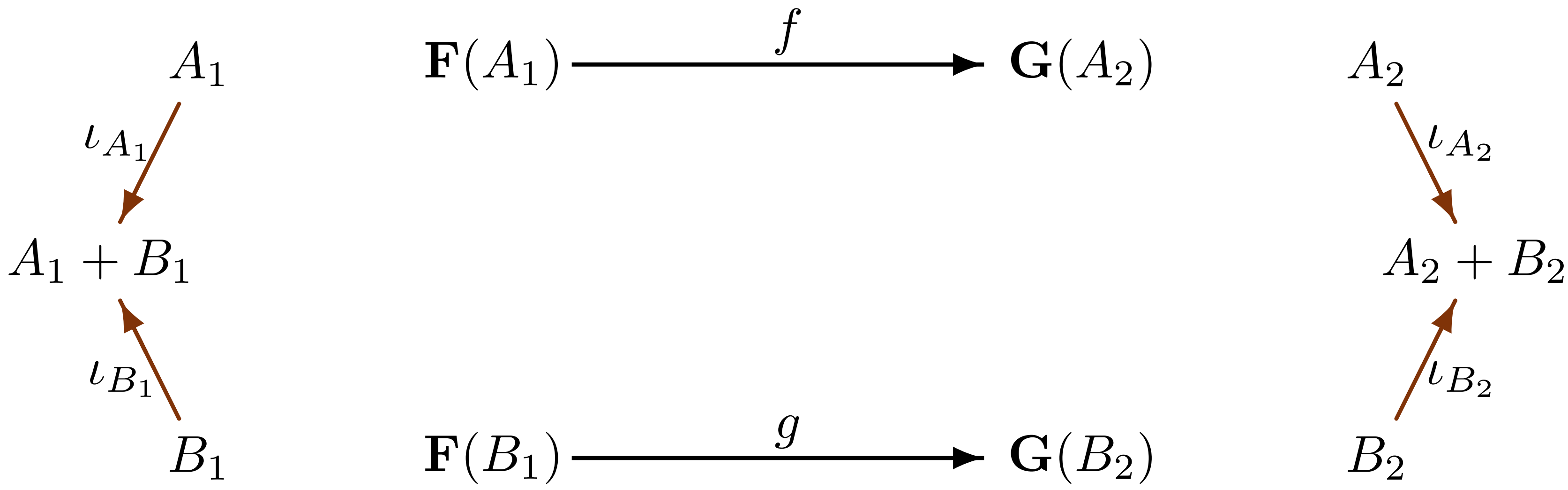
Theorem: *If* $\mathbf{K1}$ *and* $\mathbf{K2}$ *are (finitely) complete categories,* $\mathbf{F}\colon \mathbf{K1} \to \mathbf{K}$ *is a functor, and* $\mathbf{G}\colon \mathbf{K2} \to \mathbf{K}$ *is a (finitely) continuous functor then the comma category* $(\mathbf{F}, \mathbf{G})$ *is (finitely) complete.*

Coproducts:

$$A_1 \qquad \mathbf{F}(A_1) \xrightarrow{\;f\;} \mathbf{G}(A_2) \qquad A_2$$

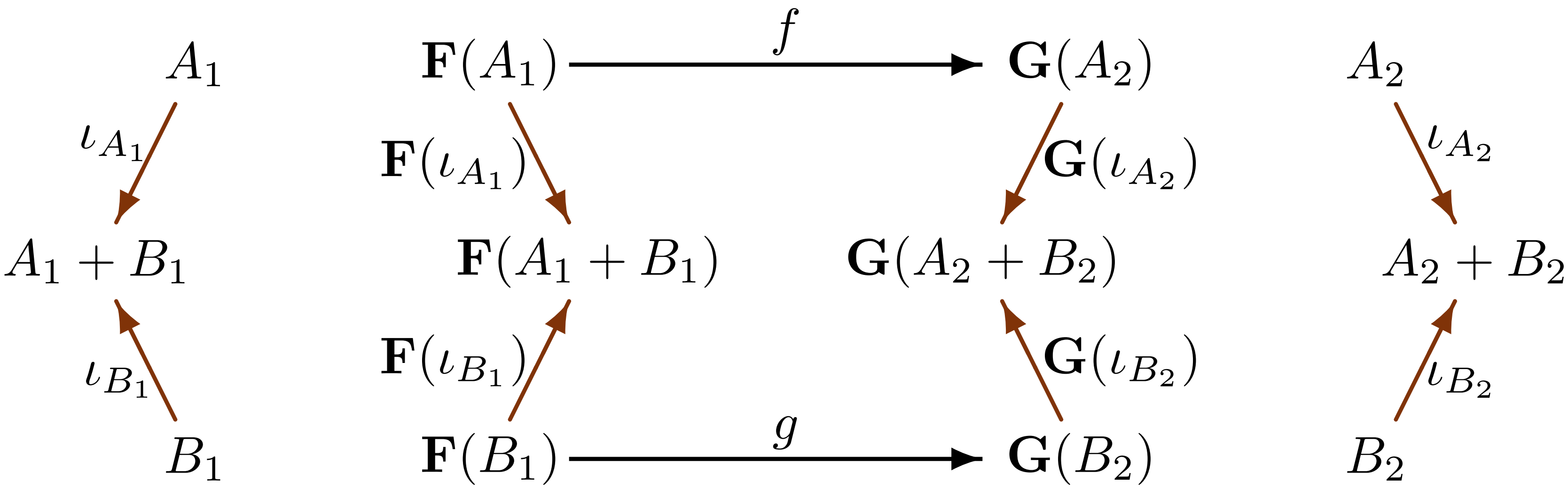
$$B_1 \qquad \mathbf{F}(B_1) \xrightarrow{\;g\;} \mathbf{G}(B_2) \qquad B_2$$

Coproducts:



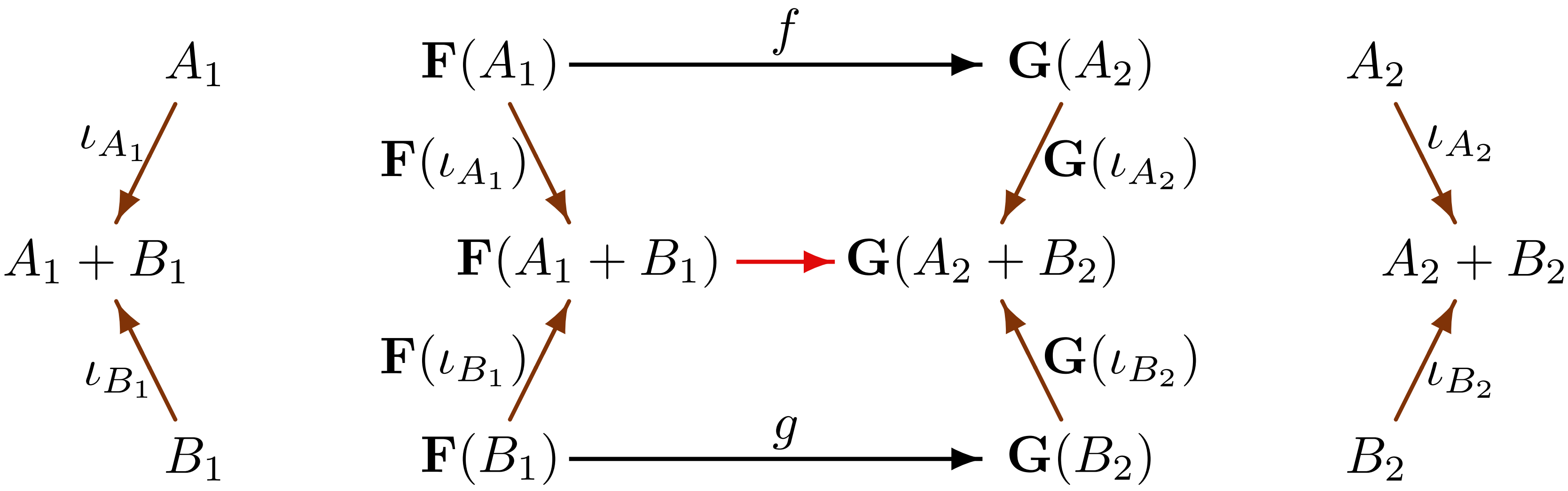
A_1
ι_{A_1}
$A_1 + B_1$
ι_{B_1}
B_1
$\mathbf{F}(A_1)$
f
$\mathbf{G}(A_2)$
$\mathbf{F}(B_1)$
g
$\mathbf{G}(B_2)$
A_2
ι_{A_2}
$A_2 + B_2$
ι_{B_2}
B_2

Coproducts:

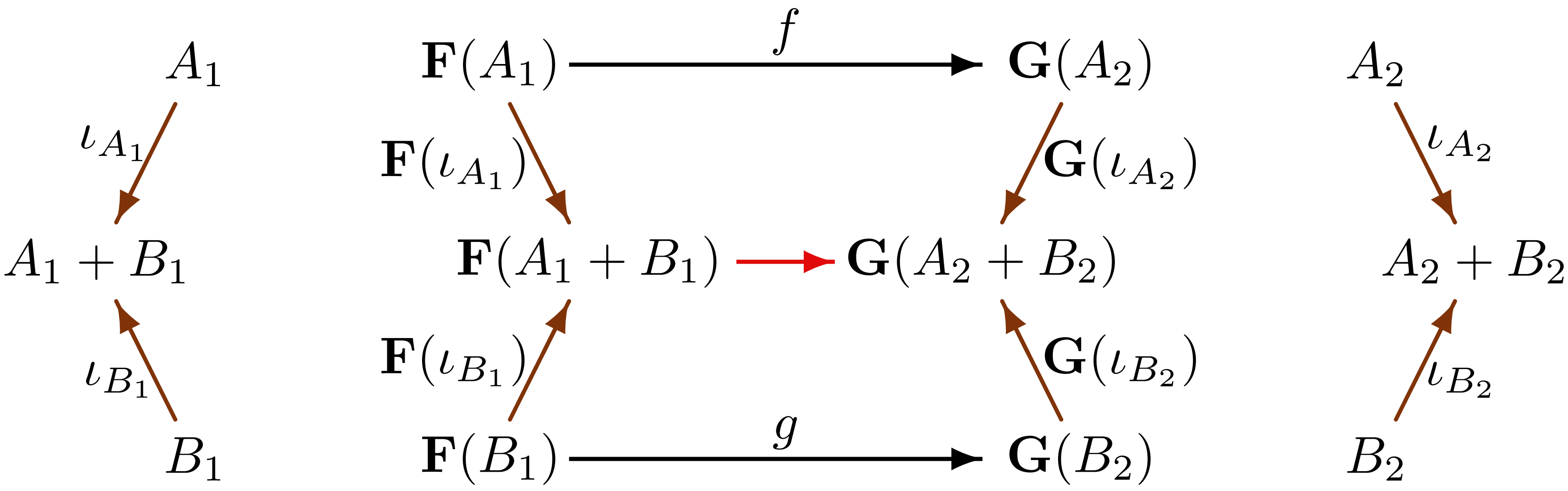


A_1
ι_{A_1}
$A_1 + B_1$
ι_{B_1}
B_1
$\mathbf{F}(A_1)$
f
$\mathbf{G}(A_2)$
$\mathbf{F}(\iota_{A_1})$
$\mathbf{G}(\iota_{A_2})$
$\mathbf{F}(A_1 + B_1)$
$\mathbf{G}(A_2 + B_2)$
$\mathbf{F}(\iota_{B_1})$
$\mathbf{G}(\iota_{B_2})$
$\mathbf{F}(B_1)$
g
$\mathbf{G}(B_2)$
A_2
ι_{A_2}
$A_2 + B_2$
ι_{B_2}
B_2

Coproducts:

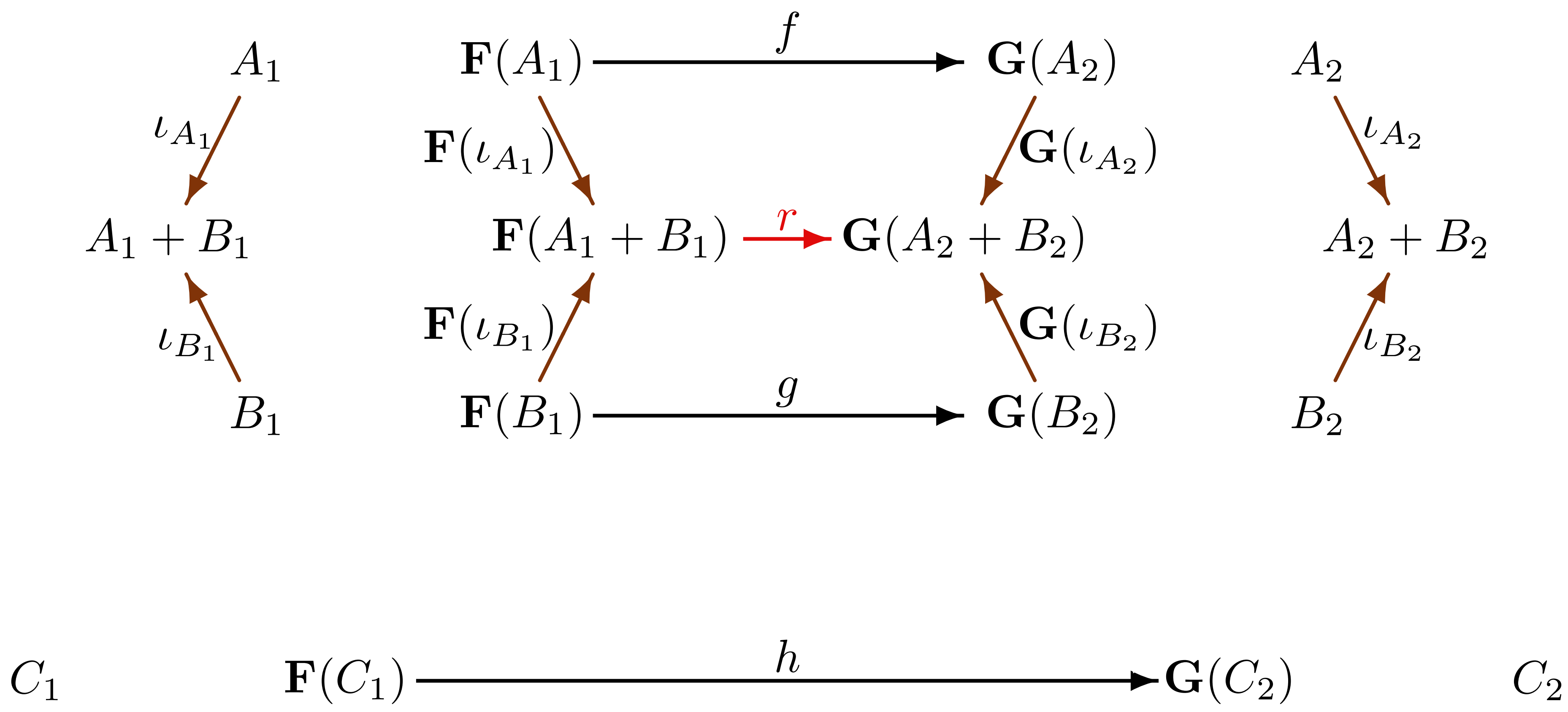

A_1
ι_{A_1}
$A_1 + B_1$
ι_{B_1}
B_1
$\mathbf{F}(A_1)$
f
$\mathbf{G}(A_2)$
$\mathbf{F}(\iota_{A_1})$
$\mathbf{G}(\iota_{A_2})$
$\mathbf{F}(A_1 + B_1)$
$\mathbf{G}(A_2 + B_2)$
$\mathbf{F}(\iota_{B_1})$
$\mathbf{G}(\iota_{B_2})$
$\mathbf{F}(B_1)$
g
$\mathbf{G}(B_2)$
A_2
ι_{A_2}
$A_2 + B_2$
ι_{B_2}
B_2

Coproducts:



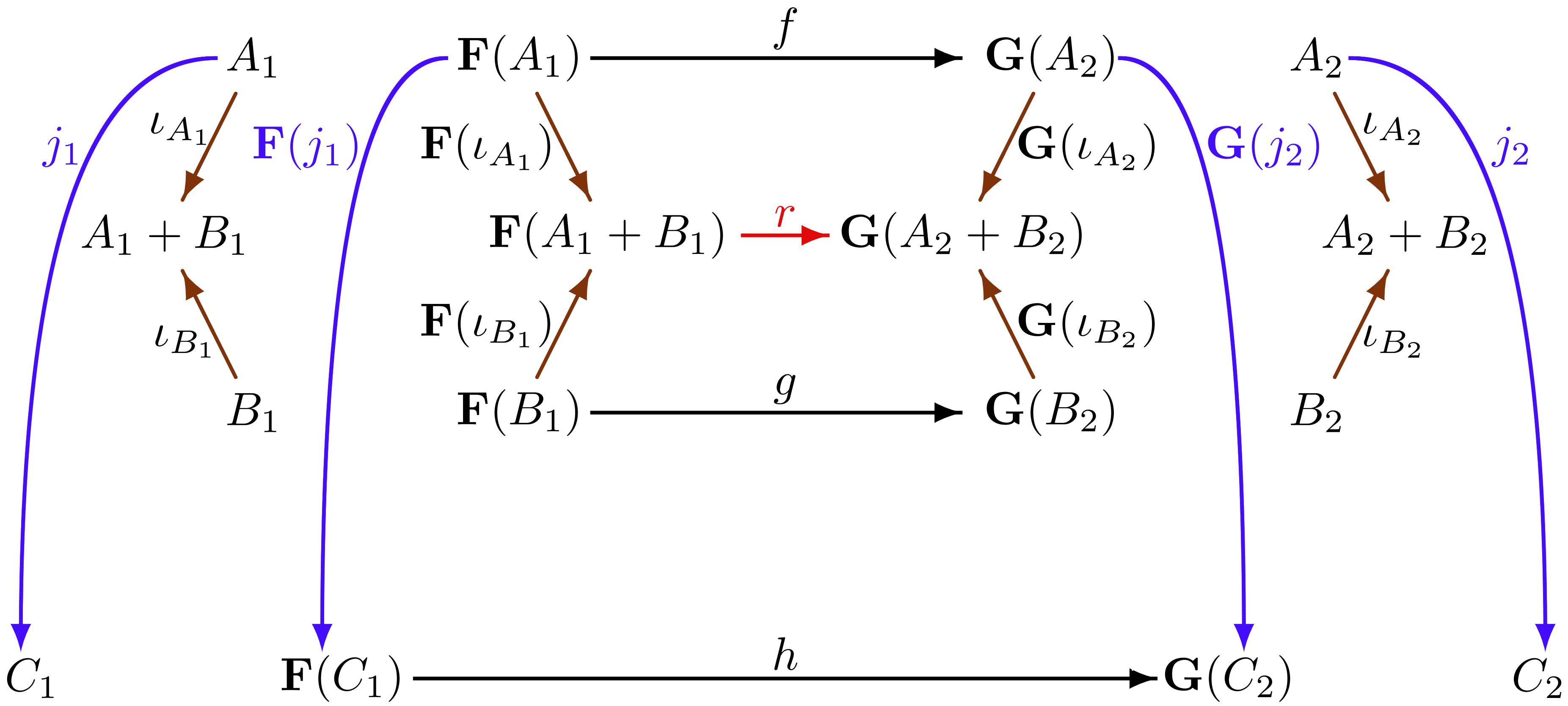
Fact: $\langle A_1 + B_1, [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})] \colon \mathbf{F}(A_1 + B_1) \to \mathbf{G}(A_2 + B_2), A_2 + B_2\rangle$ *with injections* $\langle \iota_{A_1}, \iota_{A_2}\rangle$ *and* $\langle \iota_{B_1}, \iota_{B_2}\rangle$ *is a coproduct of* $\langle A_1, f \colon \mathbf{F}(A_1) \to \mathbf{G}(A_2), A_2\rangle$ *and* $\langle B_1, g \colon \mathbf{F}(B_1) \to \mathbf{G}(B_2), B_2\rangle$ *in* $(\mathbf{F}, \mathbf{G})$.

Coproducts:



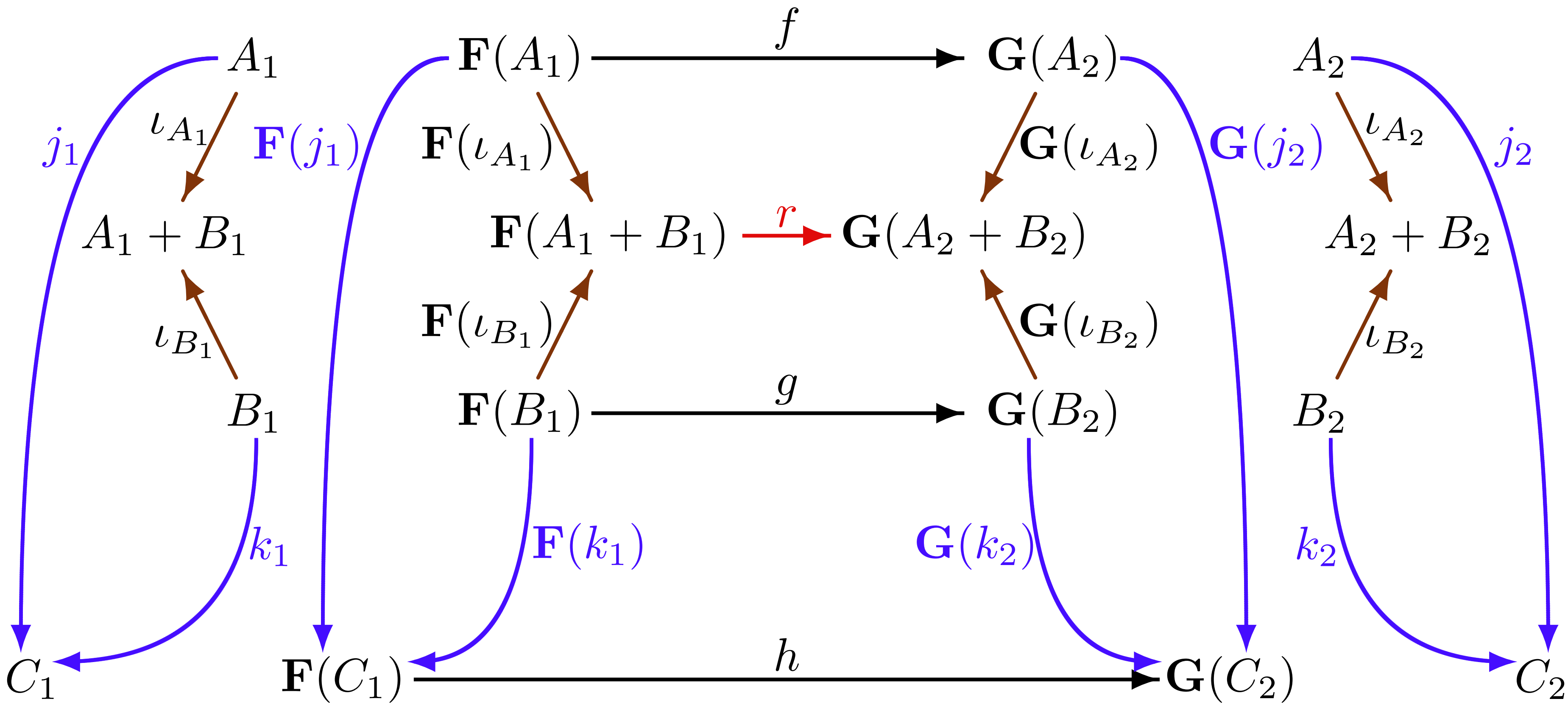
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$,

Coproducts:



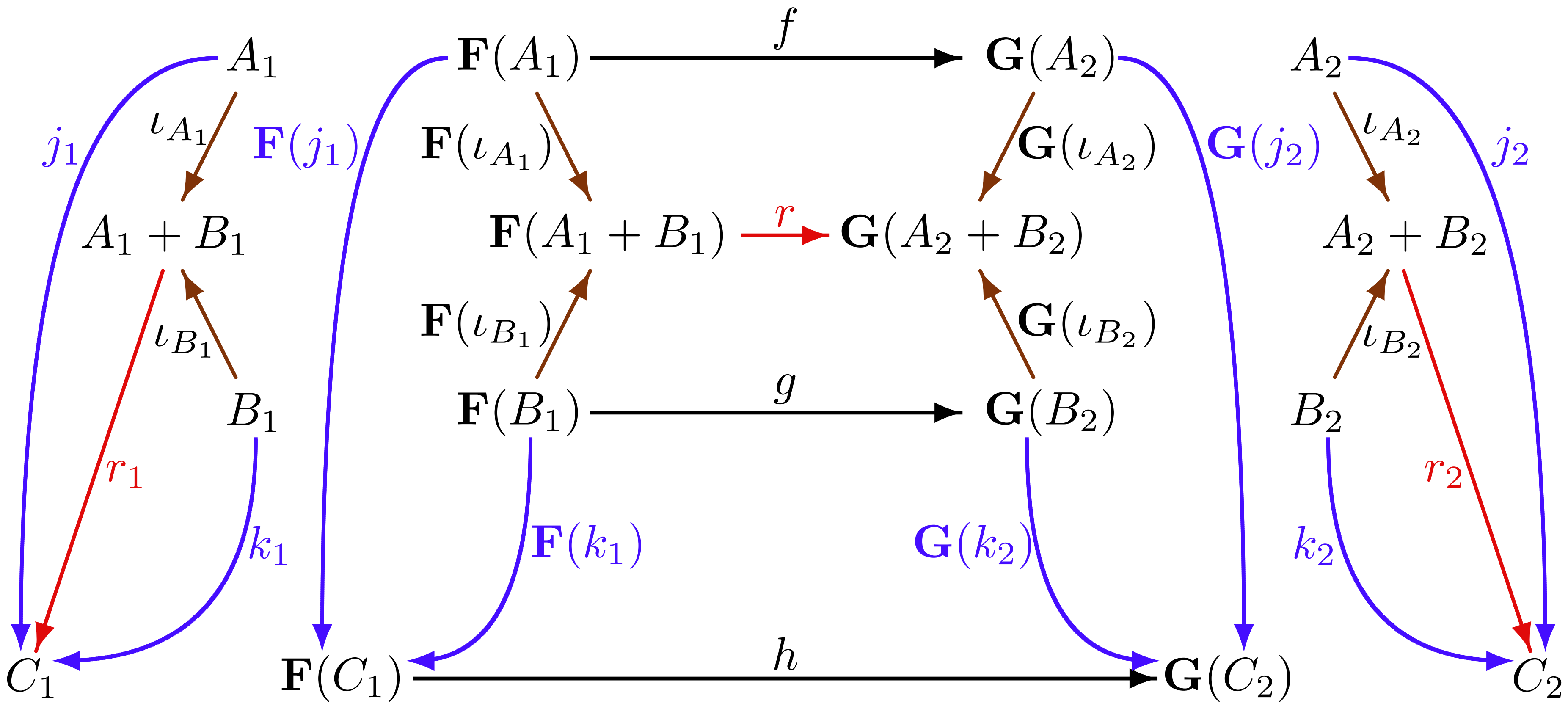
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$,

Coproducts:



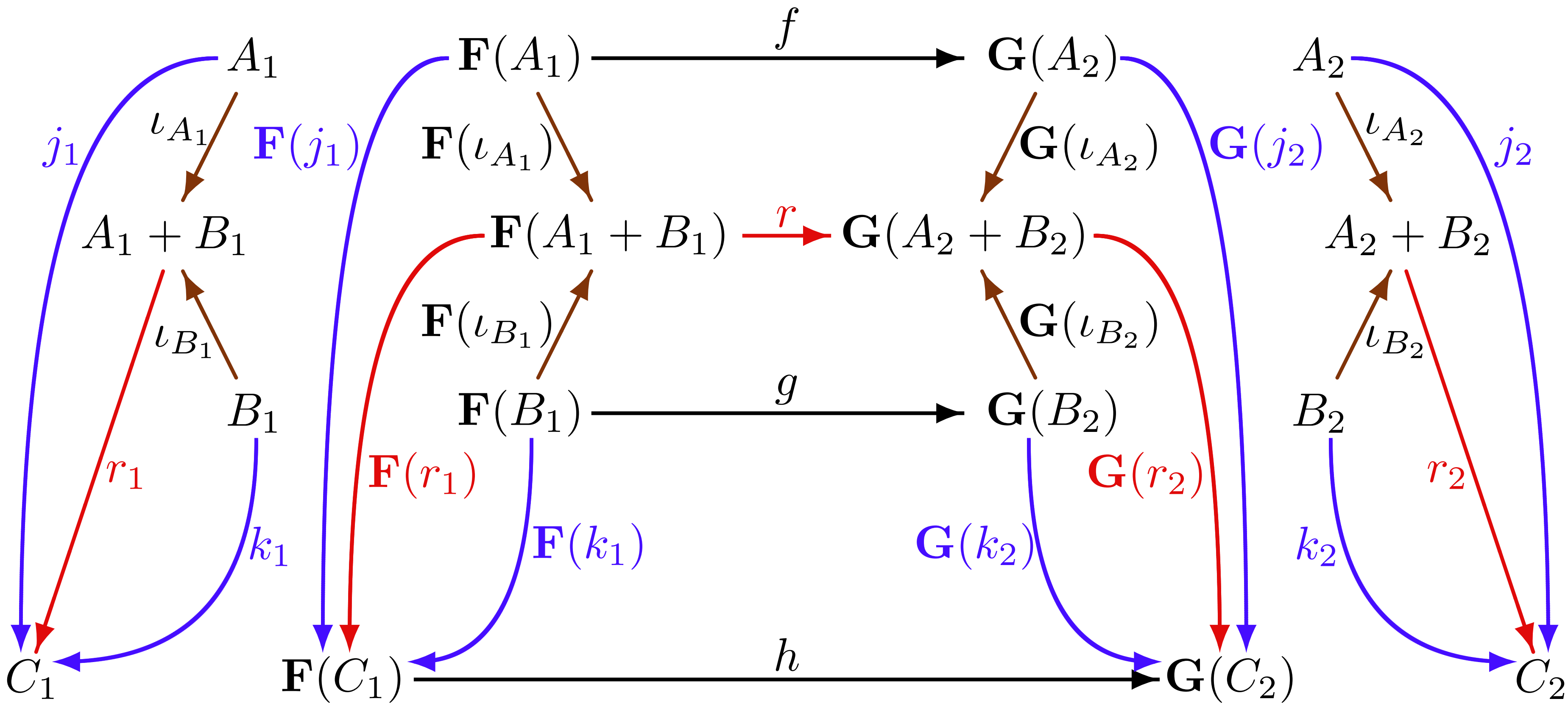
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$,

Coproducts:



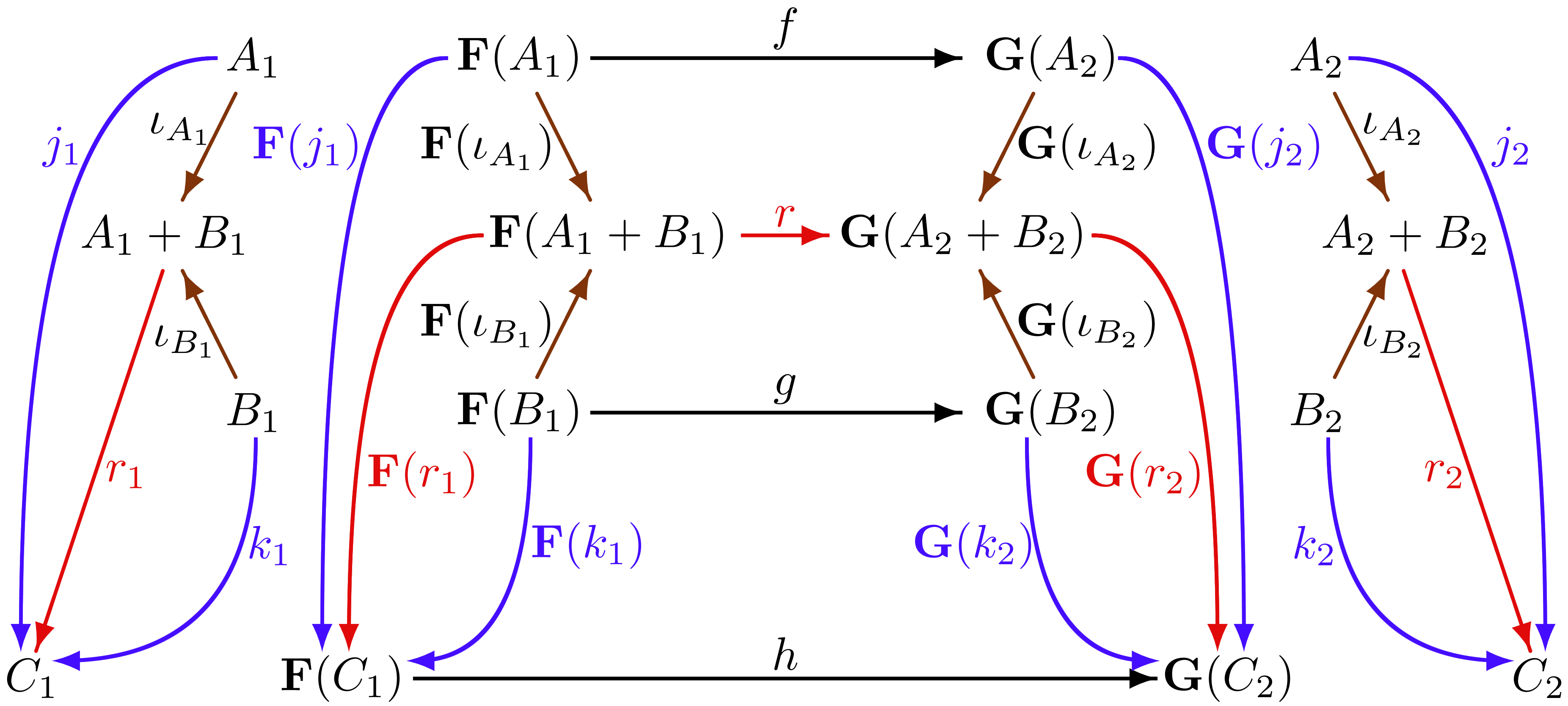
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$, $r_1 = [j_1, k_1]$, $r_2 = [j_2, k_2]$.

Coproducts:



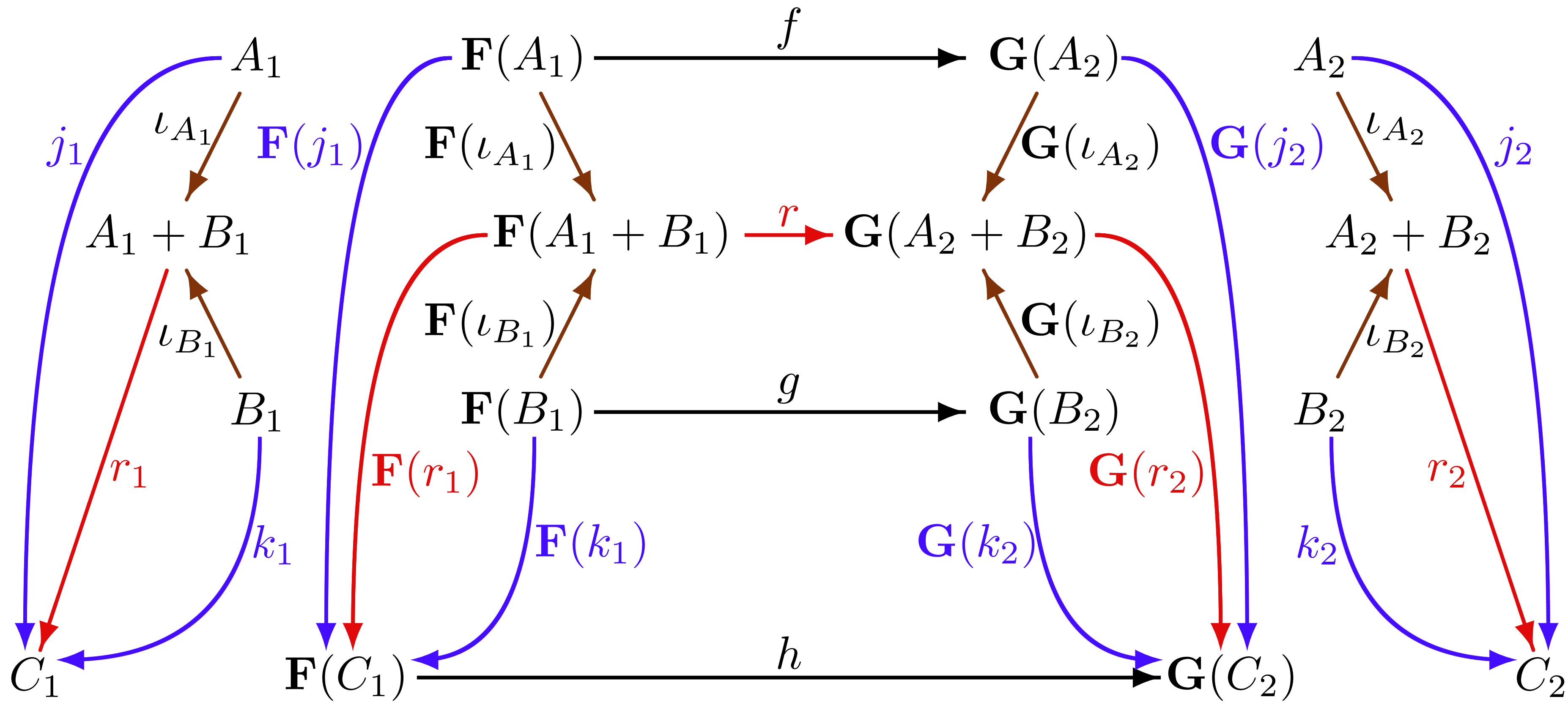
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$, $r_1 = [j_1, k_1]$, $r_2 = [j_2, k_2]$.

Coproducts:



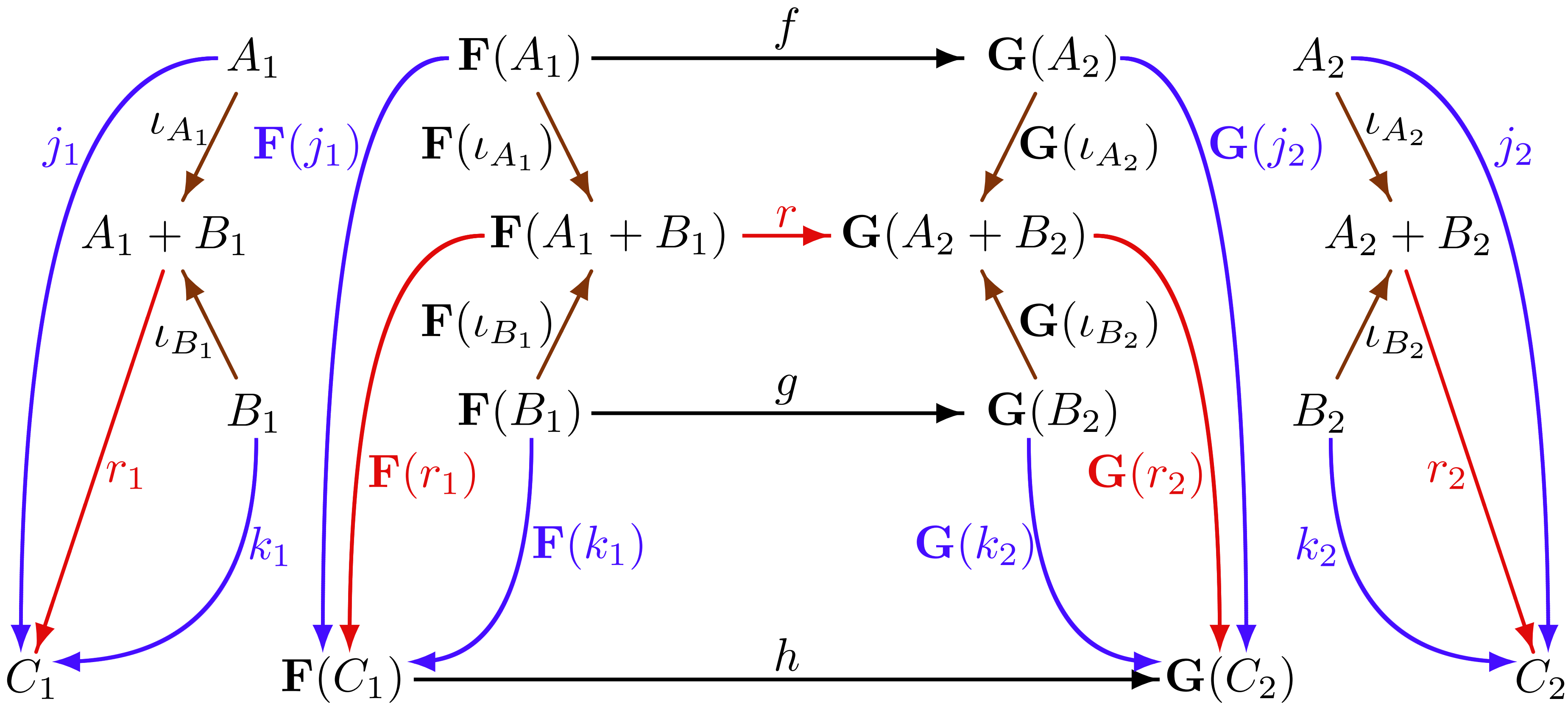
where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$,
$r_1 = [j_1, k_1]$, $r_2 = [j_2, k_2]$. We need $\boxed{\mathbf{F}(r_1);h = r;\mathbf{G}(r_2)}$

Coproducts:



where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$, $r_1 = [j_1, k_1]$, $r_2 = [j_2, k_2]$. We need $\boxed{\mathbf{F}(r_1);h = r;\mathbf{G}(r_2)}$ This follows from $\mathbf{F}(\iota_{A_1});\mathbf{F}(r_1);h = \mathbf{F}(\iota_{A_1});r;\mathbf{G}(r_2)$ and $\mathbf{F}(\iota_{B_1});\mathbf{F}(r_1);h = \mathbf{F}(\iota_{B_1});r;\mathbf{G}(r_2)$.

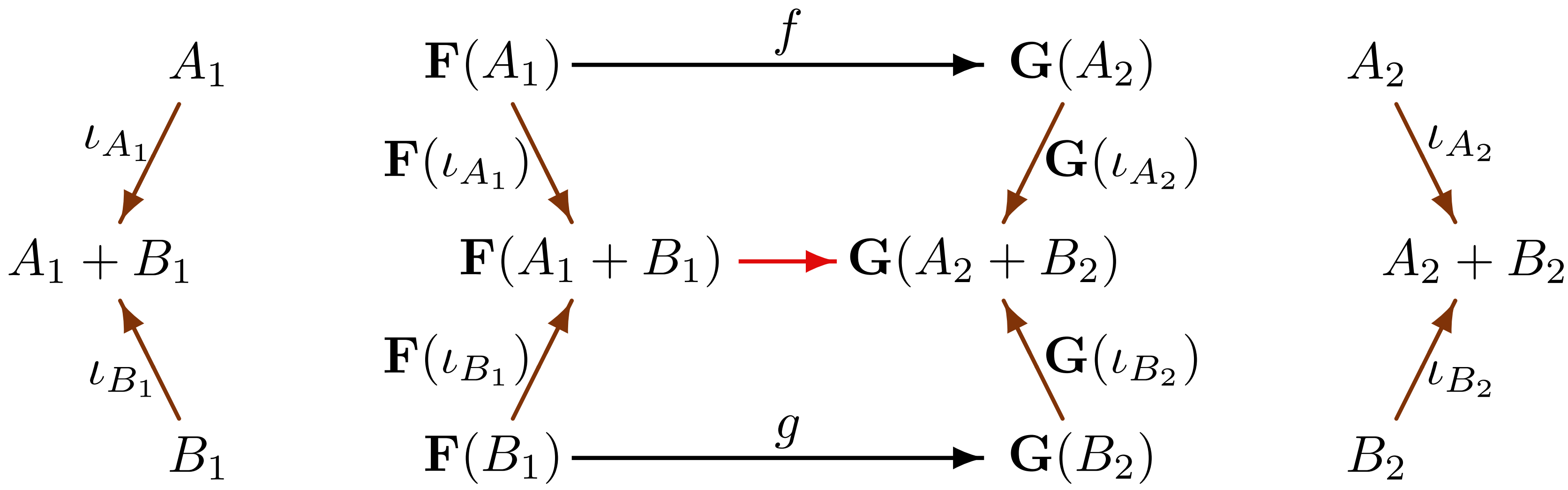
Coproducts:



where $r = [f;\mathbf{G}(\iota_{A_2}), g;\mathbf{G}(\iota_{B_2})]$, $\mathbf{F}(j_1);h = f;\mathbf{G}(j_2)$, $\mathbf{F}(k_1);h = g;\mathbf{G}(k_2)$, $r_1 = [j_1, k_1]$, $r_2 = [j_2, k_2]$. We need $\boxed{\mathbf{F}(r_1);h = r;\mathbf{G}(r_2)}$ This follows from $\mathbf{F}(\iota_{A_1});\mathbf{F}(r_1);h = \mathbf{F}(\iota_{A_1});r;\mathbf{G}(r_2)$ and $\mathbf{F}(\iota_{B_1});\mathbf{F}(r_1);h = \mathbf{F}(\iota_{B_1});r;\mathbf{G}(r_2)$.

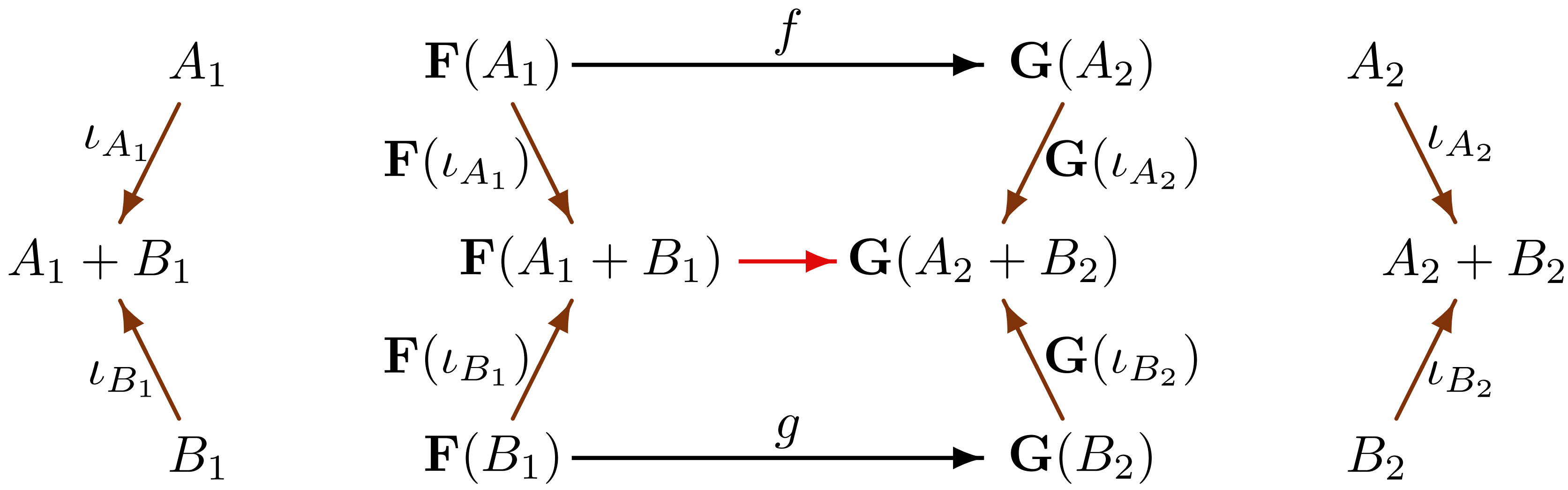
– $\mathbf{F}(\iota_{A_1});\mathbf{F}(r_1);h = \mathbf{F}(j_1);h = f;\mathbf{G}(j_2) = f;\mathbf{G}(\iota_{A_2});\mathbf{G}(r_2) = \mathbf{F}(\iota_{A_1});r;\mathbf{G}(r_2)$

Coproducts:

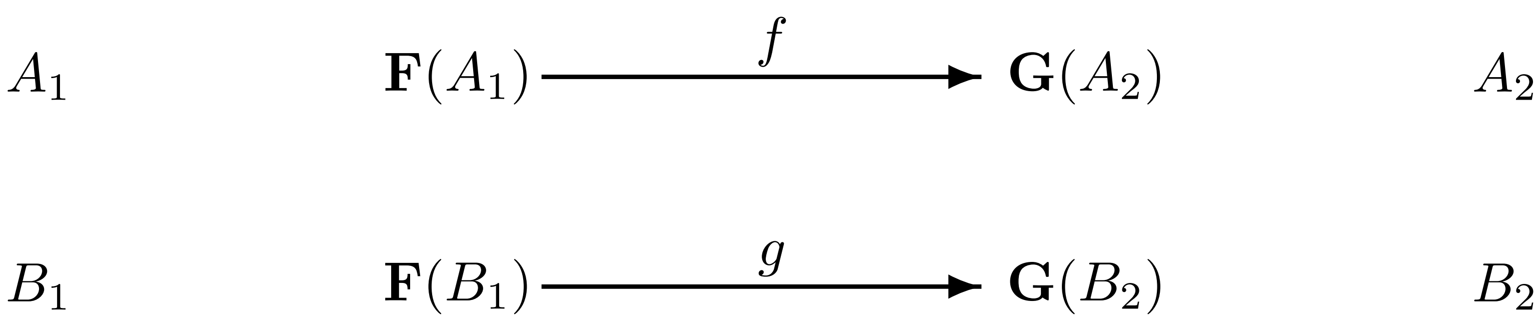


Coequalisers:

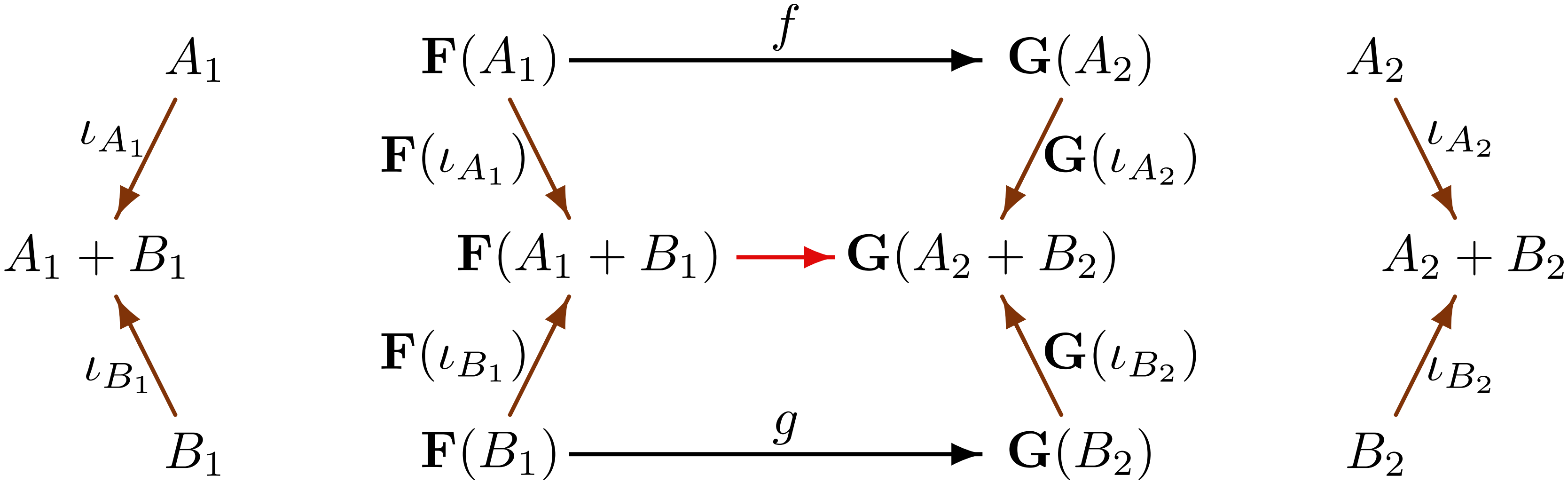
Coproducts:



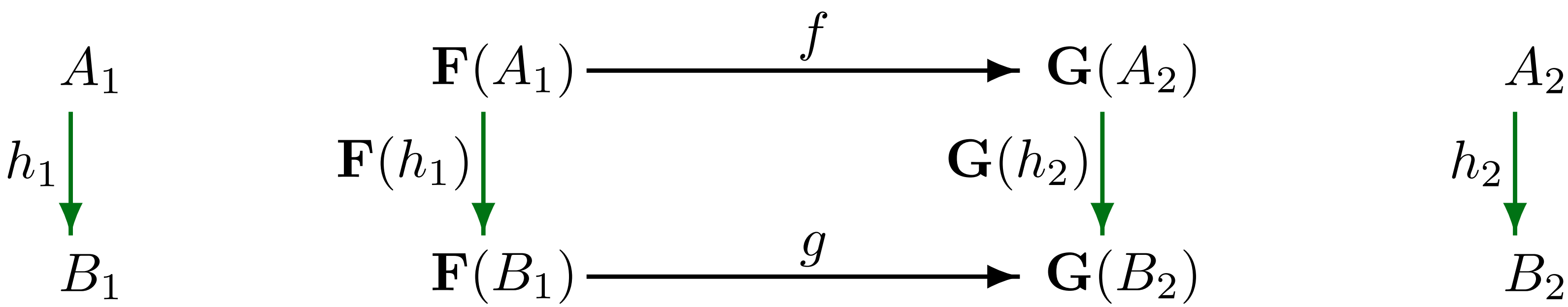
Coequalisers:



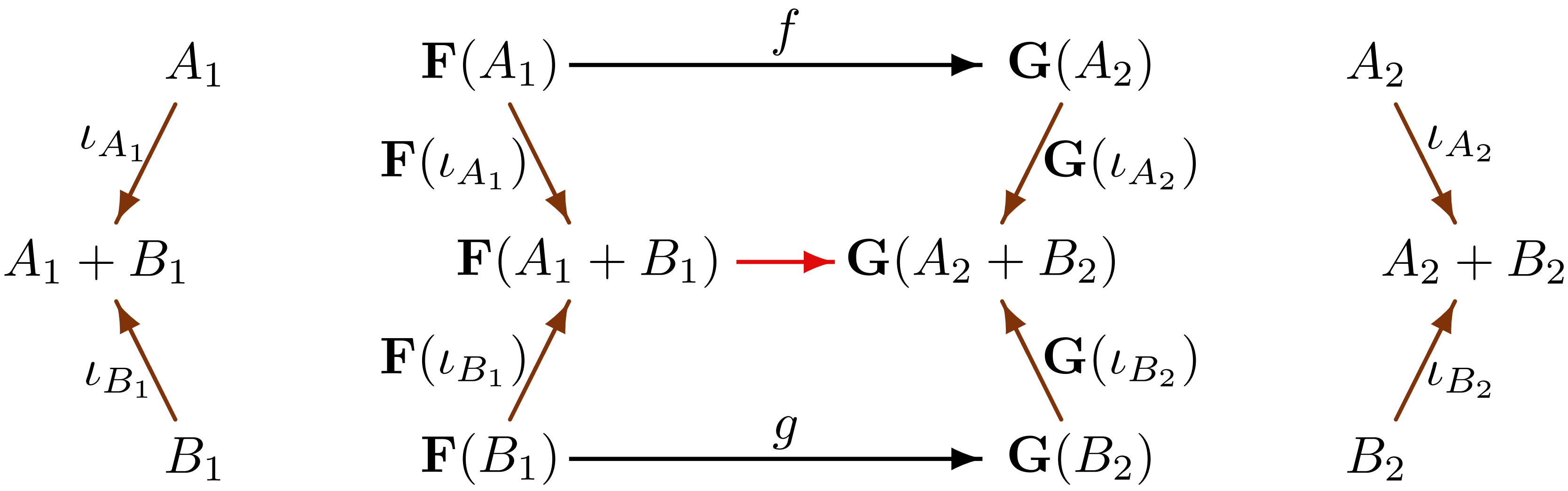
Coproducts:



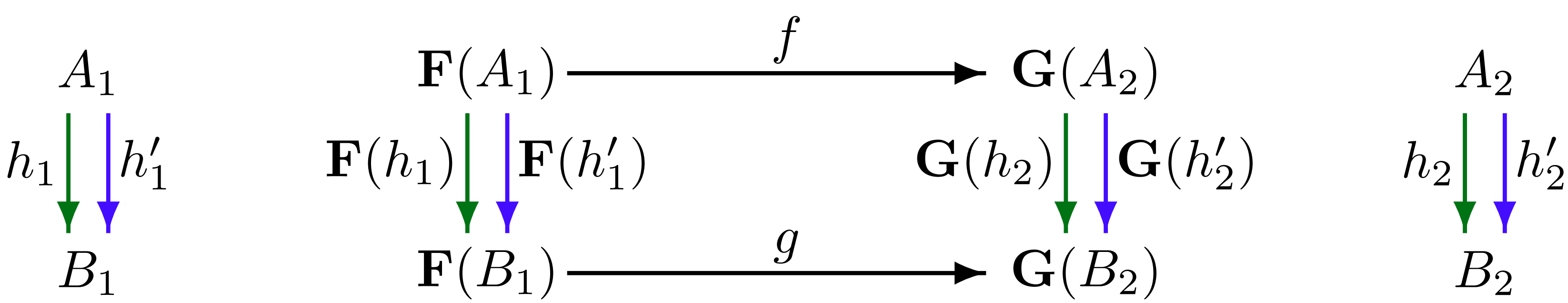
Coequalisers:



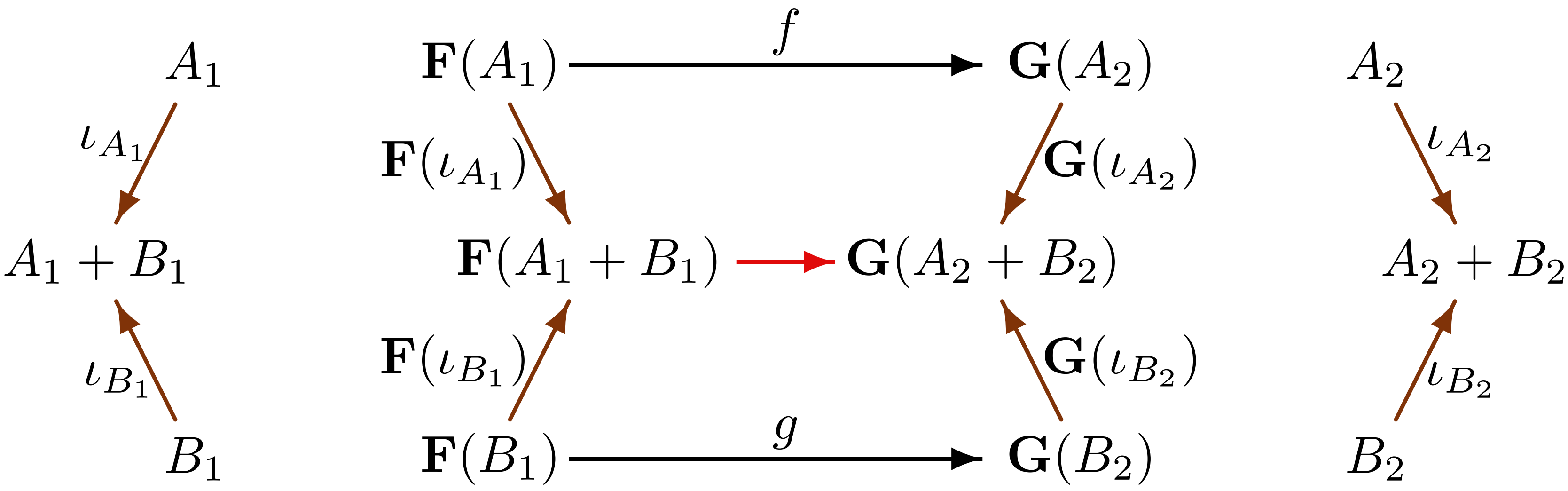
Coproducts:



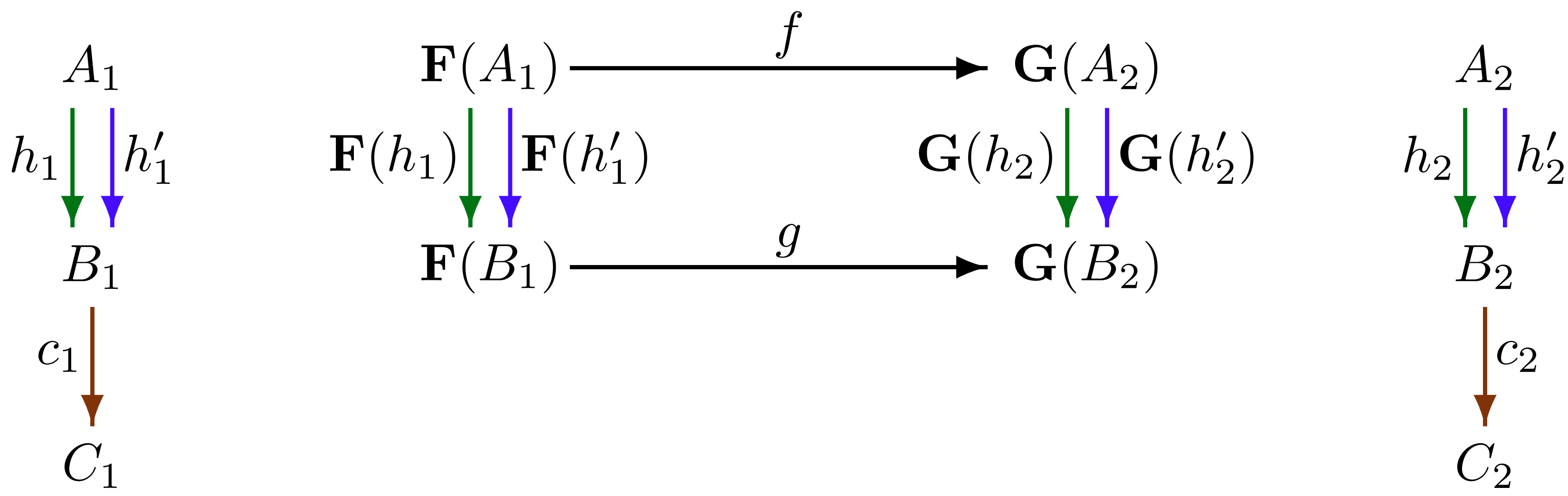
Coequalisers:



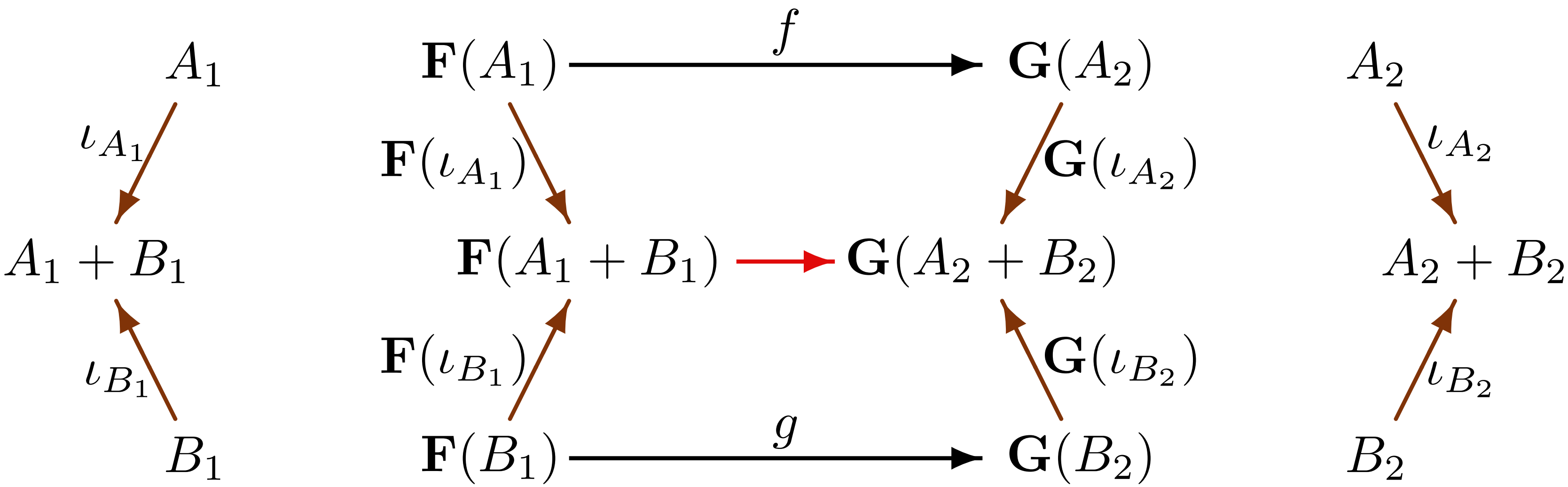
Coproducts:



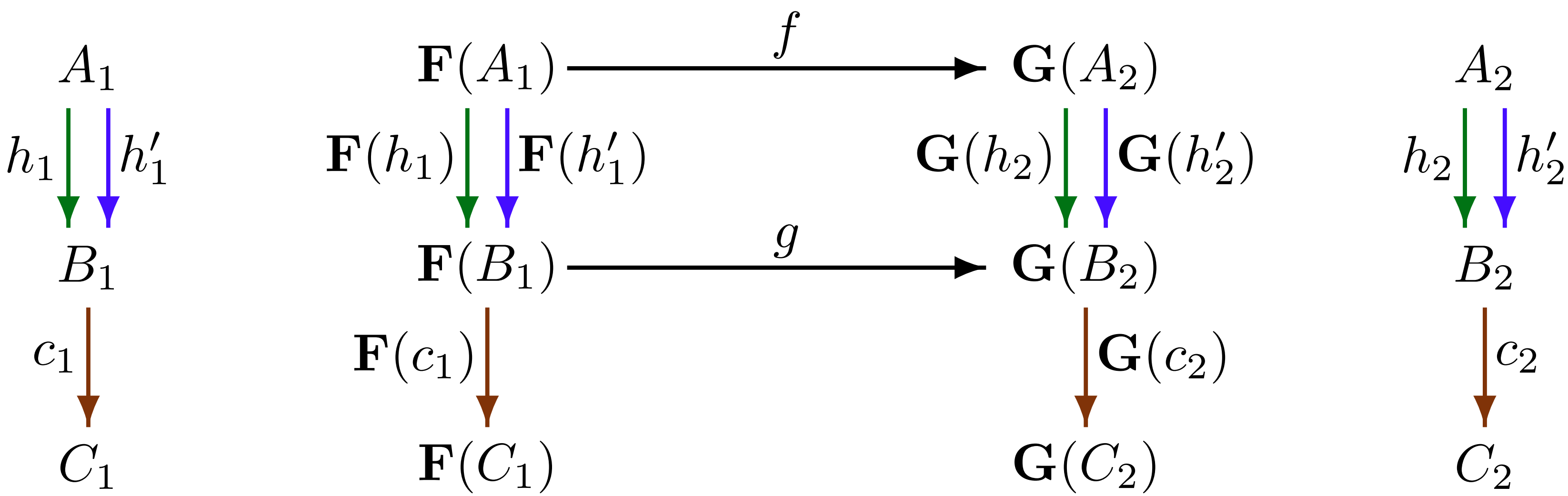
Coequalisers:



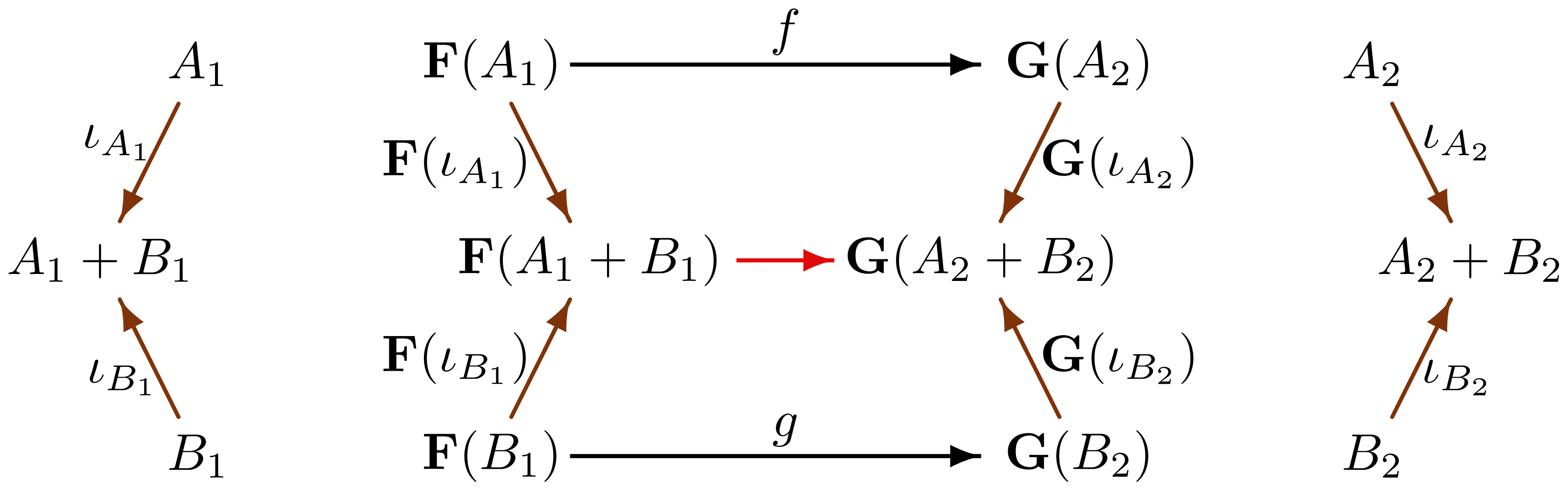
Coproducts:



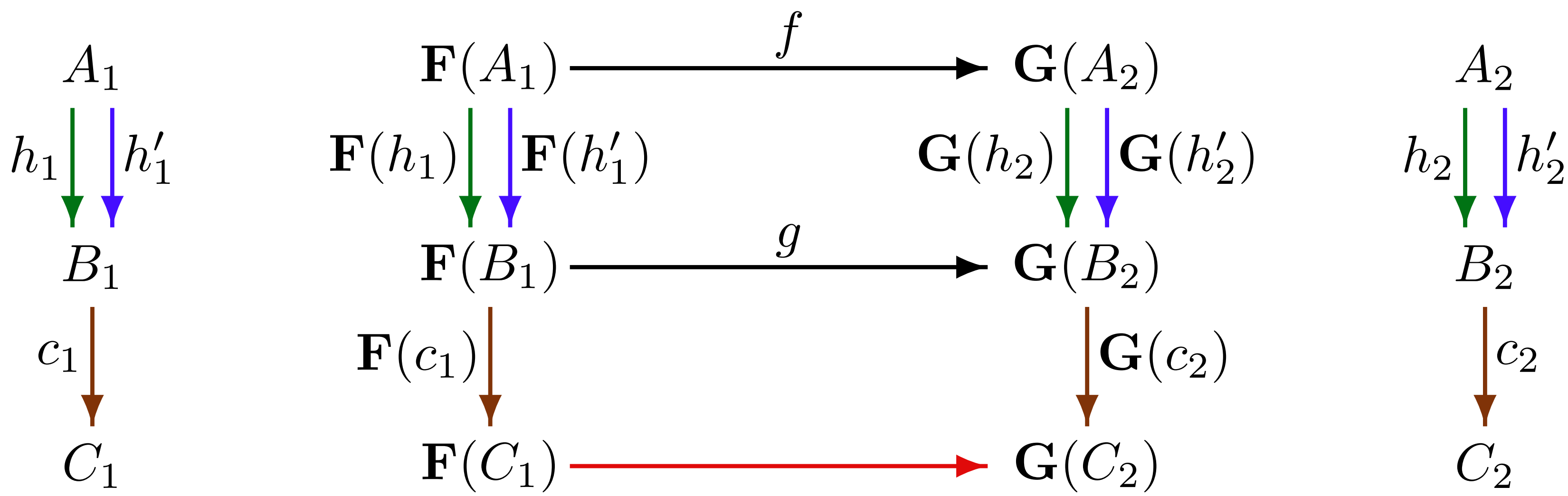
Coequalisers:



Coproducts:



Coequalisers:



Indexed categories

Indexed categories

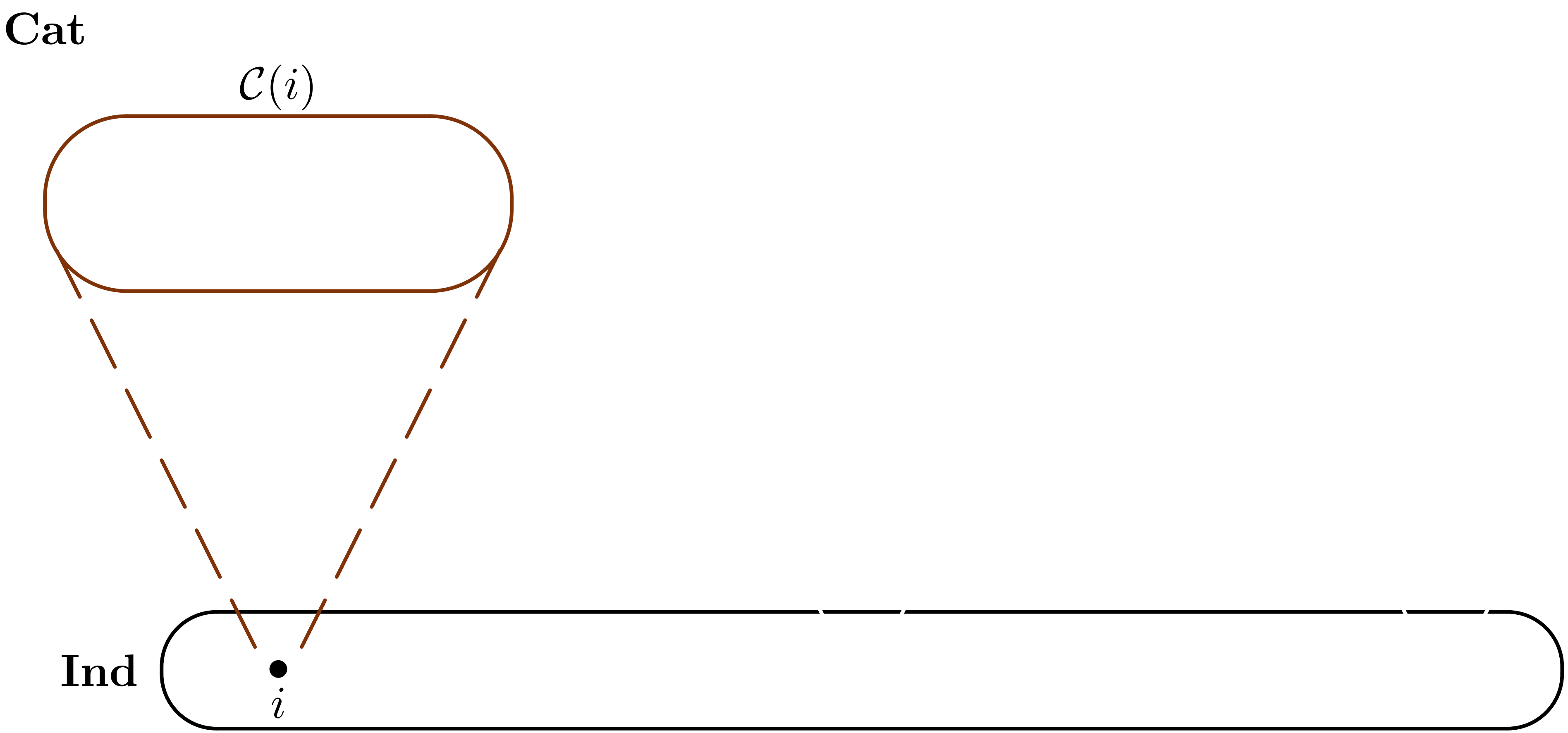
Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

Indexed categories

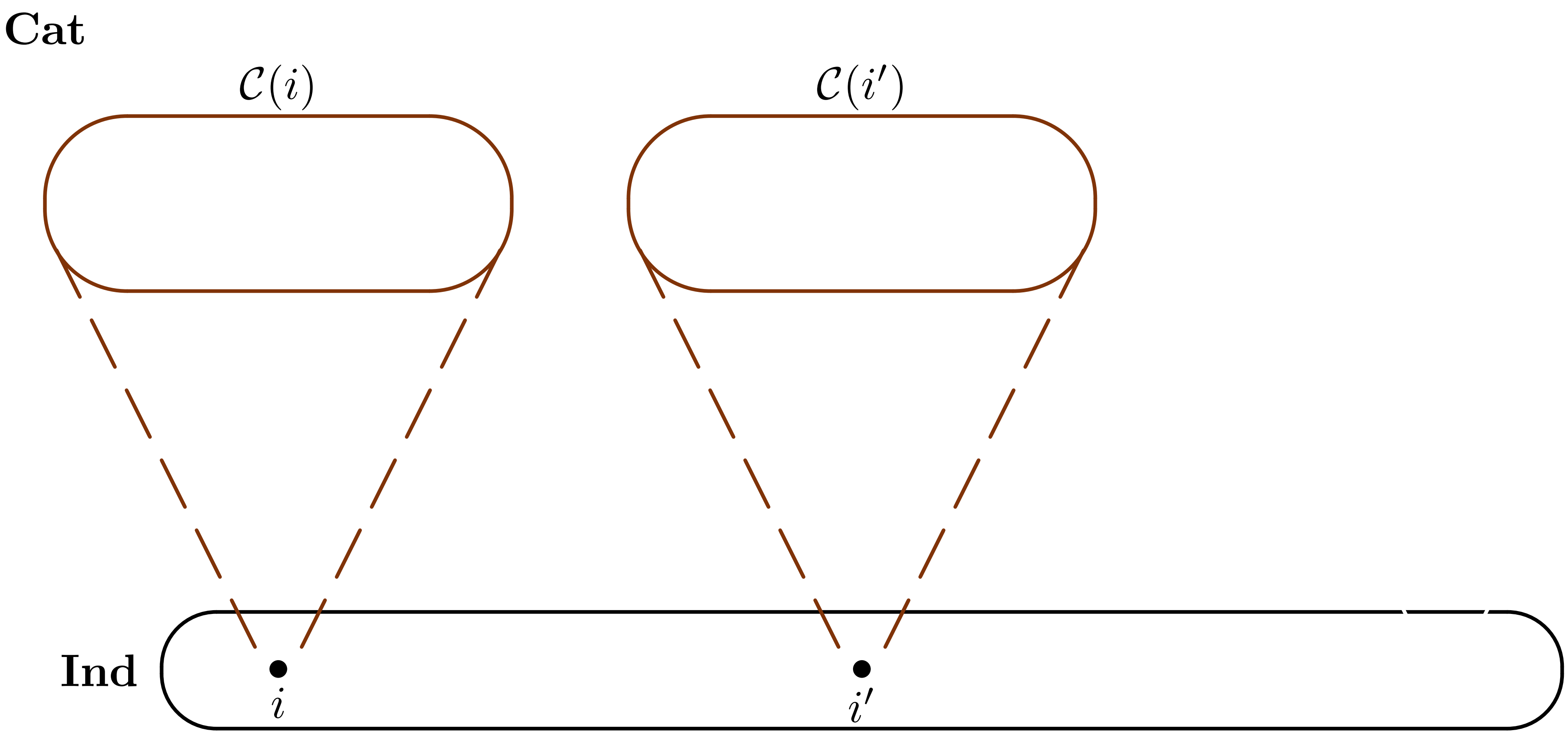
An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

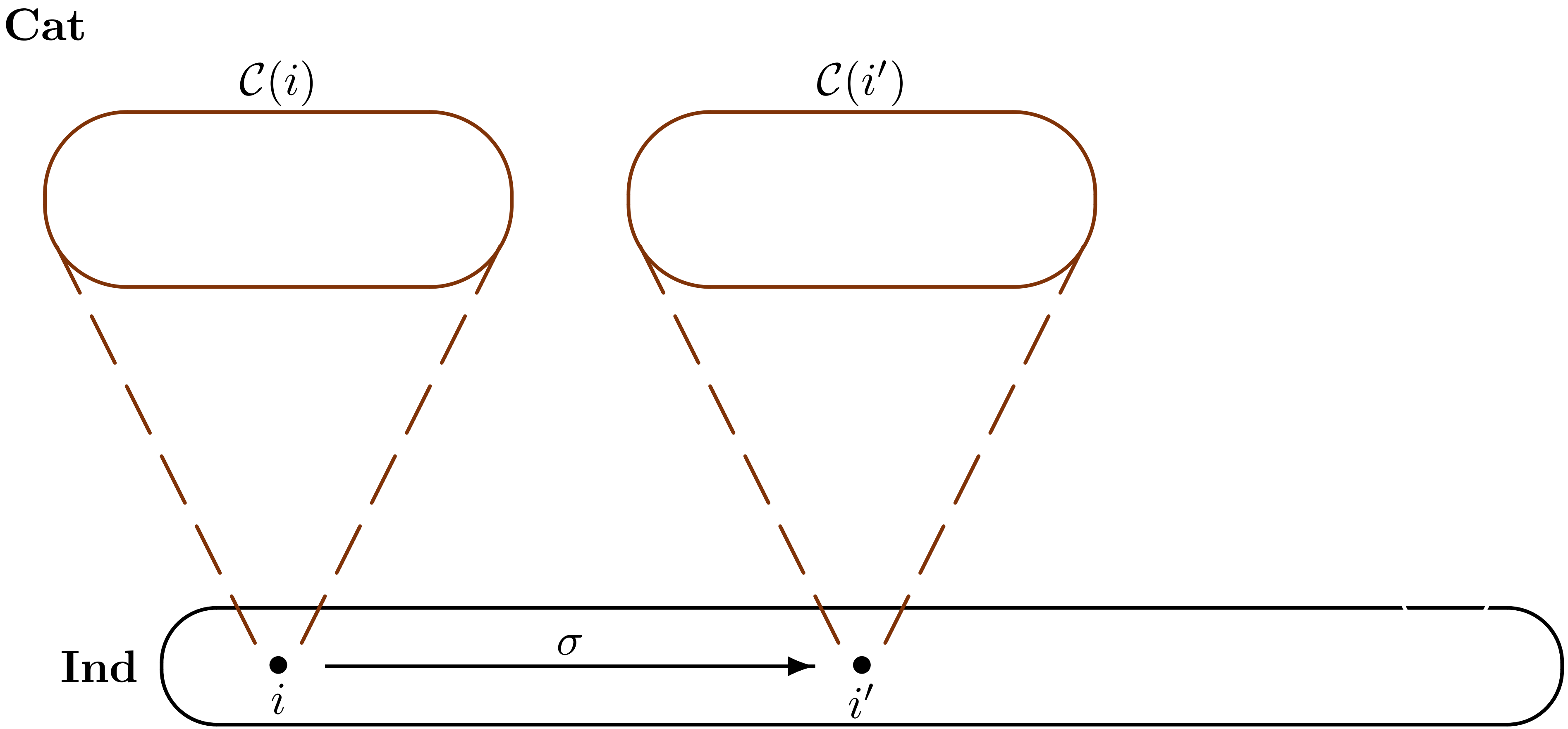

Ind
i



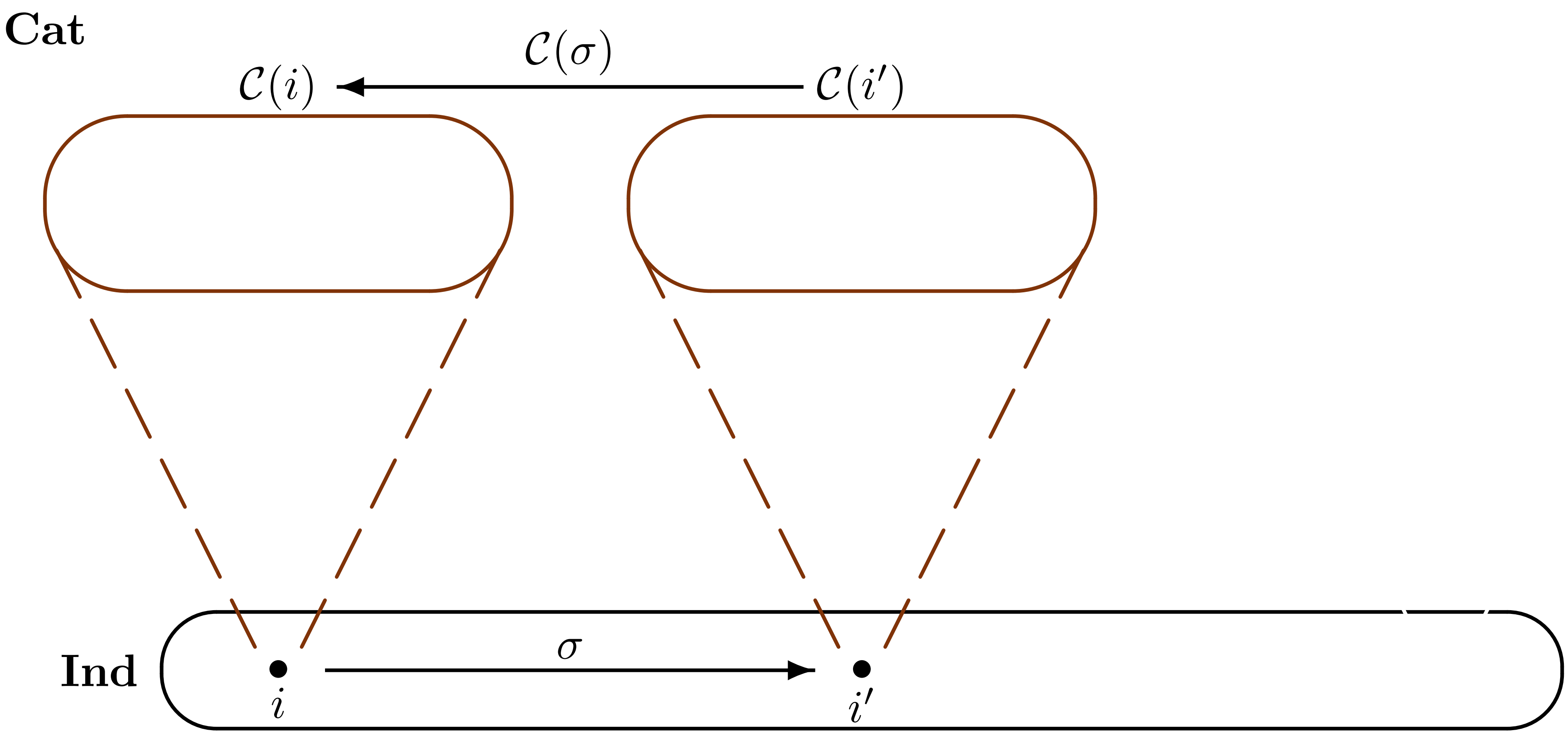
Cat
$\mathcal{C}(i)$
Ind
i



Cat
$\mathcal{C}(i)$
$\mathcal{C}(i')$
Ind
i
i'



Cat
$\mathcal{C}(i)$
$\mathcal{C}(i')$
Ind
σ
i
i'


Cat
C(σ)
C(i)
C(i′)
Ind
σ
i
i′

Indexed categories

An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

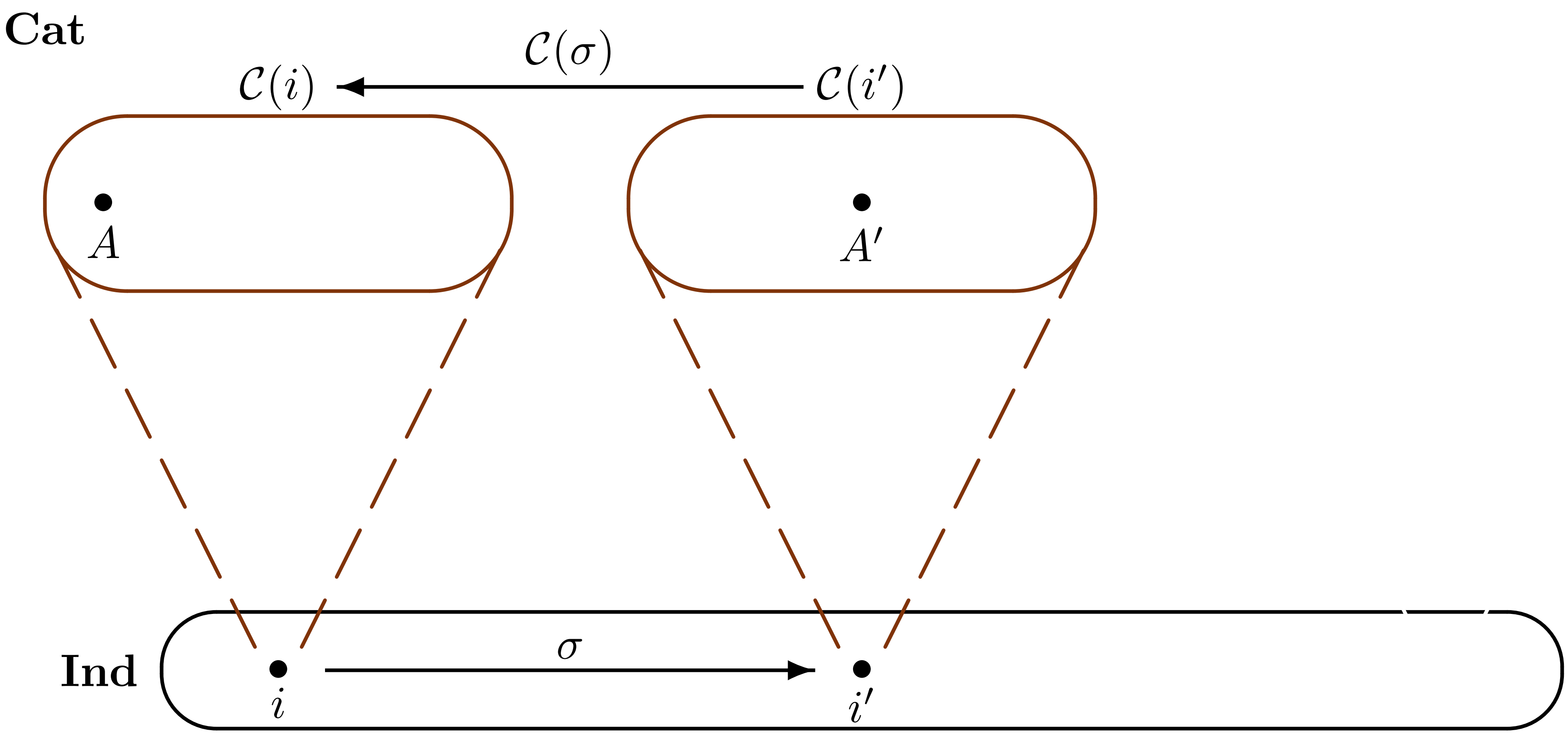
Indexed categories

An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

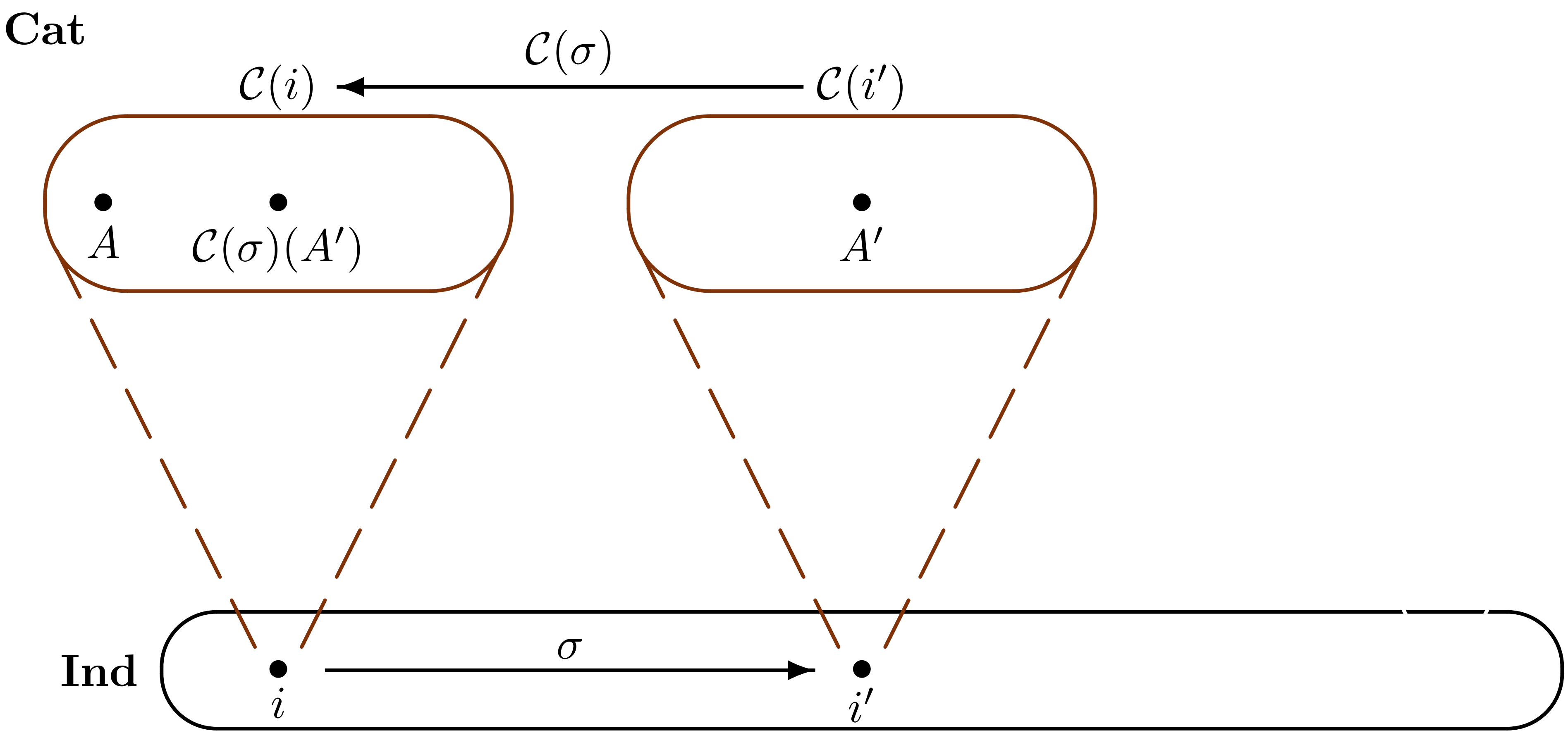
Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

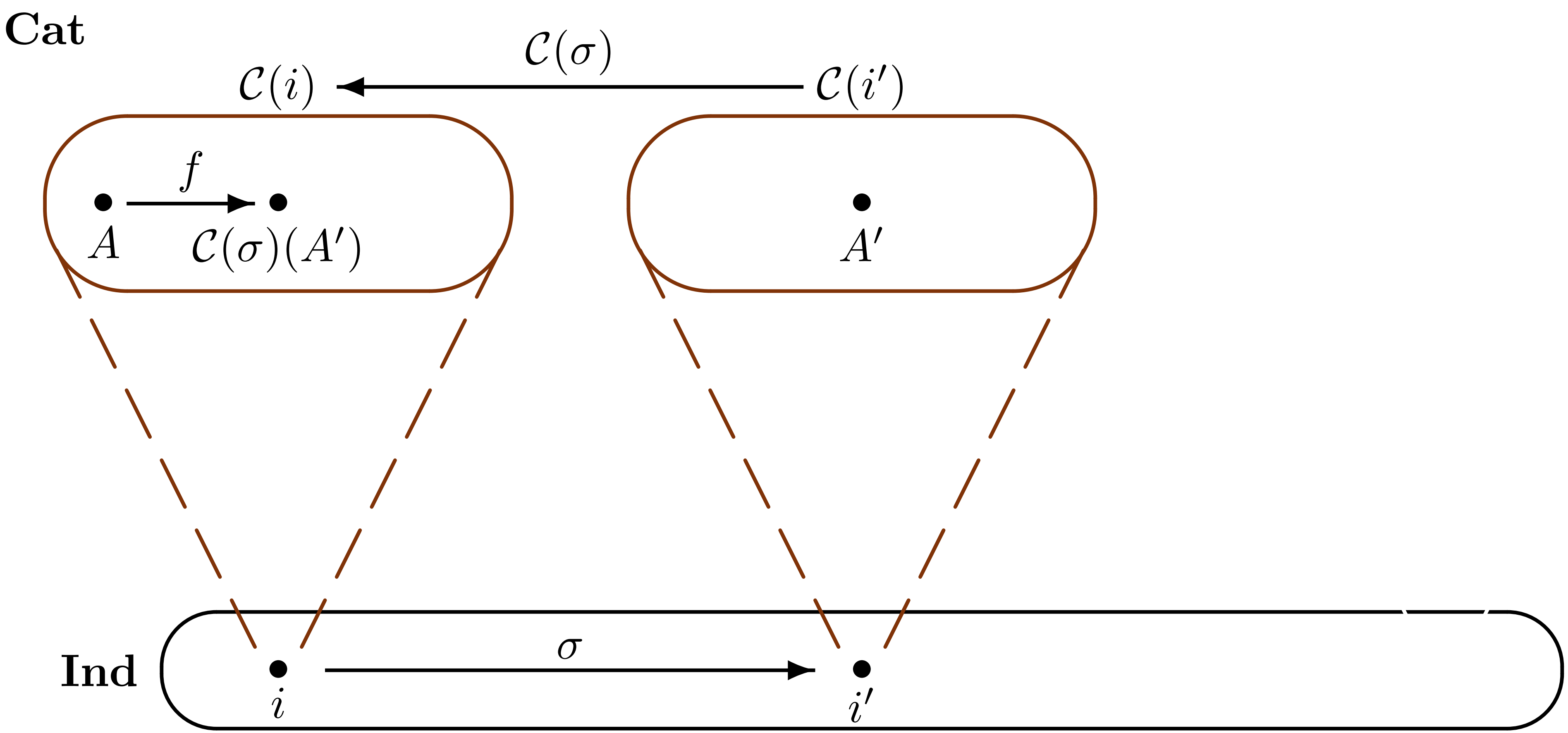
– objects: $\langle i, A\rangle$ for all $i \in |\mathbf{Ind}|$, $A \in |\mathcal{C}(i)|$



Cat
$\mathcal{C}(\sigma)$
$\mathcal{C}(i)$
$\mathcal{C}(i')$
A
A'
Ind
σ
i
i'



Cat
$\mathcal{C}(\sigma)$
$\mathcal{C}(i)$
$\mathcal{C}(i')$
A
$\mathcal{C}(\sigma)(A')$
A'
Ind
σ
i
i'


Cat
C(σ)
C(i)
C(i′)
f
A
C(σ)(A′)
A′
Ind
σ
i
i′

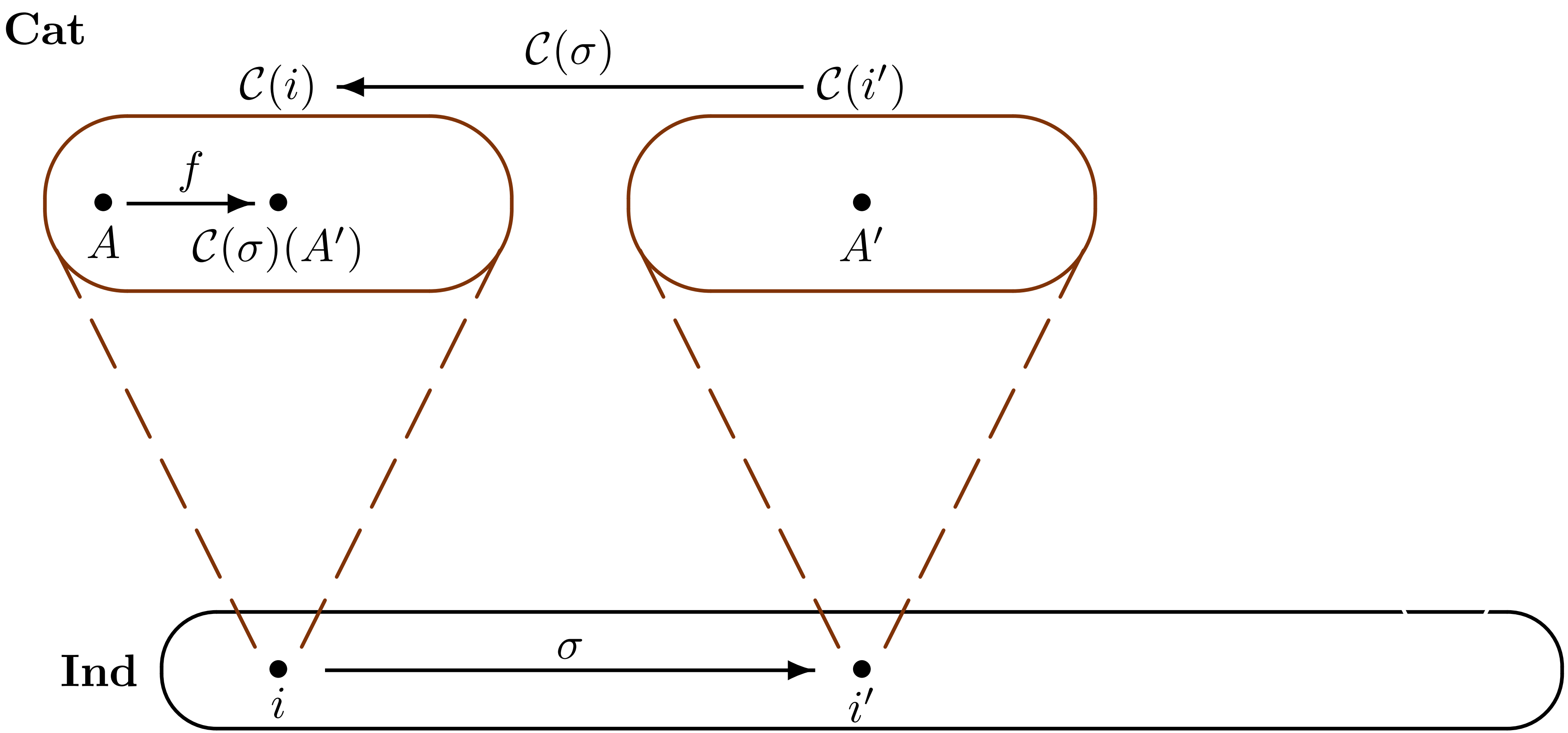
Indexed categories

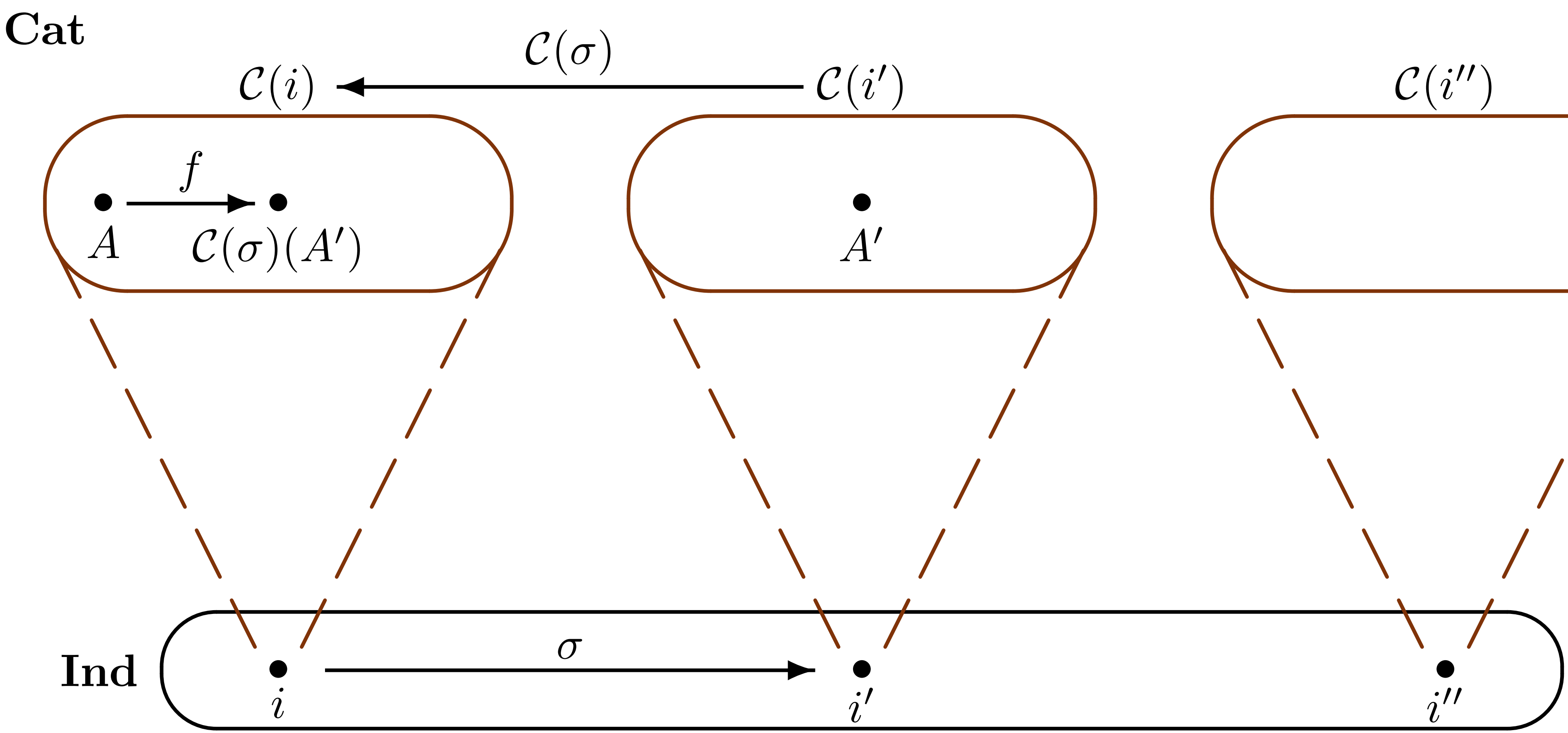
An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

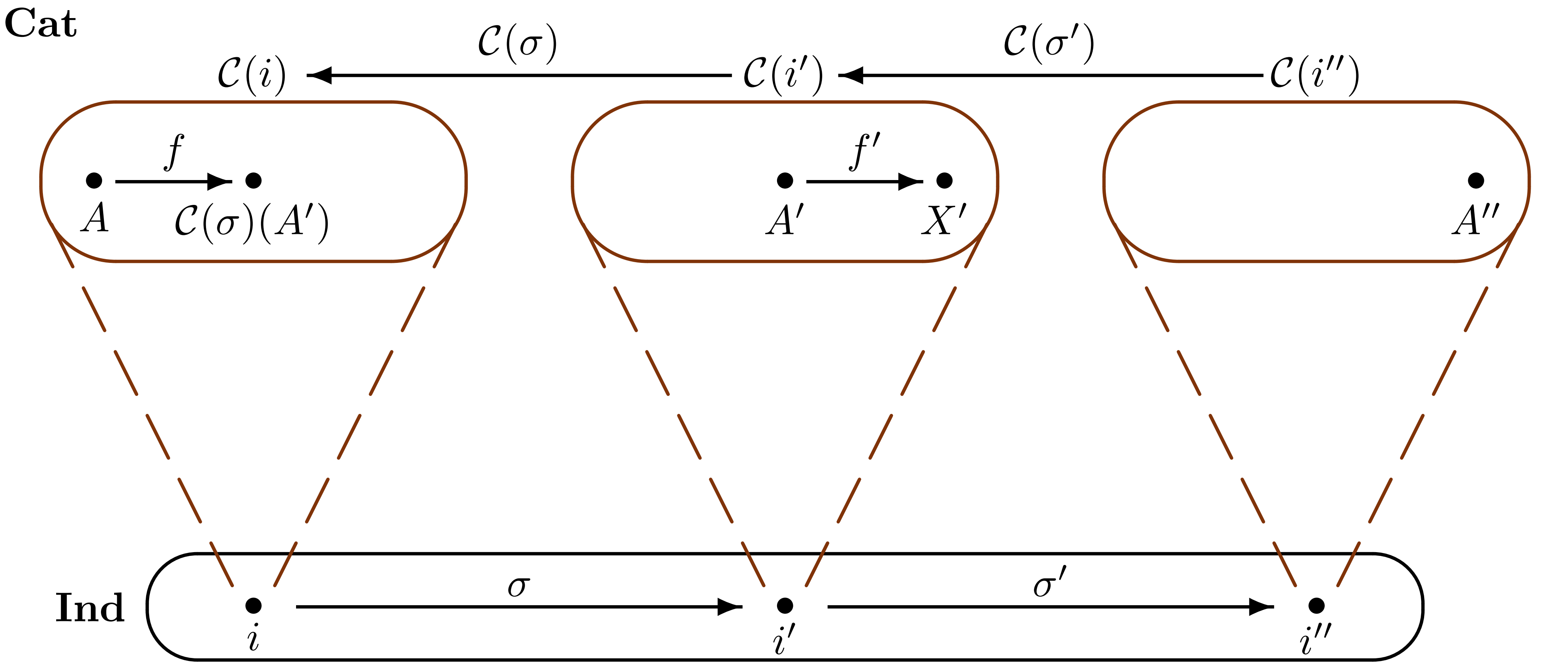
Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

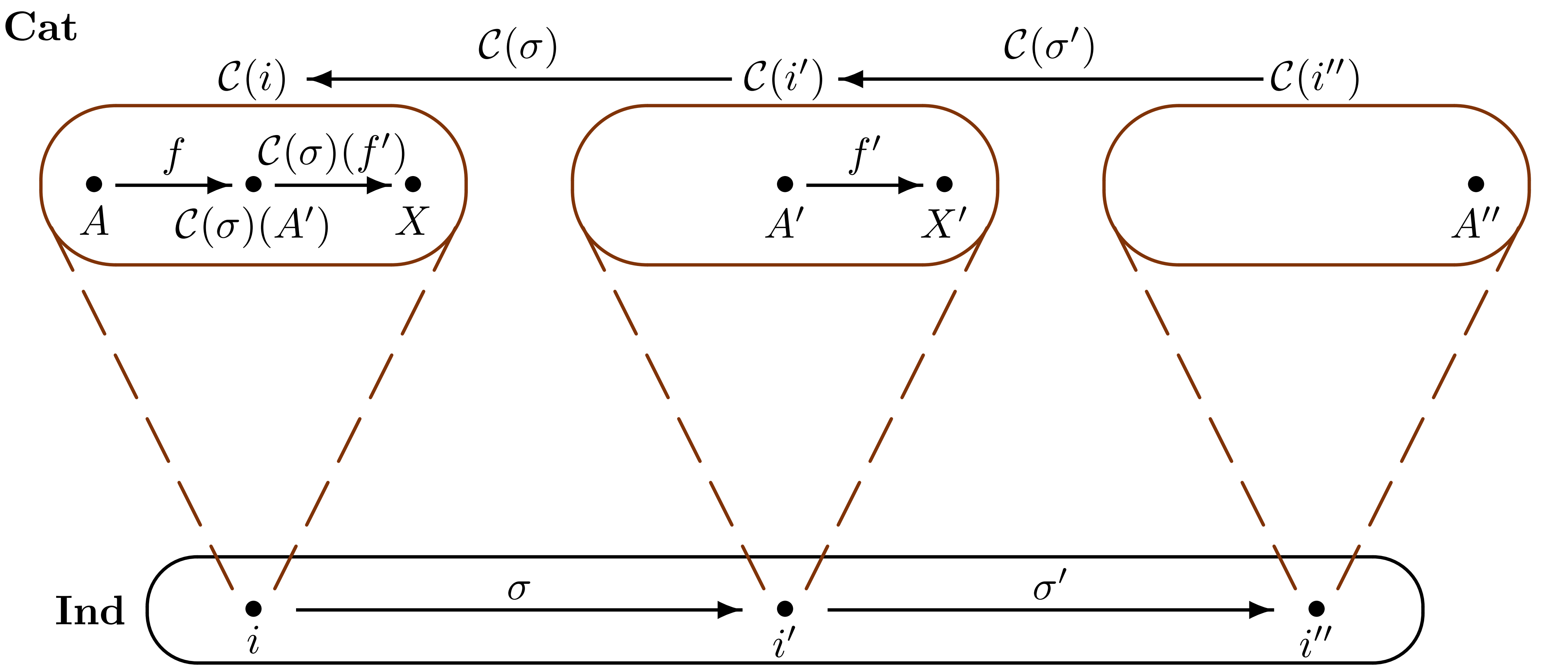
- objects: $\langle i, A\rangle$ for all $i \in |\mathbf{Ind}|$, $A \in |\mathcal{C}(i)|$
- morphisms: a morphism from $\langle i, A\rangle$ to $\langle i', A'\rangle$, $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$, consists of a morphism $\sigma\colon i \to j$ in $\mathbf{Ind}$ and a morphism $f\colon A \to \mathcal{C}(\sigma)(A')$ in $\mathcal{C}(i)$


Cat
$\mathcal{C}(\sigma)$
$\mathcal{C}(i)$
$\mathcal{C}(i')$
f
A
$\mathcal{C}(\sigma)(A')$
A'
Ind
σ
i
i'

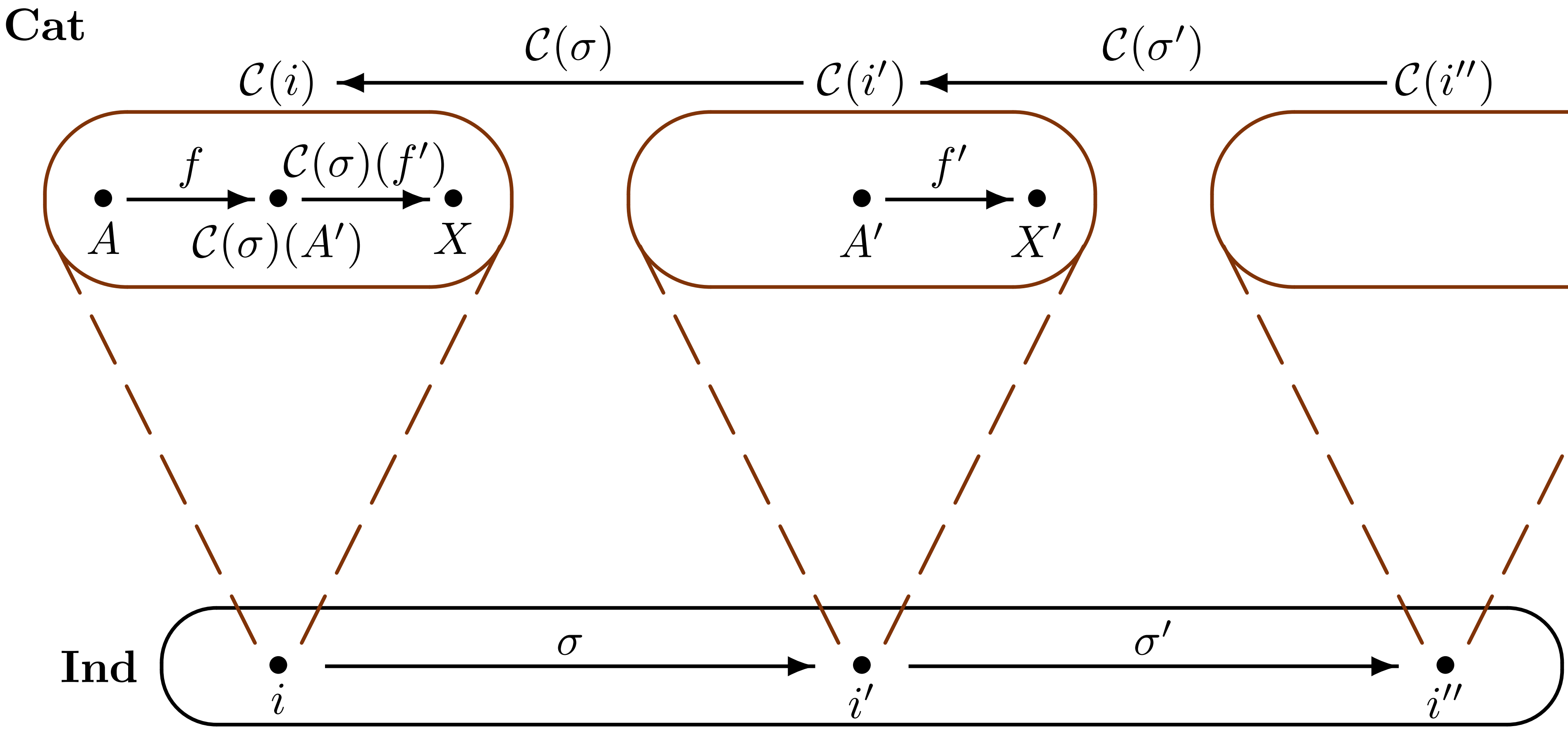

Cat
C(σ)
C(i)
C(i′)
C(i″)
f
A
C(σ)(A′)
A′
A″
Ind
σ
i
i′
i″



where $X' = \mathcal{C}(\sigma')(A'')$



where $X' = \mathcal{C}(\sigma')(A'')$ and $X = \mathcal{C}(\sigma)(X') = \mathcal{C}(\sigma)(\mathcal{C}(\sigma')(A''))$.



where $X' = \mathcal{C}(\sigma')(A'')$ and $X = \mathcal{C}(\sigma)(X') = \mathcal{C}(\sigma)(\mathcal{C}(\sigma')(A''))$.

This works fine, since $\mathcal{C}(\sigma;\sigma') = \mathcal{C}(\sigma');\mathcal{C}(\sigma)$, and so:

$X = \mathcal{C}(\sigma)(\mathcal{C}(\sigma')(A'')) = \mathcal{C}(\sigma;\sigma')(A'')$, and so $f;\mathcal{C}(\sigma)(f') \colon A \to \mathcal{C}(\sigma;\sigma')(A'')$.

Indexed categories

An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

- objects: $\langle i, A\rangle$ for all $i \in |\mathbf{Ind}|$, $A \in |\mathcal{C}(i)|$
- morphisms: a morphism from $\langle i, A\rangle$ to $\langle i', A'\rangle$, $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$, consists of a morphism $\sigma\colon i \to j$ in $\mathbf{Ind}$ and a morphism $f\colon A \to \mathcal{C}(\sigma)(A')$ in $\mathcal{C}(i)$
- composition: given $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$ and $\langle \sigma', f'\rangle\colon \langle i', A'\rangle \to \langle i'', A''\rangle$, their composition in $\mathbf{Flat}(\mathcal{C})$, $\langle \sigma, f\rangle;\langle \sigma', f'\rangle\colon \langle i, A\rangle \to \langle i'', A''\rangle$, is given by

$$\langle \sigma, f\rangle;\langle \sigma', f'\rangle = \langle \sigma;\sigma', f;\mathcal{C}(\sigma)(f')\rangle$$

Indexed categories

An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

- objects: $\langle i, A\rangle$ for all $i \in |\mathbf{Ind}|$, $A \in |\mathcal{C}(i)|$
- morphisms: a morphism from $\langle i, A\rangle$ to $\langle i', A'\rangle$, $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$, consists of a morphism $\sigma\colon i \to j$ in $\mathbf{Ind}$ and a morphism $f\colon A \to \mathcal{C}(\sigma)(A')$ in $\mathcal{C}(i)$
- composition: given $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$ and $\langle \sigma', f'\rangle\colon \langle i', A'\rangle \to \langle i'', A''\rangle$, their composition in $\mathbf{Flat}(\mathcal{C})$, $\langle \sigma, f\rangle;\langle \sigma', f'\rangle\colon \langle i, A\rangle \to \langle i'', A''\rangle$, is given by

$$\langle \sigma, f\rangle;\langle \sigma', f'\rangle = \langle \sigma;\sigma', f;\mathcal{C}(\sigma)(f')\rangle$$

Theorem: *If $\mathbf{Ind}$ is complete, $\mathcal{C}(i)$ are complete for all $i \in |\mathbf{Ind}|$, and $\mathcal{C}(\sigma)$ are continuous for all $\sigma\colon i \to j$ in $\mathbf{Ind}$, then $\mathbf{Flat}(\mathcal{C})$ is complete.*

Indexed categories

An *indexed category* is a functor $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$.

Standard example: $\mathbf{Alg}\colon \mathbf{AlgSig}^{op} \to \mathbf{Cat}$

The Grothendieck construction: Given $\mathcal{C}\colon \mathbf{Ind}^{op} \to \mathbf{Cat}$, define a category $\mathbf{Flat}(\mathcal{C})$:

- objects: $\langle i, A\rangle$ for all $i \in |\mathbf{Ind}|$, $A \in |\mathcal{C}(i)|$
- morphisms: a morphism from $\langle i, A\rangle$ to $\langle i', A'\rangle$, $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$, consists of a morphism $\sigma\colon i \to j$ in $\mathbf{Ind}$ and a morphism $f\colon A \to \mathcal{C}(\sigma)(A')$ in $\mathcal{C}(i)$
- composition: given $\langle \sigma, f\rangle\colon \langle i, A\rangle \to \langle i', A'\rangle$ and $\langle \sigma', f'\rangle\colon \langle i', A'\rangle \to \langle i'', A''\rangle$, their composition in $\mathbf{Flat}(\mathcal{C})$, $\langle \sigma, f\rangle;\langle \sigma', f'\rangle\colon \langle i, A\rangle \to \langle i'', A''\rangle$, is given by

$$\langle \sigma, f\rangle;\langle \sigma', f'\rangle = \langle \sigma;\sigma', f;\mathcal{C}(\sigma)(f')\rangle$$

Theorem: *If $\mathbf{Ind}$ is complete, $\mathcal{C}(i)$ are complete for all $i \in |\mathbf{Ind}|$, and $\mathcal{C}(\sigma)$ are continuous for all $\sigma\colon i \to j$ in $\mathbf{Ind}$, then $\mathbf{Flat}(\mathcal{C})$ is complete.*

Try to formulate and prove a theorem concerning cocompleteness of $\mathbf{Flat}(\mathcal{C})$