Łoś Ultraproduct Theorem

Anthony Wilkie

May 2, 2022

Introduction

- Jerzy Łoś (1920-1998) was a Polish mathematician, economist, logician, and philosopher.

Łoś Ultraproduct Theorem

The ultraproduct of models is itself a model.

- This has been applied (by Micheal Makkai and more recently Jacob Lurie) to recover the syntax of a theory by using categories with an "ultraproduct" structure.

Introduction

- Jerzy Łoś (1920-1998) was a Polish mathematician, economist, logician, and philosopher.

Łoś Ultraproduct Theorem

The ultraproduct of models is itself a model.

- This has been applied (by Micheal Makkai and more recently Jacob Lurie) to recover the syntax of a theory by using categories with an "ultraproduct" structure.

Introduction

- Jerzy Łoś (1920-1998) was a Polish mathematician, economist, logician, and philosopher.

Łoś Ultraproduct Theorem

The ultraproduct of models is itself a model.

- This has been applied (by Micheal Makkai and more recently Jacob Lurie) to recover the syntax of a theory by using categories with an "ultraproduct" structure.

Language and Terms

- A (single sorted, first-order) *language* $\mathcal{L}$ is a set of symbols divided into three groups:
  - *Relation symbols* R which has a specified arity n,
  - *Function symbols* f which has a specified arity n,
  - *Constant symbols* c,

  together with logical symbols $($, $)$, $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$, $\top$, $\bot$, $\Rightarrow$, $=$ and an infinite collection of variables $x_1, x_2, \ldots$.
- The *terms* over $\mathcal{L}$ are defined inductively as follows:
  - A variable x is a term.
  - If $t_1, \ldots, t_n$ are terms and f is a function symbol of $\mathcal{L}$ of arity n, then $f(t_1, \ldots, t_n)$ is a term.
- We write $t(\overline{x})$ to mean t is a term in which the variables $\overline{x} = x_1, \ldots, x_n$ appear in t.

Language and Terms

- A (single sorted, first-order) *language* $\mathcal{L}$ is a set of symbols divided into three groups:
  - *Relation symbols* R which has a specified arity n,
  - *Function symbols* f which has a specified arity n,
  - *Constant symbols* c,

  together with logical symbols $($, $)$, $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$, $\top$, $\bot$, $\Rightarrow$, $=$ and an infinite collection of variables $x_1, x_2, \ldots$.
- The *terms* over $\mathcal{L}$ are defined inductively as follows:
  - A variable x is a term.
  - If $t_1, \ldots, t_n$ are terms and f is a function symbol of $\mathcal{L}$ of arity n, then $f(t_1, \ldots, t_n)$ is a term.
- We write $t(\overline{x})$ to mean t is a term in which the variables $\overline{x} = x_1, \ldots, x_n$ appear in t.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Formulae

- The *formulae* ϕ over $\mathcal{L}$ with *free variables* $FV(\phi)$ are defined inductively as follows (Note: the free variables of a term $t(\overline{x})$ is $\overline{x} = x_1, \ldots, x_n$):
  - If $t_1(\overline{x})$ and $t_2(\overline{x})$ are terms, then $t_1(\overline{x}) = t_2(\overline{x})$ is a formula with free variables $FV(t_1 = t_2) = \overline{x}$.
  - If $t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n)$ are terms and R is a relation symbol of arity n, then $R(t_1(\overline{x}_1), \ldots, t_n(\overline{x}_n))$ is a formula with $FV(R) = \{\overline{x}_1, \ldots, \overline{x}_n\}$.
  - If $\phi(\overline{x})$ and $\psi(\overline{x})$ are formulae, then $(\phi \vee \psi)(\overline{x})$ is a formula with $FV(\phi \vee \psi) = FV(\phi) \cup FV(\psi)$.
  - If $\phi(\overline{x})$ is a formula, then so is $\neg\phi(\overline{x})$ with $FV(\neg\phi) = FV(\phi)$.
  - If $\psi(\overline{x}, y)$ is a formula and y is a variable, then $\exists y.\psi(\overline{x}, y)$ is a formula with with free variables $FV(\psi) \setminus \{y\}$.

Theory

- A variable x is called *bound* if it is in the scope of a quantifier, i.e. if $\exists x$ or $\forall x$.
- A formula $\sigma(\overline{x})$ is called a *sentence* if $FV(\sigma) = \emptyset$, that is, all of its variables are bound.
- We will use the common abbreviations for the following:
  - $\phi(\overline{x}) \wedge \psi(\overline{x}) \stackrel{\text{def}}{=} \neg((\neg\phi(\overline{x})) \vee (\neg\psi(\overline{x})))$.
  - $\phi(\overline{x}) \Rightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\neg\phi(\overline{x})) \vee \psi(\overline{x})$.
  - $\phi(\overline{x}) \Leftrightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\phi(\overline{x}) \Rightarrow \psi(\overline{x})) \wedge (\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.
  - $\forall y.\phi(\overline{x}, y) \stackrel{\text{def}}{=} \neg\exists y.\neg\phi(\overline{x}, y)$.
  - $\top \stackrel{\text{def}}{=} \forall x.(x = x)$.
  - $\bot \stackrel{\text{def}}{=} \exists x.(x \neq x)$.
- A *theory* T is a set of sentences which we call the *axioms* of T.

Theory

- A variable x is called *bound* if it is in the scope of a quantifier, i.e. if $\exists x$ or $\forall x$.
- A formula $\sigma(\overline{x})$ is called a *sentence* if $FV(\sigma) = \emptyset$, that is, all of its variables are bound.
- We will use the common abbreviations for the following:
  - $\phi(\overline{x}) \wedge \psi(\overline{x}) \stackrel{\text{def}}{=} \neg((\neg\phi(\overline{x})) \vee (\neg\psi(\overline{x})))$.
  - $\phi(\overline{x}) \Rightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\neg\phi(\overline{x})) \vee \psi(\overline{x})$.
  - $\phi(\overline{x}) \Leftrightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\phi(\overline{x}) \Rightarrow \psi(\overline{x})) \wedge (\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.
  - $\forall y.\phi(\overline{x}, y) \stackrel{\text{def}}{=} \neg\exists y.\neg\phi(\overline{x}, y)$.
  - $\top \stackrel{\text{def}}{=} \forall x.(x = x)$.
  - $\bot \stackrel{\text{def}}{=} \exists x.(x \neq x)$.
- A *theory* T is a set of sentences which we call the *axioms* of T.

Theory

- A variable x is called *bound* if it is in the scope of a quantifier, i.e. if $\exists x$ or $\forall x$.
- A formula $\sigma(\overline{x})$ is called a *sentence* if $FV(\sigma) = \emptyset$, that is, all of its variables are bound.
- We will use the common abbreviations for the following:
  - $\phi(\overline{x}) \wedge \psi(\overline{x}) \stackrel{\text{def}}{=} \neg((\neg\phi(\overline{x})) \vee (\neg\psi(\overline{x})))$.
  - $\phi(\overline{x}) \Rightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\neg\phi(\overline{x})) \vee \psi(\overline{x})$.
  - $\phi(\overline{x}) \Leftrightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\phi(\overline{x}) \Rightarrow \psi(\overline{x})) \wedge (\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.
  - $\forall y.\phi(\overline{x}, y) \stackrel{\text{def}}{=} \neg\exists y.\neg\phi(\overline{x}, y)$.
  - $\top \stackrel{\text{def}}{=} \forall x.(x = x)$.
  - $\bot \stackrel{\text{def}}{=} \exists x.(x \neq x)$.
- A *theory* T is a set of sentences which we call the *axioms* of T.

Theory

- A variable x is called *bound* if it is in the scope of a quantifier, i.e. if $\exists x$ or $\forall x$.
- A formula $\sigma(\overline{x})$ is called a *sentence* if $FV(\sigma) = \emptyset$, that is, all of its variables are bound.
- We will use the common abbreviations for the following:
  - $\phi(\overline{x}) \wedge \psi(\overline{x}) \stackrel{\text{def}}{=} \neg((\neg\phi(\overline{x})) \vee (\neg\psi(\overline{x})))$.
  - $\phi(\overline{x}) \Rightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\neg\phi(\overline{x})) \vee \psi(\overline{x})$.
  - $\phi(\overline{x}) \Leftrightarrow \psi(\overline{x}) \stackrel{\text{def}}{=} (\phi(\overline{x}) \Rightarrow \psi(\overline{x})) \wedge (\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.
  - $\forall y.\phi(\overline{x}, y) \stackrel{\text{def}}{=} \neg\exists y.\neg\phi(\overline{x}, y)$.
  - $\top \stackrel{\text{def}}{=} \forall x.(x = x)$.
  - $\bot \stackrel{\text{def}}{=} \exists x.(x \neq x)$.
- A *theory* T is a set of sentences which we call the *axioms* of T.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

Example of Theory

- We define the *theory of partially ordered sets (posets)* as follows:
  - $\mathcal{L} = \{\leq\}$, where $\leq$ is a 2-ary relation symbol.
  - The theory consists of the following axioms:
    (i) *Reflexivity*: $\forall x.(x \leq x)$.
    (ii) *Asymmetry*: $\forall x, y.(x \leq y \wedge y \leq x \Rightarrow x = y)$.
    (iii) *Transitivity*: $\forall x, y, z.(x \leq y \wedge y \leq z \Rightarrow x \leq z)$.
  - We obtain the *theory of directed posets* if we add the following axiom:
    (iv) *Directed*: $\forall x, y.\exists z.(x \leq z \wedge y \leq z)$.
- We can define the theory of groups, rings, fields, number theory, etc.
- We CANNOT define the theory of topological spaces.

$\mathcal{L}$-Structure

- We define a *$\mathcal{L}$-structure* to be a function from our language to a *set* M such that:
  - For each relation symbol R of arity n, we have an *interpretation* of R as a subset $M(R) \subseteq M^n$.
  - For each function symbol f of arity n, we have an *interpretation* of f as a function $M(f) : M^n \to M$.
  - For each constant symbol c, we have an *interpretation* of c as an element $M(c) \in M$.

Semantics

- The *value* of a term $t(\overline{x})$ at $\overline{a} = a_1, \ldots, a_n$ in a $\mathcal{L}$-structure M is defined as follows:
  - If $t = x_i$ is a variable, then $t(\overline{a}) = a_i$.
  - If $t = f(t_1, \ldots, t_n)$, f a function symbol and t_i are terms, then $t(\overline{a}) = M(f)(t_1(\overline{a}), \ldots, t_n(\overline{a}))$.
- Given a $\mathcal{L}$-structure M, a formula $\phi(\overline{x})$, and a tuple of element $\overline{a} = a_1, \ldots, a_n \in M$, we define M *satisfies* ϕ with $\overline{a}$, denoted $M \vDash \phi(\overline{a})$, as follows:
  - If $\phi(\overline{x}) := t_1(\overline{x}) = t_2(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff $t_1(\overline{a}) = t_2(\overline{a})$.
  - If $\phi(\overline{x}) := R(t_1, \ldots, t_n)$, then $M \vDash \phi(\overline{a})$ iff $t_1(\overline{a}), \ldots, t_n(\overline{a}) \in M(R)$.
  - If $\phi(\overline{x}) := (\psi_1 \vee \psi_2)(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff $M \vDash \psi_1(\overline{a})$ or $M \vDash \psi_2(\overline{a})$.
  - If $\phi(\overline{x}) := (\neg\psi)(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff not $M \vDash \psi(\overline{a})$.
  - If $\phi(\overline{x}) := \exists y.\psi(\overline{x}, y)$, then $M \vDash \phi(\overline{a})$ iff there exists $b \in M$ such that $M \vDash \psi(\overline{a}, b)$.

Semantics

- The *value* of a term $t(\overline{x})$ at $\overline{a} = a_1, \ldots, a_n$ in a $\mathcal{L}$-structure M is defined as follows:
  - If $t = x_i$ is a variable, then $t(\overline{a}) = a_i$.
  - If $t = f(t_1, \ldots, t_n)$, f a function symbol and t_i are terms, then $t(\overline{a}) = M(f)(t_1(\overline{a}), \ldots, t_n(\overline{a}))$.
- Given a $\mathcal{L}$-structure M, a formula $\phi(\overline{x})$, and a tuple of element $\overline{a} = a_1, \ldots, a_n \in M$, we define M *satisfies* ϕ with $\overline{a}$, denoted $M \vDash \phi(\overline{a})$, as follows:
  - If $\phi(\overline{x}) := t_1(\overline{x}) = t_2(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff $t_1(\overline{a}) = t_2(\overline{a})$.
  - If $\phi(\overline{x}) := R(t_1, \ldots, t_n)$, then $M \vDash \phi(\overline{a})$ iff $t_1(\overline{a}), \ldots, t_n(\overline{a}) \in M(R)$.
  - If $\phi(\overline{x}) := (\psi_1 \vee \psi_2)(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff $M \vDash \psi_1(\overline{a})$ or $M \vDash \psi_2(\overline{a})$.
  - If $\phi(\overline{x}) := (\neg\psi)(\overline{x})$, then $M \vDash \phi(\overline{a})$ iff not $M \vDash \psi(\overline{a})$.
  - If $\phi(\overline{x}) := \exists y.\psi(\overline{x}, y)$, then $M \vDash \phi(\overline{a})$ iff there exists $b \in M$ such that $M \vDash \psi(\overline{a}, b)$.

- If $T = \{\sigma_i\}_{i \in I}$ is a theory with axioms σ_i, then a $\mathcal{L}$-structure M is a *model* of T if $M \vDash T$, that is, $M \vDash \sigma_i$ for each $\sigma_i \in T$.
- A poset $(P, \leq_P)$ is a model of the theory of posets with

$$x \leq_P y \Leftrightarrow (x, y) \in \leq_P \ (\subseteq P \times P).$$

- Let M and N be models of a theory T. A function $f : M \to N$ is an *elementary embedding* if for every formula $\phi(\overline{x})$ and $\overline{a} \in M^n$, we have

$$M \vDash \phi(\overline{a}) \Leftrightarrow N \vDash \phi(f(\overline{a})).$$

- An elementary embedding between posets is a monotone function.

Models

- If $T = \{\sigma_i\}_{i \in I}$ is a theory with axioms σ_i, then a $\mathcal{L}$-structure M is a *model* of T if $M \vDash T$, that is, $M \vDash \sigma_i$ for each $\sigma_i \in T$.
- A poset $(P, \leq_P)$ is a model of the theory of posets with

$$x \leq_P y \Leftrightarrow (x, y) \in \leq_P \ (\subseteq P \times P).$$

- Let M and N be models of a theory T. A function $f : M \to N$ is an *elementary embedding* if for every formula $\phi(\overline{x})$ and $\overline{a} \in M^n$, we have

$$M \vDash \phi(\overline{a}) \Leftrightarrow N \vDash \phi(f(\overline{a})).$$

- An elementary embedding between posets is a monotone function.

Models

- If $T = \{\sigma_i\}_{i \in I}$ is a theory with axioms σ_i, then a $\mathcal{L}$-structure M is a *model* of T if $M \vDash T$, that is, $M \vDash \sigma_i$ for each $\sigma_i \in T$.
- A poset $(P, \leq_P)$ is a model of the theory of posets with

$$x \leq_P y \Leftrightarrow (x, y) \in \leq_P \ (\subseteq P \times P).$$

- Let M and N be models of a theory T. A function $f : M \to N$ is an *elementary embedding* if for every formula $\phi(\overline{x})$ and $\overline{a} \in M^n$, we have

$$M \vDash \phi(\overline{a}) \Leftrightarrow N \vDash \phi(f(\overline{a})).$$

- An elementary embedding between posets is a monotone function.

Models

- If $T = \{\sigma_i\}_{i \in I}$ is a theory with axioms σ_i, then a $\mathcal{L}$-structure M is a *model* of T if $M \vDash T$, that is, $M \vDash \sigma_i$ for each $\sigma_i \in T$.
- A poset $(P, \leq_P)$ is a model of the theory of posets with

$$x \leq_P y \Leftrightarrow (x, y) \in \leq_P \ (\subseteq P \times P).$$

- Let M and N be models of a theory T. A function $f : M \to N$ is an *elementary embedding* if for every formula $\phi(\overline{x})$ and $\overline{a} \in M^n$, we have

$$M \vDash \phi(\overline{a}) \Leftrightarrow N \vDash \phi(f(\overline{a})).$$

- An elementary embedding between posets is a monotone function.

Ultrafilters

- An *ultrafilter* $\mathcal{U}$ on a set S is a subset of the powerset $\mathcal{P}(S)$ such that
  - $S \in \mathcal{U}$.
  - For each $S_0, S_1 \in \mathcal{U}$, we have $S_0 \cap S_1 \in \mathcal{U}$.
  - If $S_0 \in \mathcal{U}$ and $S_1 \subseteq S_0$ then $S_1 \in \mathcal{U}$.
  - For every $S_0 \subseteq S$, exactly one of S_0 and $S \setminus S_0$ is in $\mathcal{U}$.
- If $s \in S$, then we have the *principal ultrafilter*

$$\mathcal{U}_s = \{S_0 \subseteq S : s \in S_0\}.$$

Ultrafilters

- An *ultrafilter* $\mathcal{U}$ on a set S is a subset of the powerset $\mathcal{P}(S)$ such that
  - $S \in \mathcal{U}$.
  - For each $S_0, S_1 \in \mathcal{U}$, we have $S_0 \cap S_1 \in \mathcal{U}$.
  - If $S_0 \in \mathcal{U}$ and $S_1 \subseteq S_0$ then $S_1 \in \mathcal{U}$.
  - For every $S_0 \subseteq S$, exactly one of S_0 and $S \setminus S_0$ is in $\mathcal{U}$.
- If $s \in S$, then we have the *principal ultrafilter*

$$\mathcal{U}_s = \{S_0 \subseteq S : s \in S_0\}.$$

- Given a collection of (nonempty) sets $\{X_s\}_{s\in S}$ and an ultrafilter $\mathcal{U}$ on S, we define an equivalence relation $\sim_{\mathcal{U}}$ on $\prod_{s\in S} X_s$ by

$$\left(\{x_s\}_{s\in S} \sim_{\mathcal{U}} \{y_s\}_{s\in S}\right) \Leftrightarrow \{s\in S : x_s = y_s\} \in \mathcal{U}.$$

  We let $x^{\mathcal{U}}$ denote the equivalence class of $\{x_s\}_{s\in S}$.
- Then the *ultraproduct of* $\{X_s\}_{s\in S}$ *w.r.t* $\mathcal{U}$, denoted

$$\left(\prod_{s\in S} X_s\right)/\mathcal{U} = \left\{x^{\mathcal{U}} : \{x_s\}_{s\in S} \in \prod_{s\in S} X_s\right\},$$

  is the set of equivalence classes of the relation $\sim_{\mathcal{U}}$.

- Given a collection of (nonempty) sets $\{X_s\}_{s\in S}$ and an ultrafilter $\mathcal{U}$ on S, we define an equivalence relation $\sim_{\mathcal{U}}$ on $\prod_{s\in S} X_s$ by

$$\left(\{x_s\}_{s\in S} \sim_{\mathcal{U}} \{y_s\}_{s\in S}\right) \Leftrightarrow \{s \in S : x_s = y_s\} \in \mathcal{U}.$$

  We let $x^{\mathcal{U}}$ denote the equivalence class of $\{x_s\}_{s\in S}$.
- Then the *ultraproduct of* $\{X_s\}_{s\in S}$ *w.r.t* $\mathcal{U}$, denoted

$$\left(\prod_{s\in S} X_s\right)/\mathcal{U} = \left\{x^{\mathcal{U}} : \{x_s\}_{s\in S} \in \prod_{s\in S} X_s\right\},$$

  is the set of equivalence classes of the relation $\sim_{\mathcal{U}}$.

Ultraproducts

Let T be a theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}},\dots,x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}},\dots,x_n^{\mathcal{U}}) \Leftrightarrow \{s\in S : M_s(R)(x_{s,1},\dots,x_{s,n})\}\in\mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}},\dots,x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}},\dots,x_n^{\mathcal{U}}) = \left(M_s(f)(x_{s,1},\dots,x_{s,n}) : s\in S\right)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(c) = \left(M_s(c) : s\in S\right)^{\mathcal{U}} \in \left(\prod_{s\in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s,\leq_{P_s})\}_{s\in S}$ are posets

$$\{x_s\}_{s\in S} \leq_{UP} \{y_s\}_{s\in S} \Leftrightarrow \{s\in S : x_s \leq_{P_s} y_s\}\in\mathcal{U}.$$

Ultraproducts

Let T be a theory and let $\{M_s\}_{s \in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s \in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s \in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s \in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) \Leftrightarrow \{s \in S : M_s(R)(x_{s,1}, \ldots, x_{s,n})\} \in \mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s \in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) = (M_s(f)(x_{s,1}, \ldots, x_{s,n}) : s \in S)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s \in S} M_s\right)/\mathcal{U}\right)(c) = (M_s(c) : s \in S)^{\mathcal{U}} \in \left(\prod_{s \in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s, \leq_{P_s})\}_{s \in S}$ are posets

$$\{x_s\}_{s \in S} \leq_{UP} \{y_s\}_{s \in S} \Leftrightarrow \{s \in S : x_s \leq_{P_s} y_s\} \in \mathcal{U}.$$

Let T be a theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) \Leftrightarrow \{s \in S : M_s(R)(x_{s,1}, \ldots, x_{s,n})\} \in \mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) = (M_s(f)(x_{s,1}, \ldots, x_{s,n}) : s \in S)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(c) = (M_s(c) : s \in S)^{\mathcal{U}} \in \left(\prod_{s\in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s, \leq_{P_s})\}_{s\in S}$ are posets

$$\{x_s\}_{s\in S} \leq_{UP} \{y_s\}_{s\in S} \Leftrightarrow \{s \in S : x_s \leq_{P_s} y_s\} \in \mathcal{U}.$$

Ultraproducts

Let T be a theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) \Leftrightarrow \{s \in S : M_s(R)(x_{s,1}, \ldots, x_{s,n})\} \in \mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) = (M_s(f)(x_{s,1}, \ldots, x_{s,n}) : s \in S)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(c) = (M_s(c) : s \in S)^{\mathcal{U}} \in \left(\prod_{s\in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s, \leq_{P_s})\}_{s\in S}$ are posets

$$\{x_s\}_{s\in S} \leq_{UP} \{y_s\}_{s\in S} \Leftrightarrow \{s \in S : x_s \leq_{P_s} y_s\} \in \mathcal{U}.$$

Ultraproducts

Let T be a theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}},\ldots,x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}},\ldots,x_n^{\mathcal{U}}) \Leftrightarrow \{s\in S : M_s(R)(x_{s,1},\ldots,x_{s,n})\}\in\mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}},\ldots,x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}},\ldots,x_n^{\mathcal{U}}) = (M_s(f)(x_{s,1},\ldots,x_{s,n}) : s\in S)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(c) = (M_s(c) : s\in S)^{\mathcal{U}} \in \left(\prod_{s\in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s,\le_{P_s})\}_{s\in S}$ are posets

$$\{x_s\}_{s\in S} \le_{UP} \{y_s\}_{s\in S} \Leftrightarrow \{s\in S : x_s \le_{P_s} y_s\}\in\mathcal{U}.$$

Ultraproducts

Let T be a theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Let $\mathcal{U}$ be an ultrafilter on S. Then the *ultraproduct of the models* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$ is the $\mathcal{L}$-structure $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ defined by:

- For each relation symbol R of arity n and each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(R)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) \Leftrightarrow \{s \in S : M_s(R)(x_{s,1}, \ldots, x_{s,n})\} \in \mathcal{U}.$$

- For each function symbol f of arity n each tuple of equivalence classes $x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}$,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(f)(x_1^{\mathcal{U}}, \ldots, x_n^{\mathcal{U}}) = (M_s(f)(x_{s,1}, \ldots, x_{s,n}) : s \in S)^{\mathcal{U}}$$

- For each constant symbol c,

$$\left(\left(\prod_{s\in S} M_s\right)/\mathcal{U}\right)(c) = (M_s(c) : s \in S)^{\mathcal{U}} \in \left(\prod_{s\in S} M_s\right)/\mathcal{U}.$$

- If $\{(P_s, \leq_{P_s})\}_{s\in S}$ are posets

$$\{x_s\}_{s\in S} \leq_{UP} \{y_s\}_{s\in S} \Leftrightarrow \{s \in S : x_s \leq_{P_s} y_s\} \in \mathcal{U}.$$

Łoś Ultraproduct Theorem

Let $\{M_s\}_{s\in S}$ be a collection of models of a theory T and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ is also a model of T.

- The ultraproduct of a collection of posets (groups, rings, etc.) is a poset (group, ring, etc.).
- We will prove a more general version of Łoś Ultraproduct Theorem using Category Theory.

Łoś Ultraproduct Theorem

Łoś Ultraproduct Theorem

Let $\{M_s\}_{s\in S}$ be a collection of models of a theory T and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ is also a model of T.

- The ultraproduct of a collection of posets (groups, rings, etc.) is a poset (group, ring, etc.).
- We will prove a more general version of Łoś Ultraproduct Theorem using Category Theory.

Łoś Ultraproduct Theorem

Łoś Ultraproduct Theorem

Let $\{M_s\}_{s\in S}$ be a collection of models of a theory T and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct $\left(\prod_{s\in S} M_s\right)/\mathcal{U}$ is also a model of T.

- The ultraproduct of a collection of posets (groups, rings, etc.) is a poset (group, ring, etc.).
- We will prove a more general version of Łoś Ultraproduct Theorem using Category Theory.

Set Functions

- We can define functions between sets, i.e. $f : X \to Y$, where $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For any three sets X, Y, Z and functions $f : X \to Y$, $g : Y \to Z$ between them, we are able to compose f and g to create a new function

$$g \circ f : X \to Z.$$

- This composition is *associative*, that is, for any three functions $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$ we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- For any set X, we have the identity function $\mathrm{id}_X : X \to X$.
- This identity function is *unital*, i.e. has the property that for any $f : X \to Y$, we have

$$f \circ \mathrm{id}_X = f = \mathrm{id}_Y \circ f.$$

Set Functions

- We can define functions between sets, i.e. $f : X \to Y$, where $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For any three sets X, Y, Z and functions $f : X \to Y$, $g : Y \to Z$ between them, we are able to compose f and g to create a new function

$$g \circ f : X \to Z.$$

- This composition is *associative*, that is, for any three functions $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$ we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- For any set X, we have the identity function $\mathrm{id}_X : X \to X$.
- This identity function is *unital*, i.e. has the property that for any $f : X \to Y$, we have

$$f \circ \mathrm{id}_X = f = \mathrm{id}_Y \circ f.$$

Set Functions

- We can define functions between sets, i.e. $f : X \to Y$, where $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For any three sets X, Y, Z and functions $f : X \to Y$, $g : Y \to Z$ between them, we are able to compose f and g to create a new function

$$g \circ f : X \to Z.$$

- This composition is *associative*, that is, for any three functions $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$ we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- For any set X, we have the identity function $\mathrm{id}_X : X \to X$.
- This identity function is *unital*, i.e. has the property that for any $f : X \to Y$, we have

$$f \circ \mathrm{id}_X = f = \mathrm{id}_Y \circ f.$$

Set Functions

- We can define functions between sets, i.e. $f : X \to Y$, where $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For any three sets X, Y, Z and functions $f : X \to Y$, $g : Y \to Z$ between them, we are able to compose f and g to create a new function

$$g \circ f : X \to Z.$$

- This composition is *associative*, that is, for any three functions $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$ we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- For any set X, we have the identity function $\mathrm{id}_X : X \to X$.
- This identity function is *unital*, i.e. has the property that for any $f : X \to Y$, we have

$$f \circ \mathrm{id}_X = f = \mathrm{id}_Y \circ f.$$

Set Functions

- We can define functions between sets, i.e. $f : X \to Y$, where $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For any three sets X, Y, Z and functions $f : X \to Y$, $g : Y \to Z$ between them, we are able to compose f and g to create a new function

$$g \circ f : X \to Z.$$

- This composition is *associative*, that is, for any three functions $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$ we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

- For any set X, we have the identity function $\mathrm{id}_X : X \to X$.
- This identity function is *unital*, i.e. has the property that for any $f : X \to Y$, we have

$$f \circ \mathrm{id}_X = f = \mathrm{id}_Y \circ f.$$

Category

A *category* C consists of the following data:

- a collection C_0 of *objects:* $X, Y, Z, \ldots$
- a collection C_1 of *morphisms* (or *arrows*): $f, g, h, \ldots$

such that

- Each $f \in \mathsf{C}_1$ has a *domain* and a *codomain* object, with $f : X \to Y$ the notation used to denote that $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For each $X \in \mathsf{C}_0$, there is an *identity morphism* $\mathrm{id}_X : X \to X$.
- For each pair $f, g \in \mathsf{C}_1$ such that $\mathrm{dom}(g) = \mathrm{cod}(f)$, there exist a *composite morphism* $g \circ f \in \mathsf{C}_1$ with $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ and $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$.

This data is required to satisfy the following axioms:

- For any $f : X \to Y$, we have $\mathrm{id}_Y \circ f = f = f \circ \mathrm{id}_X$, that is, composition is *unital*.
- For any composable triple $f, g, h \in \mathsf{C}_1$, we have that $h \circ (g \circ f) = (h \circ g) \circ f$, that is, composition is *associative*.

Category

A *category* C consists of the following data:

- a collection C_0 of *objects:* $X, Y, Z, \ldots$
- a collection C_1 of *morphisms* (or *arrows*): $f, g, h, \ldots$

such that

- Each $f \in \mathsf{C}_1$ has a *domain* and a *codomain* object, with $f : X \to Y$ the notation used to denote that $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For each $X \in \mathsf{C}_0$, there is an *identity morphism* $\mathrm{id}_X : X \to X$.
- For each pair $f, g \in \mathsf{C}_1$ such that $\mathrm{dom}(g) = \mathrm{cod}(f)$, there exist a *composite morphism* $g \circ f \in \mathsf{C}_1$ with $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ and $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$.

This data is required to satisfy the following axioms:

- For any $f : X \to Y$, we have $\mathrm{id}_Y \circ f = f = f \circ \mathrm{id}_X$, that is, composition is *unital*.
- For any composable triple $f, g, h \in \mathsf{C}_1$, we have that $h \circ (g \circ f) = (h \circ g) \circ f$, that is, composition is *associative*.

A *category* C consists of the following data:

- a collection C_0 of *objects:* $X, Y, Z, \ldots$
- a collection C_1 of *morphisms* (or *arrows*): $f, g, h, \ldots$

such that

- Each $f \in \mathsf{C}_1$ has a *domain* and a *codomain* object, with $f : X \to Y$ the notation used to denote that $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$.
- For each $X \in \mathsf{C}_0$, there is an *identity morphism* $\mathrm{id}_X : X \to X$.
- For each pair $f, g \in \mathsf{C}_1$ such that $\mathrm{dom}(g) = \mathrm{cod}(f)$, there exist a *composite morphism* $g \circ f \in \mathsf{C}_1$ with $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ and $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$.

This data is required to satisfy the following axioms:

- For any $f : X \to Y$, we have $\mathrm{id}_Y \circ f = f = f \circ \mathrm{id}_X$, that is, composition is *unital*.
- For any composable triple $f, g, h \in \mathsf{C}_1$, we have that $h \circ (g \circ f) = (h \circ g) \circ f$, that is, composition is *associative*.

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$, \qquad 0, \qquad 0 \longrightarrow 1$$

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$, \qquad 0, \qquad 0 \longrightarrow 1$$

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$
\quad , \qquad
\begin{tikzcd} 0 \arrow[loop above] \end{tikzcd}, \qquad
\begin{tikzcd} 0 \arrow[loop left] \arrow[r] & 1 \arrow[loop right] \end{tikzcd}
$$

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$ \quad , \qquad 0 , \qquad 0 \longrightarrow 1 $$

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$ \quad , \qquad \begin{tikzcd} 0 \arrow[loop above] \end{tikzcd}, \qquad \begin{tikzcd} 0 \arrow[loop left] \arrow[r] & 1 \arrow[loop right] \end{tikzcd} $$

Examples

- Set: the category of sets and set functions.
- Top: the category of topological spaces and continuous functions.
- Grp: the category of groups and group homomorphisms.
- Mod(T): the category of models of a first order theory T and elementary embeddings.
- Any set X can be regarded as a category, whose objects are the elements and the only arrows are the identity arrows. This is known as a *discrete category*.
- Any poset $(P, \leq)$ can be regarded as a category, whose objects are the elements and there exists an arrow $f : x \to y$ if and only if $x \leq y$. In particular, we have categories $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ that look as follows:

$$
, \qquad \begin{tikzcd} 0 \arrow[loop above] \end{tikzcd}, \qquad \begin{tikzcd} 0 \arrow[loop left] \arrow[r] & 1 \arrow[loop right] \end{tikzcd}
$$

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

More Examples

- If C is a category, then a *subcategory* D of C consists of subcollections $\mathsf{D}_0 \subseteq \mathsf{C}_0$ and $\mathsf{D}_1 \subseteq \mathsf{C}_1$ such that:
  - If $f : X \to Y$ is in D, then $X, Y \in \mathsf{D}_0$.
  - If $f, g \in \mathsf{D}_1$, then $g \circ f \in \mathsf{D}_1$.
  - If $X \in \mathsf{D}_0$, then $\mathrm{id}_X \in \mathsf{D}_1$.
- We say D is a *full subcategory* of C if the following also holds:
  - If $X, Y \in \mathsf{D}_0$ and $f : X \to Y$ is in C, then $f \in \mathsf{D}_1$.
- We have the category of finite sets Finset is a full subcategory of Set.
- The *opposite category* C^{op} of C is the category with the same objects but $f^{\mathrm{op}} : X \to Y$ is in C^{op} iff $f : Y \to X$ is in C.
- Given two categories C, D, we can form their *product category* $\mathsf{C} \times \mathsf{D}$.
- If C is a category and S is a set, then $\mathsf{C}^S = \prod_{s \in S} \mathsf{C}$ is the *product category* of C with itself S many times.

Arrows

Let C be a category and $f : X \to Y$ be an arrow.

- We say f is an *isomorphism* if there exists a unique arrow $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$.
- We say f is a *monomorphism* (or *monic*) if for every parallel pair of arrows $g, h : Z \rightrightarrows X$,

$$Z \overset{g}{\underset{h}{\rightrightarrows}} X \overset{f}{\rightarrowtail} Y. \qquad f \circ g = f \circ h \Rightarrow g = h.$$

- We say f is an *epimorphism* (or *epic*) if for every parallel pair of arrows $g, h : Y \rightrightarrows Z$,

$$X \overset{f}{\twoheadrightarrow} Y \overset{g}{\underset{h}{\rightrightarrows}} Z. \qquad g \circ f = h \circ f \Rightarrow g = h.$$

In Set, these are the bijections, injections, and surjections.

Let $\mathcal{C}$ be a category and $f : X \to Y$ be an arrow.

- We say f is an *isomorphism* if there exists a unique arrow $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$.
- We say f is a *monomorphism* (or *monic*) if for every parallel pair of arrows $g, h : Z \rightrightarrows X$,

$$Z \overset{g}{\underset{h}{\rightrightarrows}} X \overset{f}{\rightarrowtail} Y. \qquad f \circ g = f \circ h \Rightarrow g = h.$$

- We say f is an *epimorphism* (or *epic*) if for every parallel pair of arrows $g, h : Y \rightrightarrows Z$,

$$X \overset{f}{\twoheadrightarrow} Y \overset{g}{\underset{h}{\rightrightarrows}} Z. \qquad g \circ f = h \circ f \Rightarrow g = h.$$

In **Set**, these are the bijections, injections, and surjections.

Arrows

Let C be a category and $f : X \to Y$ be an arrow.

- We say f is an *isomorphism* if there exists a unique arrow $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$.
- We say f is a *monomorphism* (or *monic*) if for every parallel pair of arrows $g, h : Z \rightrightarrows X$,

$$Z \underset{h}{\overset{g}{\rightrightarrows}} X \overset{f}{\rightarrowtail} Y. \qquad f \circ g = f \circ h \Rightarrow g = h.$$

- We say f is an *epimorphism* (or *epic*) if for every parallel pair of arrows $g, h : Y \rightrightarrows Z$,

$$X \overset{f}{\twoheadrightarrow} Y \underset{h}{\overset{g}{\rightrightarrows}} Z. \qquad g \circ f = h \circ f \Rightarrow g = h.$$

In Set, these are the bijections, injections, and surjections.

Let C be a category and $f : X \to Y$ be an arrow.

- We say f is an *isomorphism* if there exists a unique arrow $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$.
- We say f is a *monomorphism* (or *monic*) if for every parallel pair of arrows $g, h : Z \rightrightarrows X$,

$$Z \overset{g}{\underset{h}{\rightrightarrows}} X \overset{f}{\rightarrowtail} Y. \qquad f \circ g = f \circ h \Rightarrow g = h.$$

- We say f is an *epimorphism* (or *epic*) if for every parallel pair of arrows $g, h : Y \rightrightarrows Z$,

$$X \overset{f}{\twoheadrightarrow} Y \overset{g}{\underset{h}{\rightrightarrows}} Z. \qquad g \circ f = h \circ f \Rightarrow g = h.$$

In Set, these are the bijections, injections, and surjections.

Arrows

Let C be a category and $f : X \to Y$ be an arrow.

- We say f is an *isomorphism* if there exists a unique arrow $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$.
- We say f is a *monomorphism* (or *monic*) if for every parallel pair of arrows $g, h : Z \rightrightarrows X$,

$$Z \underset{h}{\overset{g}{\rightrightarrows}} X \overset{f}{\rightarrowtail} Y. \qquad f \circ g = f \circ h \Rightarrow g = h.$$

- We say f is an *epimorphism* (or *epic*) if for every parallel pair of arrows $g, h : Y \rightrightarrows Z$,

$$X \overset{f}{\twoheadrightarrow} Y \underset{h}{\overset{g}{\rightrightarrows}} Z. \qquad g \circ f = h \circ f \Rightarrow g = h.$$

In Set, these are the bijections, injections, and surjections.

Functors

- Let C and D be categories. A *(covariant) functor* $F : \mathsf{C} \to \mathsf{D}$ is a morphism consisting of the following data:
  - for each object $X \in \mathsf{C}_0$ there is an object $F(C) \in \mathsf{D}_0$,
  - for each morphism $f : X \to Y \in \mathsf{C}_1$, there is an arrow $F(f) : F(X) \to F(Y) \in \mathsf{D}_1$.
- The morphism F is required to satisfy the following axioms:
  - For any composable pair $f, g \in \mathsf{C}_1$, we have $F(g) \circ F(f) = F(g \circ f)$,
  - For each object $X \in \mathsf{C}_0$, we have $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$.

Functors

- Let C and D be categories. A *(covariant) functor* $F : \mathsf{C} \to \mathsf{D}$ is a morphism consisting of the following data:
  - for each object $X \in \mathsf{C}_0$ there is an object $F(C) \in \mathsf{D}_0$,
  - for each morphism $f : X \to Y \in \mathsf{C}_1$, there is an arrow $F(f) : F(X) \to F(Y) \in \mathsf{D}_1$.
- The morphism F is required to satisfy the following axioms:
  - For any composable pair $f, g \in \mathsf{C}_1$, we have $F(g) \circ F(f) = F(g \circ f)$,
  - For each object $X \in \mathsf{C}_0$, we have $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$.

Examples of Functors

- For each category C we have the *identity functor* $\mathrm{Id}_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$.
- If $Y \in \mathsf{D}_0$, we have the *constant functor* $\hat{Y} : \mathsf{C} \to \mathsf{F}$ with $\hat{Y}(X) = Y$ and $\hat{Y}(f) = \mathrm{id}_Y$.
- There are *forgetful functors* $U : \mathsf{Grp} \to \mathsf{Set}$ and $U : \mathsf{Top} \to \mathsf{Set}$ that forgets the structure and sends an object to the underlying set.
- The *fundamental group* is a functor $\pi_1 : \mathsf{Top} \to \mathsf{Grp}$.

Examples of Functors

- For each category C we have the *identity functor* $\mathrm{Id}_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$.
- If $Y \in \mathsf{D}_0$, we have the *constant functor* $\hat{Y} : \mathsf{C} \to \mathsf{F}$ with $\hat{Y}(X) = Y$ and $\hat{Y}(f) = \mathrm{id}_Y$.
- There are *forgetful functors* $U : \mathsf{Grp} \to \mathsf{Set}$ and $U : \mathsf{Top} \to \mathsf{Set}$ that forgets the structure and sends an object to the underlying set.
- The *fundamental group* is a functor $\pi_1 : \mathsf{Top} \to \mathsf{Grp}$.

Examples of Functors

- For each category C we have the *identity functor* $\mathrm{Id}_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$.
- If $Y \in \mathsf{D}_0$, we have the *constant functor* $\hat{Y} : \mathsf{C} \to \mathsf{F}$ with $\hat{Y}(X) = Y$ and $\hat{Y}(f) = \mathrm{id}_Y$.
- There are *forgetful functors* $U : \mathsf{Grp} \to \mathsf{Set}$ and $U : \mathsf{Top} \to \mathsf{Set}$ that forgets the structure and sends an object to the underlying set.
- The *fundamental group* is a functor $\pi_1 : \mathsf{Top} \to \mathsf{Grp}$.

Examples of Functors

- For each category C we have the *identity functor* $\mathrm{Id}_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$.
- If $Y \in \mathsf{D}_0$, we have the *constant functor* $\hat{Y} : \mathsf{C} \to \mathsf{F}$ with $\hat{Y}(X) = Y$ and $\hat{Y}(f) = \mathrm{id}_Y$.
- There are *forgetful functors* $U : \mathsf{Grp} \to \mathsf{Set}$ and $U : \mathsf{Top} \to \mathsf{Set}$ that forgets the structure and sends an object to the underlying set.
- The *fundamental group* is a functor $\pi_1 : \mathsf{Top} \to \mathsf{Grp}$.

Natural Transformations

- Given categories C and D and functors $F, G : \mathsf{C} \rightrightarrows \mathsf{D}$, a *natural transformation* $\alpha : F \Rightarrow G$ consists of:
  - For each object $X \in \mathsf{C}$ an arrow $\alpha_X : F(X) \to G(X)$ in D, the collection of which define the *components* of α,

  so that, for any morphism $f : X \to Y$ in C, TFDC in D

$$\begin{tikzcd} F(X) \arrow[r, "\alpha_X"] \arrow[d, "F(f)"'] & G(X) \arrow[d, "G(f)"] \\ F(Y) \arrow[r, "\alpha_Y"] & G(Y) \end{tikzcd} \qquad G(f) \circ \alpha_X = \alpha_Y \circ F(f).$$

- A *natural isomorphism* is a natural transformation $\alpha : F \Rightarrow G$ in which every component α_X is an isomorphism.

Natural Transformations

- Given categories C and D and functors $F, G : \mathsf{C} \rightrightarrows \mathsf{D}$, a *natural transformation* $\alpha : F \Rightarrow G$ consists of:
  - For each object $X \in \mathsf{C}$ an arrow $\alpha_X : F(X) \to G(X)$ in D, the collection of which define the *components* of α,

so that, for any morphism $f : X \to Y$ in C, TFDC in D

$$\begin{tikzcd} F(X) \arrow[r, "\alpha_X"] \arrow[d, "F(f)"'] & G(X) \arrow[d, "G(f)"] \\ F(Y) \arrow[r, "\alpha_Y"] & G(Y) \end{tikzcd} \qquad G(f) \circ \alpha_X = \alpha_Y \circ F(f).$$

- A *natural isomorphism* is a natural transformation $\alpha : F \Rightarrow G$ in which every component α_X is an isomorphism.

Natural Transformations

- Given categories C and D and functors $F, G : \mathsf{C} \rightrightarrows \mathsf{D}$, a *natural transformation* $\alpha : F \Rightarrow G$ consists of:
  - For each object $X \in \mathsf{C}$ an arrow $\alpha_X : F(X) \to G(X)$ in D, the collection of which define the *components* of α,

  so that, for any morphism $f : X \to Y$ in C, TFDC in D

$$\begin{array}{ccc} F(X) & \xrightarrow{\alpha_X} & G(X) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(Y) & \xrightarrow{\alpha_Y} & G(Y) \end{array} \qquad G(f) \circ \alpha_X = \alpha_Y \circ F(f).$$

- A *natural isomorphism* is a natural transformation $\alpha : F \Rightarrow G$ in which every component α_X is an isomorphism.

Equivalence of Categories

- We say that a functor $F : \mathsf{C} \to \mathsf{D}$ is an *equivalence of categories* if there exists a functor $G : \mathsf{D} \to \mathsf{C}$ such that we have natural isomorphism

$$G \circ F \cong \mathrm{Id}_{\mathsf{C}} \text{ and } F \circ G \cong \mathrm{Id}_{\mathsf{D}}.$$

- Given two categories C and D, the *functor category* $\mathrm{Fun}(\mathsf{C}, \mathsf{D})$ whose objects are functors $F : \mathsf{C} \to \mathsf{D}$ and whose morphisms are natural transformations.

Equivalence of Categories

- We say that a functor $F : \mathsf{C} \to \mathsf{D}$ is an *equivalence of categories* if there exists a functor $G : \mathsf{D} \to \mathsf{C}$ such that we have natural isomorphism

$$G \circ F \cong \mathrm{Id}_{\mathsf{C}} \text{ and } F \circ G \cong \mathrm{Id}_{\mathsf{D}}.$$

- Given two categories C and D, the *functor category* $\mathrm{Fun}(\mathsf{C}, \mathsf{D})$ whose objects are functors $F : \mathsf{C} \to \mathsf{D}$ and whose morphisms are natural transformations.

Syntactic Category

Let T be a first-order theory.

We can construct the *syntactic category of T*, $\mathrm{Syn}(T)$, as follows:

- The objects of $\mathrm{Syn}(T)$ will be the formulae $\phi(\overline{x})$ of the language of T (which we will denote by $[\phi(\overline{x})]$ to avoid confusion).

  Intuitively, the collection of tuples $\overline{x}$ satisfying ϕ.

- Let $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ be objects of $\mathrm{Syn}(T)$. Then a morphism from $[\phi(\overline{x})]$ to $[\psi(\overline{y})]$ will be an equivalence class of formulae $\theta(\overline{x}, \overline{y})$ satisfying

$$T \vDash (\forall \overline{x}, \overline{y}.[\theta(\overline{x}, \overline{y}) \Rightarrow \phi(\overline{x}) \wedge \psi(\overline{y})] \wedge \forall \overline{x}.[\phi(\overline{x}) \Rightarrow \exists! \overline{y}.\theta(\overline{x}, \overline{y})]),$$

  where two formulae $\theta(\overline{x}, \overline{y}), \theta'(\overline{x}, \overline{y})$ equivalent if

$$T \vDash \forall \overline{x}, \overline{y}.[\theta(\overline{x}, \overline{y}) \Leftrightarrow \theta'(\overline{x}, \overline{y})].$$

  Intuitively, T implies that θ defines the graph from $\overline{x}$ satisfying ϕ to $\overline{y}$ satisfying ψ.

Syntactic Category

Let T be a first-order theory.

We can construct the *syntactic category of T*, $\mathrm{Syn}(T)$, as follows:

- The objects of $\mathrm{Syn}(T)$ will be the formulae $\phi(\overline{x})$ of the language of T (which we will denote by $[\phi(\overline{x})]$ to avoid confusion).
  Intuitively, the collection of tuples $\overline{x}$ satisfying ϕ.
- Let $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ be objects of $\mathrm{Syn}(T)$. Then a morphism from $[\phi(\overline{x})]$ to $[\psi(\overline{y})]$ will be an equivalence class of formulae $\theta(\overline{x}, \overline{y})$ satisfying

  $$T \vDash (\forall \overline{x}, \overline{y}.[\theta(\overline{x}, \overline{y}) \Rightarrow \phi(\overline{x}) \wedge \psi(\overline{y})] \wedge \forall \overline{x}.[\phi(\overline{x}) \Rightarrow \exists! \overline{y}.\theta(\overline{x}, \overline{y})]),$$

  where two formulae $\theta(\overline{x}, \overline{y}), \theta'(\overline{x}, \overline{y})$ equivalent if

  $$T \vDash \forall \overline{x}, \overline{y}.[\theta(\overline{x}, \overline{y}) \Leftrightarrow \theta'(\overline{x}, \overline{y})].$$

  Intuitively, T implies that θ defines the graph from $\overline{x}$ satisfying ϕ to $\overline{y}$ satisfying ψ.

Syntactic Category

- Let $[\phi(\overline{x})]$ be an object of $\mathrm{Syn}(T)$ and $\bar{x'}$ be variables not appearing in $\phi(\overline{x})$. Then the morphism

$$[\phi(\overline{x})] \xrightarrow{[\phi(\overline{x}) \wedge (\overline{x} = \bar{x'})]} [\phi(\bar{x'})]$$

is the identity morphism for $[\phi(\overline{x})]$.

- Let $[\theta(\overline{x}, \overline{y})] : [\phi(\overline{x})] \to [\psi(\overline{y})]$ and $[\theta'(\overline{y}, \overline{z})] : [\psi(\overline{y})] \to [\gamma(\overline{z})]$ be morphisms in $\mathrm{Syn}(T)$. Then their composition is $[\rho(\overline{x}, \overline{z})] = [\theta'(\overline{y}, \overline{z})] \circ [\theta(\overline{x}, \overline{y})]$, where

$$\rho(\overline{x}, \overline{z}) := \exists \overline{y}.[\theta(\overline{x}, \overline{y}) \wedge \theta'(\overline{y}, \overline{z})].$$

- Let $[\phi(\overline{x})]$ be an object of $\mathrm{Syn}(T)$ and $\bar{x'}$ be variables not appearing in $\phi(\overline{x})$. Then the morphism

$$[\phi(\overline{x})] \xrightarrow{[\phi(\overline{x})\wedge(\overline{x}=\bar{x'})]} [\phi(\bar{x'})]$$

  is the identity morphism for $[\phi(\overline{x})]$.
- Let $[\theta(\overline{x},\overline{y})] : [\phi(\overline{x})] \to [\psi(\overline{y})]$ and $[\theta'(\overline{y},\overline{z})] : [\psi(\overline{y})] \to [\gamma(\overline{z})]$ be morphisms in $\mathrm{Syn}(T)$. Then their composition is $[\rho(\overline{x},\overline{z})] = [\theta'(\overline{y},\overline{z})] \circ [\theta(\overline{x},\overline{y})]$, where

$$\rho(\overline{x},\overline{z}) := \exists \overline{y}.[\theta(\overline{x},\overline{y}) \wedge \theta'(\overline{y},\overline{z})].$$

Examples of Syntactic Categories

- The *category of definable sets* $\mathrm{Def}(T)$ has, for each model $M \vDash T$:
  - For each formula $\phi(\overline{x})$ of T we have a *definable set* $M[\phi] = \{\overline{a} \in M^n : M \vDash \phi(\overline{a})\}$.
  - Each arrow $f : M[\phi] \to M[\psi]$ is *definable* by a formula $\theta(\overline{x}, \overline{y})$ with

$$M[\theta] = \left\{(\overline{c}, \overline{d}) \in M^{n+m} : M \vDash \phi(\overline{c}) \text{ and } f(\overline{c}) = \overline{d}\right\}.$$

- For each category C (with sufficient structure) there exists a (typed) first-order theory $T(\mathrm{C})$ such that C is equivalent to $\mathrm{Syn}(T(\mathrm{C}))$.

Examples of Syntactic Categories

- The *category of definable sets* $\mathrm{Def}(T)$ has, for each model $M \models T$:
  - For each formula $\phi(\overline{x})$ of T we have a *definable set* $M[\phi] = \{\overline{a} \in M^n : M \models \phi(\overline{a})\}$.
  - Each arrow $f : M[\phi] \to M[\psi]$ is *definable* by a formula $\theta(\overline{x}, \overline{y})$ with

$$M[\theta] = \left\{(\overline{c}, \overline{d}) \in M^{n+m} : M \models \phi(\overline{c}) \text{ and } f(\overline{c}) = \overline{d}\right\}.$$

- For each category C (with sufficient structure) there exists a (typed) first-order theory $T(\mathrm{C})$ such that C is equivalent to $\mathrm{Syn}(T(\mathrm{C}))$.

Question

What structure do Set and Syn(T) have?

Products

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of* X *and* Y in C is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd}
& Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists! \beta"] \arrow[dr, "g"] & \\
X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y
\end{tikzcd}
\qquad \pi \circ \beta = f \text{ and } \pi' \circ \beta = g$$

In $\mathrm{Syn}(T)$, the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \bar{x}'))]$, $[(\phi \wedge \psi \wedge (\overline{x} = \bar{x}'))]$.

Products

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of X and Y* in C is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists!\ \beta"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y \end{tikzcd}$$

$\pi \circ \beta = f$ and $\pi' \circ \beta = g$

In Syn(T), the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}')]$, $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$.

Products

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of X and Y* in C is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists!\,\beta"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y \end{tikzcd} \qquad \pi \circ \beta = f \text{ and } \pi' \circ \beta = g$$

In $\mathrm{Syn}(T)$, the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$, $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$.

Products

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of X and Y* in C is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists!\,\beta"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y \end{tikzcd}$$

$\pi \circ \beta = f$ and $\pi' \circ \beta = g$

In $\mathrm{Syn}(T)$, the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$, $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$.

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of X and Y in C* is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists!"', "\beta"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y \end{tikzcd}$$

$$\pi \circ \beta = f \text{ and } \pi' \circ \beta = g$$

In $\mathrm{Syn}(T)$, the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$, $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$.

Given two sets X, Y, we can form the *Cartesian product* $X \times Y$, which comes with projection arrows $\pi : X \times Y \to X$, $\pi' : X \times Y \to Y$.

In general, given X, Y in C the *product of X and Y* in C is $(X \times Y, \pi, \pi')$ such that

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, dashed, "\exists!\ \beta"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi"] \arrow[r, "\pi'"'] & Y \end{tikzcd}$$

$$\pi \circ \beta = f \text{ and } \pi' \circ \beta = g$$

In $\mathrm{Syn}(T)$, the product of $[\phi(\overline{x})]$ and $[\psi(\overline{y})]$ ($\overline{x}$ and $\overline{y}$ disjoint) is $[(\phi \wedge \psi)(\overline{x}, \overline{y})]$ with arrows $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$, $[(\phi \wedge \psi \wedge (\overline{x} = \overline{x}'))]$.

Terminal Object

We have that any singleton set $\{*\}$ is such that for any other set X, there is a unique arrow

$$1_X : X \to \{*\} :: 1_X(x) = *.$$

In general, a *terminal object* in C is an object 1 such that

$$X \overset{\beta}{\underset{\exists!}{\dashrightarrow}} 1.$$

In $\mathsf{Syn}(T)$, the terminal object is $[\top]$.

Terminal Object

We have that any singleton set $\{*\}$ is such that for any other set X, there is a unique arrow

$$1_X : X \to \{*\} :: 1_X(x) = *.$$

In general, a *terminal object* in C is an object 1 such that

$$X \xrightarrow[\exists!]{\beta} 1.$$

In $\mathrm{Syn}(T)$, the terminal object is $[\top]$.

We have that any singleton set $\{*\}$ is such that for any other set X, there is a unique arrow

$$1_X : X \to \{*\} :: 1_X(x) = *.$$

In general, a *terminal object* in C is an object 1 such that

$$X \xrightarrow[\exists!]{\beta} 1.$$

In $\mathrm{Syn}(T)$, the terminal object is $[\top]$.

Terminal Object

We have that any singleton set $\{*\}$ is such that for any other set X, there is a unique arrow

$$1_X : X \to \{*\} :: 1_X(x) = *.$$

In general, a *terminal object* in $\mathcal{C}$ is an object 1 such that

$$X \overset{\beta}{\underset{\exists!}{\dashrightarrow}} 1.$$

In $\mathrm{Syn}(T)$, the terminal object is $[\top]$.

Terminal Object

We have that any singleton set $\{*\}$ is such that for any other set X, there is a unique arrow

$$1_X : X \to \{*\} :: 1_X(x) = *.$$

In general, a *terminal object* in C is an object 1 such that

$$X \overset{\beta}{\underset{\exists!}{\dashrightarrow}} 1.$$

In $\mathrm{Syn}(T)$, the terminal object is $[\top]$.

Pullbacks

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[dr, dashed, "\beta"', "\exists!"] \arrow[drr, bend left, "\lambda'"] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$$g \circ f' = f \circ g'$$
$$g \circ \lambda' = f \circ \lambda$$
$g' \circ \beta = \lambda$ and $f' \circ \beta = \lambda'$.

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Pullbacks

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[dr, dashed, "\beta", "\exists!"'] \arrow[drr, bend left, "\lambda'"] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$$g \circ f' = f \circ g'$$

$$g \circ \lambda' = f \circ \lambda$$

$$g' \circ \beta = \lambda \text{ and } f' \circ \beta = \lambda'.$$

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Pullbacks

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[dr, dashed, "\beta", "\exists!"'] \arrow[drr, bend left, "\lambda'"] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$g \circ f' = f \circ g'$

$g \circ \lambda' = f \circ \lambda$

$g' \circ \beta = \lambda$ and $f' \circ \beta = \lambda'$.

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[dr, dashed, "\beta", "\exists!"'] \arrow[drr, bend left, "\lambda'"] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$g \circ f' = f \circ g'$

$g \circ \lambda' = f \circ \lambda$

$g' \circ \beta = \lambda$ and $f' \circ \beta = \lambda'$.

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Pullbacks

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}\,.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[dr, dashed, "\beta", "\exists!"'] \arrow[drr, bend left, "\lambda'"] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$$g \circ f' = f \circ g'$$
$$g \circ \lambda' = f \circ \lambda$$
$$g' \circ \beta = \lambda \text{ and } f' \circ \beta = \lambda'.$$

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Pullbacks

Let $f : X \to Z$ and $g : Y \to Z$ be set functions. Then we have a subset of $X \times Y$ (with projections) known as the *fibered product* defined as follows:

$$X \times_Z Y = \{(x, y) \in X \times Y : f(x) = g(y)\}\,.$$

In general, given $f : X \to Z$, $g : Y \to Z$, we define their *pullback* to be $(X \times_Z Y, f', g')$ such that

$$\begin{tikzcd}
W \arrow[drr, bend left, "\lambda'"] \arrow[dr, dashed, "\beta", "\exists!"'] \arrow[ddr, bend right, "\lambda"'] & & \\
& X \times_Z Y \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, phantom, "\lrcorner"] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"] & Z.
\end{tikzcd}$$

$$g \circ f' = f \circ g'$$
$$g \circ \lambda' = f \circ \lambda$$
$$g' \circ \beta = \lambda \text{ and } f' \circ \beta = \lambda'.$$

In $\mathrm{Syn}(T)$, the pullback of $[\theta(\overline{x}, \overline{z})]$ and $[\theta'(\overline{y}, \overline{z})]$ is $[\rho(\overline{x}, \overline{y})] := \exists \overline{z}.(\theta(\overline{x}, \overline{z}) \wedge \theta'(\overline{y}, \overline{z}))$.

Limits

- A *diagram* of *shape* J in a category C is a functor $D : \mathsf{J} \to \mathsf{C}$.
- The diagrams for product, terminal object, and pullback were

$$\bullet \qquad \bullet, \qquad\qquad , \qquad \bullet \longrightarrow \bullet \longleftarrow \bullet.$$

- A *limit of D* is a natural transformation $\lambda : \widehat{\lim_{\mathsf{J}} D} \Rightarrow D$ such that

$$\begin{tikzcd}
C \arrow[rr, dashed, "\beta", "\exists!"'] \arrow[dr, "\pi_J"] \arrow[ddr, bend right, "\pi_K"'] & & \lim_{\mathsf{J}} D \arrow[dl, "\lambda_J"'] \arrow[ddl, bend left, "\lambda_K"] \\
& D(J) \arrow[d, "D(f)"] & \\
& D(K) &
\end{tikzcd}$$

$$\begin{gathered}
\lambda_K = D(f) \circ \lambda_J \\
\pi_K = D(f) \circ \pi_J \\
\forall J \in \mathsf{J}_0 \\
\pi_J = \lambda_J \circ \beta.
\end{gathered}$$

Limits

- A *diagram* of *shape* J in a category C is a functor $D : \mathsf{J} \to \mathsf{C}$.
- The diagrams for product, terminal object, and pullback were

$$\bullet \qquad \bullet, \qquad\qquad , \qquad \bullet \longrightarrow \bullet \longleftarrow \bullet.$$

- A *limit of* D is a natural transformation $\lambda : \widehat{\lim_{\mathsf{J}} D} \Rightarrow D$ such that

$$\begin{tikzcd}
C \arrow[rr, dashed, "\beta", "\exists!"'] \arrow[dr, "\pi_J"'] \arrow[ddr, bend right, "\pi_K"'] & & \lim_{\mathsf{J}} D \arrow[dl, "\lambda_J"] \arrow[ddl, bend left, "\lambda_K"] \\
& D(J) \arrow[d, "D(f)"] & \\
& D(K) &
\end{tikzcd}$$

$$\lambda_K = D(f) \circ \lambda_J$$
$$\pi_K = D(f) \circ \pi_J$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \lambda_J \circ \beta.$$

Limits

- A *diagram* of *shape* J in a category C is a functor $D : \mathsf{J} \to \mathsf{C}$.
- The diagrams for product, terminal object, and pullback were

$$\bullet \qquad \bullet, \qquad\qquad , \qquad\qquad \bullet \longrightarrow \bullet \longleftarrow \bullet.$$

- A *limit of D* is a natural transformation $\lambda : \widehat{\lim_{\mathsf{J}} D} \Rightarrow D$ such that

$$\begin{tikzcd}
C \arrow[rr, dashed, "\beta", "\exists!"'] \arrow[dr, "\pi_J"'] \arrow[ddr, bend right, "\pi_K"'] & & \lim_{\mathsf{J}} D \arrow[dl, "\lambda_J"] \arrow[ddl, bend left, "\lambda_K"] \\
& D(J) \arrow[d, "D(f)"] & \\
& D(K) &
\end{tikzcd}$$

$$\lambda_K = D(f) \circ \lambda_J$$
$$\pi_K = D(f) \circ \pi_J$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \lambda_J \circ \beta.$$

Limits

- A *diagram* of *shape* J in a category C is a functor $D : \mathsf{J} \to \mathsf{C}$.
- The diagrams for product, terminal object, and pullback were

$$\bullet \qquad \bullet, \qquad\qquad , \qquad\qquad \bullet \longrightarrow \bullet \longleftarrow \bullet.$$

- A *limit of D* is a natural transformation $\lambda : \widehat{\lim_{\mathsf{J}} D} \Rightarrow D$ such that

$$\begin{tikzcd}
C \arrow[rr, dashed, "\beta", "\exists!"'] \arrow[dr, "\pi_J"] \arrow[ddr, bend right, "\pi_K"'] & & \lim_{\mathsf{J}} D \arrow[dl, "\lambda_J"'] \arrow[ddl, bend left, "\lambda_K"] \\
& D(J) \arrow[d, "D(f)"] & \\
& D(K) &
\end{tikzcd}$$

$$\lambda_K = D(f) \circ \lambda_J$$

$$\pi_K = D(f) \circ \pi_J$$

$$\forall J \in \mathsf{J}_0$$

$$\pi_J = \lambda_J \circ \beta.$$

- A *diagram* of *shape* J in a category C is a functor $D : \mathsf{J} \to \mathsf{C}$.
- The diagrams for product, terminal object, and pullback were

$$\bullet \qquad \bullet, \qquad\qquad , \qquad\qquad \bullet \longrightarrow \bullet \longleftarrow \bullet.$$

- A *limit of D* is a natural transformation $\lambda : \widehat{\lim_{\mathsf{J}} D} \Rightarrow D$ such that

$$\begin{tikzcd}
C \arrow[rr, dashed, "\beta", "\exists!"'] \arrow[dr, "\pi_J"'] \arrow[ddr, bend right, "\pi_K"'] & & \lim_{\mathsf{J}} D \arrow[dl, "\lambda_J"] \arrow[ddl, bend left, "\lambda_K"] \\
& D(J) \arrow[d, "D(f)"] & \\
& D(K) &
\end{tikzcd}$$

$$\lambda_K = D(f) \circ \lambda_J$$
$$\pi_K = D(f) \circ \pi_J$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \lambda_J \circ \beta.$$

Finite Limits

- A diagram $D : \mathsf{J} \to \mathsf{C}$ is said to be *finite* if J is finite, that is, J has a finite number of morphisms.
- We say a category C has *finite limits* if it has a limit for every finite diagram.
- A category has finite limits if it has *pullbacks* for every pair of arrows and a *terminal object*.
- Both Set and $\mathsf{Syn}(T)$ have finite limits.

Finite Limits

- A diagram $D : \mathsf{J} \to \mathsf{C}$ is said to be *finite* if J is finite, that is, J has a finite number of morphisms.
- We say a category C has *finite limits* if it has a limit for every finite diagram.
- A category has finite limits if it has *pullbacks* for every pair of arrows and a *terminal object*.
- Both Set and $\mathsf{Syn}(T)$ have finite limits.

Finite Limits

- A diagram $D : \mathsf{J} \to \mathsf{C}$ is said to be *finite* if J is finite, that is, J has a finite number of morphisms.
- We say a category C has *finite limits* if it has a limit for every finite diagram.
- A category has finite limits if it has *pullbacks* for every pair of arrows and a *terminal object*.
- Both Set and $\mathsf{Syn}(T)$ have finite limits.

Finite Limits

- A diagram $D : \mathsf{J} \to \mathsf{C}$ is said to be *finite* if J is finite, that is, J has a finite number of morphisms.
- We say a category C has *finite limits* if it has a limit for every finite diagram.
- A category has finite limits if it has *pullbacks* for every pair of arrows and a *terminal object*.
- Both Set and $\mathsf{Syn}(T)$ have finite limits.

Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram.

A *colimit of* D is a natural transformation $\lambda : D \Rightarrow \widehat{\operatorname{colim}_{\mathsf{J}} D}$ such that

$$\begin{tikzcd}
C & & \operatorname{colim}_{\mathsf{J}} D \arrow[ll, dashed, "\beta"', "\exists!"] \\
& D(J) \arrow[ul, "\pi_J"'] \arrow[ur, "\lambda_J"] \arrow[d, "D(f)"] & \\
& D(K) \arrow[uul, bend left, "\pi_K"] \arrow[uur, bend right, "\lambda_K"'] &
\end{tikzcd}$$

$$\lambda_J = \lambda_K \circ D(f)$$
$$\pi_J = \pi_K \circ D(f)$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \beta \circ \lambda_J.$$

In Set, we have

- Coporducts: Disjoint unions
- Initial object: $\emptyset$
- Pushouts: Quotient of a disjoint union.

Colimits

Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram.

A *colimit of D* is a natural transformation $\lambda : D \Rightarrow \widehat{\operatorname{colim}_{\mathsf{J}}} D$ such that

$$\begin{tikzcd}
C & & \operatorname{colim}_{\mathsf{J}} D \arrow[ll, dashed, "\beta"', "\exists!"] \\
& D(J) \arrow[ul, "\pi_J"] \arrow[ur, "\lambda_J"'] \arrow[d, "D(f)"] & \\
& D(K) \arrow[uul, bend left, "\pi_K"] \arrow[uur, bend right, "\lambda_K"'] &
\end{tikzcd}$$

$\lambda_J = \lambda_K \circ D(f)$

$\pi_J = \pi_K \circ D(f)$

$\forall J \in \mathsf{J}_0$

$\pi_J = \beta \circ \lambda_J.$

In Set, we have

- Coporducts: Disjoint unions
- Initial object: $\emptyset$
- Pushouts: Quotient of a disjoint union.

Colimits

Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram.

A *colimit of D* is a natural transformation $\lambda : D \Rightarrow \widehat{\operatorname{colim}_{\mathsf{J}} D}$ such that

$$
\begin{tikzcd}
C & & \operatorname{colim}_{\mathsf{J}} D \arrow[ll, dashed, "\beta"', "\exists!"] \\
& D(J) \arrow[ul, "\pi_J"'] \arrow[ur, "\lambda_J"] \arrow[d, "D(f)"] & \\
& D(K) \arrow[uul, bend left, "\pi_K"] \arrow[uur, bend right, "\lambda_K"'] &
\end{tikzcd}
$$

$$\lambda_J = \lambda_K \circ D(f)$$
$$\pi_J = \pi_K \circ D(f)$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \beta \circ \lambda_J.$$

In Set, we have

- Coporducts: Disjoint unions
- Initial object: $\emptyset$
- Pushouts: Quotient of a disjoint union.

Colimits

Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram.

A *colimit of D* is a natural transformation $\lambda : D \Rightarrow \widehat{\operatorname{colim}_{\mathsf{J}} D}$ such that

$$\begin{tikzcd}
C & & \operatorname{colim}_{\mathsf{J}} D \arrow[ll, dashed, "\beta"', "\exists!"] \\
& D(J) \arrow[ul, "\pi_J"'] \arrow[ur, "\lambda_J"] \arrow[d, "D(f)"] & \\
& D(K) \arrow[uul, bend left, "\pi_K"] \arrow[uur, bend right, "\lambda_K"'] &
\end{tikzcd}$$

$$\lambda_J = \lambda_K \circ D(f)$$
$$\pi_J = \pi_K \circ D(f)$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \beta \circ \lambda_J.$$

In Set, we have

- Copordučts: Disjoint unions
- Initial object: $\emptyset$
- Pushouts: Quotient of a disjoint union.

Colimits

Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram.

A *colimit of D* is a natural transformation $\lambda : D \Rightarrow \widehat{\operatorname{colim}_{\mathsf{J}} D}$ such that

$$\begin{tikzcd}
C & & \operatorname{colim}_{\mathsf{J}} D \arrow[ll, dashed, "\beta"', "\exists!"] \\
& D(J) \arrow[ul, "\pi_J"] \arrow[ur, "\lambda_J"'] \arrow[d, "D(f)"] & \\
& D(K) \arrow[uul, bend left, "\pi_K"] \arrow[uur, bend right, "\lambda_K"'] &
\end{tikzcd}$$

$$\lambda_J = \lambda_K \circ D(f)$$
$$\pi_J = \pi_K \circ D(f)$$
$$\forall J \in \mathsf{J}_0$$
$$\pi_J = \beta \circ \lambda_J.$$

In Set, we have

- Coporducts: Disjoint unions
- Initial object: $\emptyset$
- Pushouts: Quotient of a disjoint union.

Filtered Category

- A category J is said to be *filtered* if
  - There exists an object in J.
  - For every pair of objects $X, Y \in \mathsf{J}_0$, there exists an object $Z \in \mathsf{J}_0$ and a pair of morphisms

$$X \longrightarrow Z \longleftarrow Y.$$

  - For every parallel pair of morphism $f, g : X \rightrightarrows Y$ in J, there exists a morphism $h : Y \to Z$ such that

$$X \underset{g}{\overset{f}{\rightrightarrows}} Y \xrightarrow{h} Z \qquad h \circ f = h \circ g.$$

- A filtered category is to be thought of as a categorification of a *directed set*.

Filtered Category

- A category J is said to be *filtered* if
  - There exists an object in J.
  - For every pair of objects $X, Y \in J_0$, there exists an object $Z \in J_0$ and a pair of morphisms

$$X \longrightarrow Z \longleftarrow Y.$$

  - For every parallel pair of morphism $f, g : X \rightrightarrows Y$ in J, there exists a morphism $h : Y \to Z$ such that

$$X \overset{f}{\underset{g}{\rightrightarrows}} Y \xrightarrow{h} Z \qquad h \circ f = h \circ g.$$

- A filtered category is to be thought of as a categorification of a *directed set*.

Filtered Category

- A category J is said to be *filtered* if
  - There exists an object in J.
  - For every pair of objects $X, Y \in \mathsf{J}_0$, there exists an object $Z \in \mathsf{J}_0$ and a pair of morphisms

$$X \longrightarrow Z \longleftarrow Y.$$

  - For every parallel pair of morphism $f, g : X \rightrightarrows Y$ in J, there exists a morphism $h : Y \to Z$ such that

$$X \underset{g}{\overset{f}{\rightrightarrows}} Y \xrightarrow{h} Z \qquad h \circ f = h \circ g.$$

- A filtered category is to be thought of as a categorification of a *directed set*.

Filtered Category

- A category J is said to be *filtered* if
  - There exists an object in J.
  - For every pair of objects $X, Y \in \mathsf{J}_0$, there exists an object $Z \in \mathsf{J}_0$ and a pair of morphisms

$$X \longrightarrow Z \longleftarrow Y.$$

  - For every parallel pair of morphism $f, g : X \rightrightarrows Y$ in J, there exists a morphism $h : Y \to Z$ such that

$$X \underset{g}{\overset{f}{\rightrightarrows}} Y \xrightarrow{h} Z \qquad h \circ f = h \circ g.$$

- A filtered category is to be thought of as a categorification of a *directed set*.

Filtered Colimits

- Any ultrafilter $\mathcal{U}$ on a set S is a poset (which can be regarded as a category).
- Then $\mathcal{U}^{\mathrm{op}}$ is a directed poset (since $\mathcal{U}$ has is closed under intersections) and is a filtered category.
- A diagram $D : \mathsf{J} \to \mathsf{C}$ is called a *filtered diagram* if J is filtered.
- A colimit of a filtered diagram is called a *filtered colimit*.
- Then the ultraproduct of $\{M_s\}_{s\in S}$ is defined to be the filtered colimit of the product $\prod_{s\in S} M_s$:

$$\left(\prod_{s\in S} M_s\right)/\mathcal{U} = \operatorname*{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s.$$

Filtered Colimits

- Any ultrafilter $\mathcal{U}$ on a set S is a poset (which can be regarded as a category).
- Then $\mathcal{U}^{\mathrm{op}}$ is a directed poset (since $\mathcal{U}$ has is closed under intersections) and is a filtered category.
- A diagram $D : \mathsf{J} \to \mathsf{C}$ is called a *filtered diagram* if J is filtered.
- A colimit of a filtered diagram is called a *filtered colimit*.
- Then the ultraproduct of $\{M_s\}_{s\in S}$ is defined to be the filtered colimit of the product $\prod_{s\in S} M_s$:

$$\left(\prod_{s\in S} M_s\right)/\mathcal{U} = \operatorname*{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s.$$

Filtered Colimits

- Any ultrafilter $\mathcal{U}$ on a set S is a poset (which can be regarded as a category).
- Then $\mathcal{U}^{\mathrm{op}}$ is a directed poset (since $\mathcal{U}$ has is closed under intersections) and is a filtered category.
- A diagram $D : \mathsf{J} \to \mathsf{C}$ is called a *filtered diagram* if J is filtered.
- A colimit of a filtered diagram is called a *filtered colimit*.
- Then the ultraproduct of $\{M_s\}_{s \in S}$ is defined to be the filtered colimit of the product $\prod_{s \in S} M_s$:

$$\left(\prod_{s \in S} M_s\right) / \mathcal{U} = \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s.$$

Filtered Colimits

- Any ultrafilter $\mathcal{U}$ on a set S is a poset (which can be regarded as a category).
- Then $\mathcal{U}^{\mathrm{op}}$ is a directed poset (since $\mathcal{U}$ has is closed under intersections) and is a filtered category.
- A diagram $D : \mathsf{J} \to \mathsf{C}$ is called a *filtered diagram* if J is filtered.
- A colimit of a filtered diagram is called a *filtered colimit*.
- Then the ultraproduct of $\{M_s\}_{s\in S}$ is defined to be the filtered colimit of the product $\prod_{s\in S} M_s$:

$$\left(\prod_{s\in S} M_s\right)/\mathcal{U} = \operatorname*{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s.$$

Filtered Colimits

- Any ultrafilter $\mathcal{U}$ on a set S is a poset (which can be regarded as a category).
- Then $\mathcal{U}^{\mathrm{op}}$ is a directed poset (since $\mathcal{U}$ has is closed under intersections) and is a filtered category.
- A diagram $D : \mathsf{J} \to \mathsf{C}$ is called a *filtered diagram* if J is filtered.
- A colimit of a filtered diagram is called a *filtered colimit*.
- Then the ultraproduct of $\{M_s\}_{s \in S}$ is defined to be the filtered colimit of the product $\prod_{s \in S} M_s$:

$$\left(\prod_{s \in S} M_s \right) / \mathcal{U} = \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s .$$

Categorical Ultraproduct

Let C^+ be a category and let $\mathsf{C} \subseteq \mathsf{C}^+$ be a full subcategory. We say that C *has ultraproducts in* C^+ if the following conditions are satisfied:

- For every collection $\{M_s\}_{s \in S}$ of objects of C indexed by S, there exists a product $\prod_{s \in S} M_s$ in the category C^+.
- For every collection $\{M_s\}_{s \in S}$ of objects of C indexed by S and every ultrafilter $\mathcal{U}$ on S, the diagram

  $$D : \mathcal{U}^{\mathrm{op}} \to \mathsf{C}^+ :: (S_0 \in \mathcal{U}^{\mathrm{op}}) \mapsto \left(\prod_{s \in S_0} M_s \right)$$

  admits a colimit (in the category C^+) which belongs to the subcategory $\mathsf{C} \subseteq \mathsf{C}^+$.
  In this case, we denote this colimit by $\mathrm{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ and refer to it as the *categorical ultraproduct of* $\{M_s\}_{s \in S}$ *w.r.t.* $\mathcal{U}$.

Categorical Ultraproduct

Let C^+ be a category and let $\mathsf{C} \subseteq \mathsf{C}^+$ be a full subcategory. We say that C *has ultraproducts in* C^+ if the following conditions are satisfied:

- For every collection $\{M_s\}_{s\in S}$ of objects of C indexed by S, there exists a product $\prod_{s\in S} M_s$ in the category C^+.
- For every collection $\{M_s\}_{s\in S}$ of objects of C indexed by S and every ultrafilter $\mathcal{U}$ on S, the diagram

$$D : \mathcal{U}^{\mathrm{op}} \to \mathsf{C}^+ :: (S_0 \in \mathcal{U}^{\mathrm{op}}) \mapsto \left(\prod_{s\in S_0} M_s \right)$$

  admits a colimit (in the category C^+) which belongs to the subcategory $\mathsf{C} \subseteq \mathsf{C}^+$.

  In this case, we denote this colimit by $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s$ and refer to it as the *categorical ultraproduct of* $\{M_s\}_{s\in S}$ *w.r.t.* $\mathcal{U}$.

Categorical Ultraproduct

Let C^+ be a category and let $\mathsf{C} \subseteq \mathsf{C}^+$ be a full subcategory. We say that C *has ultraproducts in* C^+ if the following conditions are satisfied:

- For every collection $\{M_s\}_{s \in S}$ of objects of C indexed by S, there exists a product $\prod_{s \in S} M_s$ in the category C^+.
- For every collection $\{M_s\}_{s \in S}$ of objects of C indexed by S and every ultrafilter $\mathcal{U}$ on S, the diagram

$$D : \mathcal{U}^{\mathrm{op}} \to \mathsf{C}^+ :: (S_0 \in \mathcal{U}^{\mathrm{op}}) \mapsto \left(\prod_{s \in S_0} M_s \right)$$

  admits a colimit (in the category C^+) which belongs to the subcategory $\mathsf{C} \subseteq \mathsf{C}^+$.
  In this case, we denote this colimit by $\mathrm{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ and refer to it as the *categorical ultraproduct of* $\{M_s\}_{s \in S}$ *w.r.t.* $\mathcal{U}$.

Ultraproduct Maps

- For each $S_0 \in \mathcal{U}$, we have a *quotient map* given by the colimit above:

$$q_{\mathcal{U}}^{S_0} : \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s.$$

- Given a collection of functions $\{f_s : M_s \to N_s\}_{s \in S}$ in C, we let

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s : \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s$$

denote the unique map in C such that for each $S_0 \in \mathcal{U}$, TFDC in C^+:

$$\begin{tikzcd}
\prod_{s \in S_0} M_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] \arrow[d, "\prod_{s \in S_0} f_s"'] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \arrow[d, "\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s"] \\
\prod_{s \in S_0} N_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s
\end{tikzcd}
\qquad
\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s \circ q_{\mathcal{U}}^{S_0} = q_{\mathcal{U}}^{S_0} \circ \prod_{s \in S_0} f_s$$

Ultraproduct Maps

- For each $S_0 \in \mathcal{U}$, we have a *quotient map* given by the colimit above:

$$q_{\mathcal{U}}^{S_0} : \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s.$$

- Given a collection of functions $\{f_s : M_s \to N_s\}_{s \in S}$ in C, we let

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s : \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s$$

denote the unique map in C such that for each $S_0 \in \mathcal{U}$, TFDC in C^+:

$$\begin{tikzcd}
\prod_{s \in S_0} M_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] \arrow[d, "\prod_{s \in S_0} f_s"'] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \arrow[d, "\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s"] \\
\prod_{s \in S_0} N_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s
\end{tikzcd}
\qquad
\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s \circ q_{\mathcal{U}}^{S_0} = q_{\mathcal{U}}^{S_0} \circ \prod_{s \in S_0} f_s$$

Ultraproduct Maps

- For each $S_0 \in \mathcal{U}$, we have a *quotient map* given by the colimit above:

$$q_{\mathcal{U}}^{S_0} : \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s.$$

- Given a collection of functions $\{f_s : M_s \to N_s\}_{s \in S}$ in C, we let

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s : \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \to \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s$$

denote the unique map in C such that for each $S_0 \in \mathcal{U}$, TFDC in C^+:

$$\begin{tikzcd}
\prod_{s \in S_0} M_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] \arrow[d, "\prod_{s \in S_0} f_s"'] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \arrow[d, "\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s"] \\
\prod_{s \in S_0} N_s \arrow[r, "q_{\mathcal{U}}^{S_0}"] & \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} N_s
\end{tikzcd}
\qquad
\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s \circ q_{\mathcal{U}}^{S_0} = q_{\mathcal{U}}^{S_0} \circ \prod_{s \in S_0} f_s$$

Ultraproduct Functor

- We have the *ultraproduct functor*

$$\begin{aligned}\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathsf{C}^S &\to \mathsf{C} \\ \{M_s\}_{s \in S} &\mapsto \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \\ \{f_s\}_{s \in S} &\mapsto \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} f_s .\end{aligned}$$

Question

- Set and $\mathrm{Syn}(T)$ have finite limits.
- What other structure do they have?

Question

- Set and $\mathrm{Syn}(T)$ have finite limits.
- What other structure do they have?

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, hook, "i_0"'] & & X_1 \arrow[dl, hook', "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall\overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e \qquad \begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd}
X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\
& X. &
\end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathsf{Sub}(X)$ the set of subobjects of X.
- In $\mathsf{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- For any set X, we have a set $\mathcal{P}(X)$ of subsets of X.
- If X is an object in a category C, then a *subobject* is an equivalence class of monomorphisms $i_0 : X_0 \rightarrowtail X$, where

$$i_0 \sim i_1 \Leftrightarrow i_0 = i_1 \circ e$$

$$\begin{tikzcd} X_0 \arrow[rr, "e", "\exists\cong"'] \arrow[dr, tail, "i_0"'] & & X_1 \arrow[dl, tail, "i_1"] \\ & X. & \end{tikzcd}$$

- We will usually refer to a subobject of X by its domain and assume that the monomorphism is supplied.
- We denote by $\mathrm{Sub}(X)$ the set of subobjects of X.
- In $\mathrm{Syn}(T)$, a subobject of $[\phi(\overline{x})]$ has the form $[\psi(\overline{x})]$ which is such that $T \vDash \forall \overline{x}.(\psi(\overline{x}) \Rightarrow \phi(\overline{x}))$.

Subobjects

- We can define a partial order on $\mathrm{Sub}(X)$ by, for $X_0, X_1 \in \mathrm{Sub}(X)$, $X_0 \leq X_1$ if and only if there is a unique morphism $e : X_0 \to X_1$ such that $i_0 = i_1 \circ e$.
- In $\mathcal{P}(X)$, we have that $\emptyset \in \mathcal{P}(X)$ is the *least element* and for any two $X_0, X_1 \in \mathcal{P}(X)$ their union (*join*) $X_0 \cup X_1 \in \mathcal{P}(X)$.
- We have that $\mathrm{Sub}(X)$ (in the categories Set and $\mathrm{Syn}(T)$) is an *upper semilattice*, that is, it has a *least element* 0_X and *joins* $X_0 \vee X_1$.

Subobjects

- We can define a partial order on $\mathrm{Sub}(X)$ by, for $X_0, X_1 \in \mathrm{Sub}(X)$, $X_0 \leq X_1$ if and only if there is a unique morphism $e : X_0 \to X_1$ such that $i_0 = i_1 \circ e$.
- In $\mathcal{P}(X)$, we have that $\emptyset \in \mathcal{P}(X)$ is the *least element* and for any two $X_0, X_1 \in \mathcal{P}(X)$ their union (*join*) $X_0 \cup X_1 \in \mathcal{P}(X)$.
- We have that $\mathrm{Sub}(X)$ (in the categories Set and $\mathrm{Syn}(T)$) is an *upper semilattice*, that is, it has a *least element* 0_X and *joins* $X_0 \vee X_1$.

Subobjects

- We can define a partial order on $\mathrm{Sub}(X)$ by, for $X_0, X_1 \in \mathrm{Sub}(X)$, $X_0 \leq X_1$ if and only if there is a unique morphism $e : X_0 \to X_1$ such that $i_0 = i_1 \circ e$.
- In $\mathcal{P}(X)$, we have that $\emptyset \in \mathcal{P}(X)$ is the *least element* and for any two $X_0, X_1 \in \mathcal{P}(X)$ their union (*join*) $X_0 \cup X_1 \in \mathcal{P}(X)$.
- We have that $\mathrm{Sub}(X)$ (in the categories Set and $\mathrm{Syn}(T)$) is an *upper semilattice*, that is, it has a *least element* 0_X and *joins* $X_0 \vee X_1$.

Subobjects

- If $f : X \to Y$ is a morphism in C, then the *inverse morphism* between $\mathsf{Sub}(Y)$ and $\mathsf{Sub}(X)$ is defined by

$$f^{-1} : \mathsf{Sub}(Y) \to \mathsf{Sub}(X) :: f^{-1}(Y_0) = X \times_Y Y_0$$

- We have that f^{-1} is a *morphisms of upper semilattices* if $f^{-1}(0_Y) = 0_X$ and $f^{-1}(Y_0 \vee Y_1) = f^{-1}(Y_0) \vee f^{-1}(Y_1)$.

Subobjects

- If $f : X \to Y$ is a morphism in $\mathcal{C}$, then the *inverse morphism* between $\mathrm{Sub}(Y)$ and $\mathrm{Sub}(X)$ is defined by

$$f^{-1} : \mathrm{Sub}(Y) \to \mathrm{Sub}(X) :: f^{-1}(Y_0) = X \times_Y Y_0$$

- We have that f^{-1} is a *morphisms of upper semilattices* if $f^{-1}(0_Y) = 0_X$ and $f^{-1}(Y_0 \vee Y_1) = f^{-1}(Y_0) \vee f^{-1}(Y_1)$.

Coequalizers

- If $f, g : X \rightrightarrows Y$ are a parallel pair of arrows in C, then their *coequalizer* is an object Q and arrow $q : Y \to Q$ such that

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "q"] \arrow[dr, "k"'] & Q \arrow[d, dashed, "\exists!"', "\beta"] \\ & & Z. \end{tikzcd}$$

$$q \circ f = q \circ g$$
$$k \circ f = k \circ g$$
$$k = \beta \circ q.$$

- Given a set X and an equivalence relation $R \subseteq X \times X$, the set of equivalence classes X/R with the quotient map $q : X \twoheadrightarrow X/R$ is the coequalizer of the projections $\pi, \pi' : R \rightrightarrows X$.

- If $f, g : X \rightrightarrows Y$ are a parallel pair of arrows in $\mathcal{C}$, then their *coequalizer* is an object Q and arrow $q : Y \to Q$ such that

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "q"] \arrow[dr, "k"'] & Q \arrow[d, dashed, "\exists!"', "\beta"] \\ & & Z. \end{tikzcd}$$

$$q \circ f = q \circ g$$

$$k \circ f = k \circ g$$

$$k = \beta \circ q.$$

- Given a set X and an equivalence relation $R \subseteq X \times X$, the set of equivalence classes X/R with the quotient map $q : X \twoheadrightarrow X/R$ is the coequalizer of the projections $\pi, \pi' : R \rightrightarrows X$.

Coequalizers

- If $f, g : X \rightrightarrows Y$ are a parallel pair of arrows in $\mathcal{C}$, then their *coequalizer* is an object Q and arrow $q : Y \to Q$ such that

$$
\begin{tikzcd}
X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "q"] \arrow[dr, "k"'] & Q \arrow[d, dashed, "\exists! \beta"] \\
 & & Z.
\end{tikzcd}
$$

$$q \circ f = q \circ g$$
$$k \circ f = k \circ g$$
$$k = \beta \circ q.$$

- Given a set X and an equivalence relation $R \subseteq X \times X$, the set of equivalence classes X/R with the quotient map $q : X \twoheadrightarrow X/R$ is the coequalizer of the projections $\pi, \pi' : R \rightrightarrows X$.

Coequalizers

- If $f, g : X \rightrightarrows Y$ are a parallel pair of arrows in C, then their *coequalizer* is an object Q and arrow $q : Y \to Q$ such that

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "q"] \arrow[dr, "k"'] & Q \arrow[d, dashed, "\beta", "\exists!"'] \\ & & Z. \end{tikzcd}$$

$$q \circ f = q \circ g$$
$$k \circ f = k \circ g$$
$$k = \beta \circ q.$$

- Given a set X and an equivalence relation $R \subseteq X \times X$, the set of equivalence classes X/R with the quotient map $q : X \twoheadrightarrow X/R$ is the coequalizer of the projections $\pi, \pi' : R \rightrightarrows X$.

Coequalizers

- If $f, g : X \rightrightarrows Y$ are a parallel pair of arrows in $\mathcal{C}$, then their *coequalizer* is an object Q and arrow $q : Y \to Q$ such that

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "q"] \arrow[dr, "k"'] & Q \arrow[d, dashed, "\exists! \ \beta"] \\ & & Z. \end{tikzcd}$$

$$q \circ f = q \circ g$$
$$k \circ f = k \circ g$$
$$k = \beta \circ q.$$

- Given a set X and an equivalence relation $R \subseteq X \times X$, the set of equivalence classes X/R with the quotient map $q : X \twoheadrightarrow X/R$ is the coequalizer of the projections $\pi, \pi' : R \rightrightarrows X$.

Effective Epimorphisms

- Suppose $f : X \to Y$ is a surjection and let $R = \{(x_0, x_1) \in X \times X : f(x_0) = f(x_1)\}$ be an equivalence relation (notice that $R = X \times_Y X$).
- Then we have that $Y = X/R$, that is, Y is the coequalizer of the projection maps $\pi, \pi' : X \times_Y X \to X$.
- We say that a equivalence relation R is *effective* if it arises as the pullback $R = X \times_Y X$.
- Let C be a category which admits pullbacks, and suppose that $f : X \to Y$ is a morphism in C.
  Then f is an *effective epimorphism* if it exhibits Y as a coequalizer of the maps $\pi, \pi' : X \times_Y X \rightrightarrows X$.
- Every effective epimorphism is an epimorphism (in Set the converse is true).

Effective Epimorphisms

- Suppose $f : X \to Y$ is a surjection and let $R = \{(x_0, x_1) \in X \times X : f(x_0) = f(x_1)\}$ be an equivalence relation (notice that $R = X \times_Y X$).
- Then we have that $Y = X/R$, that is, Y is the coequalizer of the projection maps $\pi, \pi' : X \times_Y X \to X$.
- We say that a equivalence relation R is *effective* if it arises as the pullback $R = X \times_Y X$.
- Let C be a category which admits pullbacks, and suppose that $f : X \to Y$ is a morphism in C.
  Then f is an *effective epimorphism* if it exhibits Y as a coequalizer of the maps $\pi, \pi' : X \times_Y X \rightrightarrows X$.
- Every effective epimorphism is an epimorphism (in Set the converse is true).

Effective Epimorphisms

- Suppose $f : X \to Y$ is a surjection and let $R = \{(x_0, x_1) \in X \times X : f(x_0) = f(x_1)\}$ be an equivalence relation (notice that $R = X \times_Y X$).
- Then we have that $Y = X/R$, that is, Y is the coequalizer of the projection maps $\pi, \pi' : X \times_Y X \to X$.
- We say that a equivalence relation R is *effective* if it arises as the pullback $R = X \times_Y X$.
- Let $\mathcal{C}$ be a category which admits pullbacks, and suppose that $f : X \to Y$ is a morphism in $\mathcal{C}$.
  Then f is an *effective epimorphism* if it exhibits Y as a coequalizer of the maps $\pi, \pi' : X \times_Y X \rightrightarrows X$.
- Every effective epimorphism is an epimorphism (in Set the converse is true).

Effective Epimorphisms

- Suppose $f : X \to Y$ is a surjection and let $R = \{(x_0, x_1) \in X \times X : f(x_0) = f(x_1)\}$ be an equivalence relation (notice that $R = X \times_Y X$).
- Then we have that $Y = X/R$, that is, Y is the coequalizer of the projection maps $\pi, \pi' : X \times_Y X \to X$.
- We say that a equivalence relation R is *effective* if it arises as the pullback $R = X \times_Y X$.
- Let C be a category which admits pullbacks, and suppose that $f : X \to Y$ is a morphism in C.
  Then f is an *effective epimorphism* if it exhibits Y as a coequalizer of the maps $\pi, \pi' : X \times_Y X \rightrightarrows X$.
- Every effective epimorphism is an epimorphism (in Set the converse is true).

Effective Epimorphisms

- Suppose $f : X \to Y$ is a surjection and let $R = \{(x_0, x_1) \in X \times X : f(x_0) = f(x_1)\}$ be an equivalence relation (notice that $R = X \times_Y X$).
- Then we have that $Y = X/R$, that is, Y is the coequalizer of the projection maps $\pi, \pi' : X \times_Y X \to X$.
- We say that a equivalence relation R is *effective* if it arises as the pullback $R = X \times_Y X$.
- Let C be a category which admits pullbacks, and suppose that $f : X \to Y$ is a morphism in C.
  Then f is an *effective epimorphism* if it exhibits Y as a coequalizer of the maps $\pi, \pi' : X \times_Y X \rightrightarrows X$.
- Every effective epimorphism is an epimorphism (in Set the converse is true).

Effective Epimorphisms

- We say that the collection of effective epimorphisms in C is *closed under pullbacks* if, in C we have the following pullback square

$$\begin{tikzcd} X' \arrow[r] \arrow[d, "f'"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ Y' \arrow[r] & Y, \end{tikzcd}$$

  if f is an effective epi, then f' is also an effective epi.
- Let $f : X \to Y$ be a set function. Then we can (uniquely) *factor* f into a surjection g and an injection h:

$$\begin{tikzcd} X \arrow[r, two heads, "g"'] \arrow[rr, bend left, "f"] & \mathrm{im}(f) \arrow[r, tail, "h"'] & Y. \end{tikzcd}$$

- In general (that is in Set and $\mathrm{Syn}(T)$), we generalize this *factorization* into an effective epimorphism and a monomorphism.

Effective Epimorphisms

- We say that the collection of effective epimorphisms in C is *closed under pullbacks* if, in C we have the following pullback square

$$\begin{tikzcd} X' \arrow[r] \arrow[d, "f'"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ Y' \arrow[r] & Y, \end{tikzcd}$$

if f is an effective epi, then f' is also an effective epi.

- Let $f : X \to Y$ be a set function. Then we can (uniquely) *factor* f into a surjection g and an injection h:

$$\begin{tikzcd} X \arrow[r, two heads, "g"'] \arrow[rr, bend left, "f"] & \mathrm{im}(f) \arrow[r, tail, "h"'] & Y. \end{tikzcd}$$

- In general (that is in Set and $\mathsf{Syn}(T)$), we generalize this *factorization* into an effective epimorphism and a monomorphism.

Effective Epimorphisms

- We say that the collection of effective epimorphisms in C is *closed under pullbacks* if, in C we have the following pullback square

$$\begin{tikzcd} X' \arrow[r] \arrow[d, "f'"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ Y' \arrow[r] & Y, \end{tikzcd}$$

  if f is an effective epi, then f' is also an effective epi.
- Let $f : X \to Y$ be a set function. Then we can (uniquely) *factor* f into a surjection g and an injection h:

$$\begin{tikzcd} X \arrow[r, two heads, "g"'] \arrow[rr, bend left, "f"] & \mathrm{im}(f) \arrow[r, tail, "h"'] & Y. \end{tikzcd}$$

- In general (that is in Set and $\mathsf{Syn}(T)$), we generalize this *factorization* into an effective epimorphism and a monomorphism.

Coherent Category

- A *coherent category* C is a category satisfying the axioms:
  - The category C has finite limits.
  - For every object $X \in \mathsf{C}_0$, the poset $\mathsf{Sub}(X)$ is an upper semilattice.
  - For every morphism $f : X \to Y$ in C, the inverse map $f^{-1} : \mathsf{Sub}(Y) \to \mathsf{Sub}(X)$ is a morphism of upper semilattices.
  - Every morphism $f : X \to Z$ in C can be written as a composition $X \overset{g}{\twoheadrightarrow} \mathrm{im}(f) \overset{h}{\rightarrowtail} Z$, where g is an effective epimorphism and h is a monomorphism.
  - The collection of effective epimorphisms in C is closed under pullbacks.
- Set and $\mathsf{Syn}(T)$ are coherent categories.

Coherent Category

- A *coherent category* C is a category satisfying the axioms:
  - The category C has finite limits.
  - For every object $X \in \mathsf{C}_0$, the poset $\mathsf{Sub}(X)$ is an upper semilattice.
  - For every morphism $f : X \to Y$ in C, the inverse map $f^{-1} : \mathsf{Sub}(Y) \to \mathsf{Sub}(X)$ is a morphism of upper semilattices.
  - Every morphism $f : X \to Z$ in C can be written as a composition $X \overset{g}{\twoheadrightarrow} \mathrm{im}(f) \overset{h}{\rightarrowtail} Z$, where g is an effective epimorphism and h is a monomorphism.
  - The collection of effective epimorphisms in C is closed under pullbacks.
- Set and $\mathsf{Syn}(T)$ are coherent categories.

Coherent Functor

- Let C and D be coherent categories. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *coherent functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - For every object $X \in \mathsf{C}_0$ the induced map $\mathrm{Sub}(X) \to \mathrm{Sub}(F(X))$ is a homomorphism of upper semilattices.
- The composition of coherent functors is a coherent functor.
- We let $\mathrm{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$ denote the full subcategory of $\mathrm{Fun}(\mathsf{C}, \mathsf{D})$ whose objects are coherent functors.

Coherent Functor

- Let C and D be coherent categories. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *coherent functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - For every object $X \in \mathsf{C}_0$ the induced map $\mathsf{Sub}(X) \to \mathsf{Sub}(F(X))$ is a homomorphism of upper semilattices.
- The composition of coherent functors is a coherent functor.
- We let $\mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$ denote the full subcategory of $\mathsf{Fun}(\mathsf{C}, \mathsf{D})$ whose objects are coherent functors.

Coherent Functor

- Let C and D be coherent categories. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *coherent functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - For every object $X \in \mathsf{C}_0$ the induced map $\mathsf{Sub}(X) \to \mathsf{Sub}(F(X))$ is a homomorphism of upper semilattices.
- The composition of coherent functors is a coherent functor.
- We let $\mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$ denote the full subcategory of $\mathsf{Fun}(\mathsf{C}, \mathsf{D})$ whose objects are coherent functors.

Coherent Functor

- For any first-order theory T, we have the models of T are exactly the coherent functors $\mathrm{Syn}(T) \to \mathrm{Set}$.
- In this case, we will say, for any coherent category C, the *models of* C are the coherent functors $M : \mathsf{C} \to \mathrm{Set}$.
- We denote by

$$\mathrm{Mod}(\mathsf{C}) = \mathrm{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathrm{Set})$$

  the *category of models of* C.
- $\mathrm{Mod}(\mathrm{Syn}(T))$ is equivalent to $\mathrm{Mod}(T)$.

Coherent Functor

- For any first-order theory T, we have the models of T are exactly the coherent functors $\mathsf{Syn}(T) \to \mathsf{Set}$.
- In this case, we will say, for any coherent category C, the *models of* C are the coherent functors $M : \mathsf{C} \to \mathsf{Set}$.
- We denote by

$$\mathsf{Mod}(\mathsf{C}) = \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{Set})$$

the *category of models of* C.
- $\mathsf{Mod}(\mathsf{Syn}(T))$ is equivalent to $\mathsf{Mod}(T)$.

Coherent Functor

- For any first-order theory T, we have the models of T are exactly the coherent functors $\mathsf{Syn}(T) \to \mathsf{Set}$.
- In this case, we will say, for any coherent category C, the *models of* C are the coherent functors $M : \mathsf{C} \to \mathsf{Set}$.
- We denote by

$$\mathsf{Mod}(\mathsf{C}) = \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{Set})$$

  the *category of models of* C.
- $\mathsf{Mod}(\mathsf{Syn}(T))$ is equivalent to $\mathsf{Mod}(T)$.

Coherent Functor

- For any first-order theory T, we have the models of T are exactly the coherent functors $\mathsf{Syn}(T) \to \mathsf{Set}$.
- In this case, we will say, for any coherent category C, the *models of* C are the coherent functors $M : \mathsf{C} \to \mathsf{Set}$.
- We denote by

$$\mathsf{Mod}(\mathsf{C}) = \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{Set})$$

  the *category of models of* C.
- $\mathsf{Mod}(\mathsf{Syn}(T))$ is equivalent to $\mathsf{Mod}(T)$.

Example

- Consider two theories T and T' defined as follows:
  - The language of T has no symbols and T has a single axiom $\exists ! x.(x = x)$.
  - The language of T' has a single 1-ary relation R and a pair of axioms $\exists ! y.R(y)$ and $\exists ! z.\neg R(z)$.
- We have that (up to isomorphism) T and T' have a single model, $M = \{x\}$ and $M' = \{y, z\}$.
- So $\mathrm{Mod}(T)$ is equivalent to $\mathrm{Mod}(T')$.
- But, $\mathrm{Syn}(T)$ is equivalent to the poset $\{0 \leq 1\} = \mathbb{2}$ while $\mathrm{Syn}(T')$ is equivalent to Finset.
- The structure of a coherent category is not enough. We will want to preform a *pretopos completion* on the syntactic category.

Example

- Consider two theories T and T' defined as follows:
  - The language of T has no symbols and T has a single axiom $\exists! x.(x = x)$.
  - The language of T' has a single 1-ary relation R and a pair of axioms $\exists! y.R(y)$ and $\exists! z.\neg R(z)$.
- We have that (up to isomorphism) T and T' have a single model, $M = \{x\}$ and $M' = \{y, z\}$.
- So $\mathrm{Mod}(T)$ is equivalent to $\mathrm{Mod}(T')$.
- But, $\mathrm{Syn}(T)$ is equivalent to the poset $\{0 \leq 1\} = \mathbb{2}$ while $\mathrm{Syn}(T')$ is equivalent to Finset.
- The structure of a coherent category is not enough. We will want to preform a *pretopos completion* on the syntactic category.

Example

- Consider two theories T and T' defined as follows:
  - The language of T has no symbols and T has a single axiom $\exists! x.(x = x)$.
  - The language of T' has a single 1-ary relation R and a pair of axioms $\exists! y.R(y)$ and $\exists! z.\neg R(z)$.
- We have that (up to isomorphism) T and T' have a single model, $M = \{x\}$ and $M' = \{y, z\}$.
- So $\mathrm{Mod}(T)$ is equivalent to $\mathrm{Mod}(T')$.
- But, $\mathrm{Syn}(T)$ is equivalent to the poset $\{0 \leq 1\} = \mathbb{2}$ while $\mathrm{Syn}(T')$ is equivalent to Finset.
- The structure of a coherent category is not enough. We will want to preform a *pretopos completion* on the syntactic category.

Example

- Consider two theories T and T' defined as follows:
  - The language of T has no symbols and T has a single axiom $\exists! x.(x = x)$.
  - The language of T' has a single 1-ary relation R and a pair of axioms $\exists! y.R(y)$ and $\exists! z.\neg R(z)$.
- We have that (up to isomorphism) T and T' have a single model, $M = \{x\}$ and $M' = \{y, z\}$.
- So $\mathrm{Mod}(T)$ is equivalent to $\mathrm{Mod}(T')$.
- But, $\mathrm{Syn}(T)$ is equivalent to the poset $\{0 \leq 1\} = 2$ while $\mathrm{Syn}(T')$ is equivalent to Finset.
- The structure of a coherent category is not enough. We will want to preform a *pretopos completion* on the syntactic category.

Example

- Consider two theories T and T' defined as follows:
  - The language of T has no symbols and T has a single axiom $\exists! x.(x = x)$.
  - The language of T' has a single 1-ary relation R and a pair of axioms $\exists! y.R(y)$ and $\exists! z.\neg R(z)$.
- We have that (up to isomorphism) T and T' have a single model, $M = \{x\}$ and $M' = \{y, z\}$.
- So $\mathrm{Mod}(T)$ is equivalent to $\mathrm{Mod}(T')$.
- But, $\mathrm{Syn}(T)$ is equivalent to the poset $\{0 \leq 1\} = 2$ while $\mathrm{Syn}(T')$ is equivalent to Finset.
- The structure of a coherent category is not enough. We will want to preform a *pretopos completion* on the syntactic category.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x, 0) : x \in X\}, \quad Y' = \{(y, 1) : y \in Y\},$$

with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.

- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If C is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in C_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of C.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset 2.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x, 0) : x \in X\}, \quad Y' = \{(y, 1) : y \in Y\},$$

with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.

- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If C is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in C_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of C.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset $\mathbf{2}$.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x,0) : x \in X\}\,, \quad Y' = \{(y,1) : y \in Y\}\,,$$

  with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.
- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If $\mathbb{C}$ is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in \mathbb{C}_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of $\mathbb{C}$.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset $\mathbf{2}$.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x, 0) : x \in X\}, \quad Y' = \{(y, 1) : y \in Y\},$$

  with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.
- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If C is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in \mathsf{C}_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of C.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset $\mathbb{2}$.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x, 0) : x \in X\}, \quad Y' = \{(y, 1) : y \in Y\},$$

  with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.
- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If C is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in \mathsf{C}_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of C.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset $\mathbf{2}$.

Disjoint Coproducts

- Given two sets X, Y, we can form their *disjoint union* by creating isomorphic copies of X and Y:

$$X' = \{(x, 0) : x \in X\}, \quad Y' = \{(y, 1) : y \in Y\},$$

  with $X' \cap Y' = \emptyset$, and taking the union: $X \sqcup Y = X' \cup Y'$.
- This is the *coproduct* in Set.
- In general, coproducts are not "disjoint."
- If C is a category with pullbacks, then the coproduct $X + Y$ of $X, Y \in \mathsf{C}_0$ (if it exists) is *disjoint* if
  - The injection maps $i : X \to X + Y$, $i' : Y \to X + Y$ are monomorphism.
  - The *intersection* (*meet*) $X \times_{X+Y} Y$ of X and Y is an initial object of C.
- In the case $X_0, X_1 \in \mathrm{Sub}(X)$ such that $X_0 \wedge X_1 = \emptyset$, then $X_0 \vee X_1$ is their disjoint coproduct.
- We have disjoint coproducts in Finset but not in the poset $\mathbb{2}$.

Pretopos

- Let C be a category. We say that C is a *pretopos* if it satisfies the following axioms:
  - The category C admits finite limits.
  - The category C admits finite coproducts, and coproducts are disjoint.
  - The formation of finite coproducts in C is preserved by pullbacks.
  - Every equivalence relation if effective.
  - The collection of effective epimorphisms in C is closed under pullbacks.
- Every pretopos is a coherent category.
- Set is a pretopos. Syn(T) is NOT in general.

Pretopos

- Let C be a category. We say that C is a *pretopos* if it satisfies the following axioms:
  - The category C admits finite limits.
  - The category C admits finite coproducts, and coproducts are disjoint.
  - The formation of finite coproducts in C is preserved by pullbacks.
  - Every equivalence relation if effective.
  - The collection of effective epimorphisms in C is closed under pullbacks.
- Every pretopos is a coherent category.
- Set is a pretopos. $\mathrm{Syn}(T)$ is NOT in general.

Pretopos

- Let C be a category. We say that C is a *pretopos* if it satisfies the following axioms:
  - The category C admits finite limits.
  - The category C admits finite coproducts, and coproducts are disjoint.
  - The formation of finite coproducts in C is preserved by pullbacks.
  - Every equivalence relation if effective.
  - The collection of effective epimorphisms in C is closed under pullbacks.
- Every pretopos is a coherent category.
- Set is a pretopos. $\mathsf{Syn}(T)$ is NOT in general.

Pretopos Functor

- Let C and D be pretoposes. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *pretopos functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - F preserves finite coproducts.
- The composition of pretopos functors is a pretopos functor.
- A pretopos functor $M : \mathsf{C} \to \mathsf{Set}$ is called a *model* of C and $\mathrm{Mod}(\mathsf{C})$ is the full subcategory of $\mathrm{Fun}(\mathsf{C}, \mathsf{Set})$ whose objects are models.

Pretopos Functor

- Let C and D be pretoposes. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *pretopos functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - F preserves finite coproducts.
- The composition of pretopos functors is a pretopos functor.
- A pretopos functor $M : \mathsf{C} \to \mathsf{Set}$ is called a *model* of C and $\mathrm{Mod}(\mathsf{C})$ is the full subcategory of $\mathrm{Fun}(\mathsf{C}, \mathsf{Set})$ whose objects are models.

Pretopos Functor

- Let C and D be pretoposes. A functor $F : \mathsf{C} \to \mathsf{D}$ is a *pretopos functor* if it satisfies the following:
  - F preserves finite limits.
  - F carries effective epimorphism in C to effective epimorphisms in D.
  - F preserves finite coproducts.
- The composition of pretopos functors is a pretopos functor.
- A pretopos functor $M : \mathsf{C} \to \mathsf{Set}$ is called a *model* of C and $\mathsf{Mod}(\mathsf{C})$ is the full subcategory of $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$ whose objects are models.

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\lambda \circ : \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) \to \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$$
$$(F : \mathsf{C}^{eq} \to \mathsf{D}) \mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D})$$

which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda \circ : \mathsf{Mod}(\mathsf{C}^{eq}) \to \mathsf{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathsf{Syn}(T)$ is a pretopos.
- We will denote by $\mathsf{Syn}^{eq}(T)$ the pretopos completion of $\mathsf{Syn}(T)$.

Pretopos Completion

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\lambda\circ : \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) \to \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$$
$$(F : \mathsf{C}^{eq} \to \mathsf{D}) \mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D})$$

  which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda\circ : \mathsf{Mod}(\mathsf{C}^{eq}) \to \mathsf{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathsf{Syn}(T)$ is a pretopos.
- We will denote by $\mathsf{Syn}^{eq}(T)$ the pretopos completion of $\mathsf{Syn}(T)$.

Pretopos Completion

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\begin{aligned} \lambda\circ : \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) &\to \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D}) \\ (F : \mathsf{C}^{eq} \to \mathsf{D}) &\mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D}) \end{aligned}$$

  which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda\circ : \mathsf{Mod}(\mathsf{C}^{eq}) \to \mathsf{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathsf{Syn}(T)$ is a pretopos.
- We will denote by $\mathsf{Syn}^{eq}(T)$ the pretopos completion of $\mathsf{Syn}(T)$.

Pretopos Completion

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\lambda\circ : \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) \to \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$$
$$(F : \mathsf{C}^{eq} \to \mathsf{D}) \mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D})$$

  which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda\circ : \mathsf{Mod}(\mathsf{C}^{eq}) \to \mathsf{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathsf{Syn}(T)$ is a pretopos.
- We will denote by $\mathsf{Syn}^{eq}(T)$ the pretopos completion of $\mathsf{Syn}(T)$.

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\lambda\circ : \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) \to \mathsf{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$$
$$(F : \mathsf{C}^{eq} \to \mathsf{D}) \mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D})$$

  which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda\circ : \mathsf{Mod}(\mathsf{C}^{eq}) \to \mathsf{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathsf{Syn}(T)$ is a pretopos.
- We will denote by $\mathsf{Syn}^{eq}(T)$ the pretopos completion of $\mathsf{Syn}(T)$.

Pretopos Completion

- Given a small coherent category C, there exists a small pretopos C^{eq} and a coherent functor $\lambda : \mathsf{C} \to \mathsf{C}^{eq}$ with the property that, for any other pretopos D, composition with λ induces a functor

$$\lambda\circ : \mathrm{Fun}^{\mathrm{coh}}(\mathsf{C}^{eq}, \mathsf{D}) \to \mathrm{Fun}^{\mathrm{coh}}(\mathsf{C}, \mathsf{D})$$
$$(F : \mathsf{C}^{eq} \to \mathsf{D}) \mapsto (\lambda \circ F : \mathsf{C} \to \mathsf{D})$$

  which is an equivalence of categories.
- If $\mathsf{D} = \mathsf{Set}$, then $\lambda\circ : \mathrm{Mod}(\mathsf{C}^{eq}) \to \mathrm{Mod}(\mathsf{C})$ is an equivalence of categories.
- This process is called the *pretopos completion* of C.
- Freely adjoining finite coproducts and coequalizers of equivalence relations to C.
- We say a (typed) first-order theory T *eliminates imaginaries* if $\mathrm{Syn}(T)$ is a pretopos.
- We will denote by $\mathrm{Syn}^{eq}(T)$ the pretopos completion of $\mathrm{Syn}(T)$.

Recap

- Pretoposes ($\mathrm{Syn}^{eq}(T)$) have the correct structure to do the syntax of a theory T.
- The models of a theory can be represented as pretopos functors from $\mathrm{Syn}^{eq}(T)$ to Set.
- Ultraproducts in C can be formed in a larger category C^{+}.
- We have the ultraproduct functor

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathrm{C}^S \to \mathrm{C}.$$

- We have everything we need for Łoś Ultraproduct Theorem.

Recap

- Pretoposes ($\mathrm{Syn}^{eq}(T)$) have the correct structure to do the syntax of a theory T.
- The models of a theory can be represented as pretopos functors from $\mathrm{Syn}^{eq}(T)$ to Set.
- Ultraproducts in C can be formed in a larger category C^+.
- We have the ultraproduct functor

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathsf{C}^S \to \mathsf{C}.$$

- We have everything we need for Łoś Ultraproduct Theorem.

Recap

- Pretoposes ($\mathrm{Syn}^{eq}(T)$) have the correct structure to do the syntax of a theory T.
- The models of a theory can be represented as pretopos functors from $\mathrm{Syn}^{eq}(T)$ to Set.
- Ultraproducts in C can be formed in a larger category C^{+}.
- We have the ultraproduct functor

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathrm{C}^S \to \mathrm{C}.$$

- We have everything we need for Łoś Ultraproduct Theorem.

Recap

- Pretoposes ($\mathrm{Syn}^{eq}(T)$) have the correct structure to do the syntax of a theory T.
- The models of a theory can be represented as pretopos functors from $\mathrm{Syn}^{eq}(T)$ to Set.
- Ultraproducts in C can be formed in a larger category C^+.
- We have the ultraproduct functor

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathrm{C}^S \to \mathrm{C}.$$

- We have everything we need for Łoś Ultraproduct Theorem.

Recap

- Pretoposes ($\mathrm{Syn}^{eq}(T)$) have the correct structure to do the syntax of a theory T.
- The models of a theory can be represented as pretopos functors from $\mathrm{Syn}^{eq}(T)$ to Set.
- Ultraproducts in C can be formed in a larger category C^{+}.
- We have the ultraproduct functor

$$\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} (\bullet) : \mathrm{C}^{S} \to \mathrm{C}.$$

- We have everything we need for Łoś Ultraproduct Theorem.

Łoś Ultraproduct Theorem

Łoś Ultraproduct Theorem

Let C be a pretopos and let $\{M_s : \mathsf{C} \to \mathsf{Set}\}_{s\in S}$ be a collection of models of C. For every ultrafilter $\mathcal{U}$ on S, the ultraproduct $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s$ (formed in the category $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$) is a model of C.

- The ultraproduct $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s$ will be a pretopos functor from C to Set, which can be defined explicitly as

$$\left(\operatorname*{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s\right)(X) = \operatorname*{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s(X).$$

Lemma

Let S be a set and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct functor $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0}(\bullet) : \mathsf{Set}^S \to \mathsf{Set}$ *is a pretopos functor.*

Łoś Ultraproduct Theorem

Łoś Ultraproduct Theorem

Let C be a pretopos and let $\{M_s : \mathsf{C} \to \mathsf{Set}\}_{s \in S}$ be a collection of models of C. For every ultrafilter $\mathcal{U}$ on S, the ultraproduct $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ (formed in the category $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$) is a model of C.

- The ultraproduct $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ will be a pretopos functor from C to Set, which can be defined explicitly as

$$\left(\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \right)(X) = \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s(X).$$

Lemma

Let S be a set and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct functor $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0}(\bullet) : \mathsf{Set}^S \to \mathsf{Set}$ *is a pretopos functor.*

Łoś Ultraproduct Theorem

Let C be a pretopos and let $\{M_s : \mathsf{C} \to \mathsf{Set}\}_{s \in S}$ be a collection of models of C. For every ultrafilter $\mathcal{U}$ on S, the ultraproduct $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ (formed in the category $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$) is a model of C.

- The ultraproduct $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ will be a pretopos functor from C to Set, which can be defined explicitly as

$$\left(\operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s \right)(X) = \operatorname*{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s(X).$$

Lemma

Let S be a set and let $\mathcal{U}$ be an ultrafilter on S. Then the ultraproduct functor $\operatorname{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0}(\bullet) : \mathsf{Set}^S \to \mathsf{Set}$ is a pretopos functor.

Proof.

Let $\{M_s : \mathsf{C} \to \mathsf{Set}\}_{s \in S}$ be a collection of models of C and let $\mathcal{U}$ be an ultrafilter on the index set S. Notice that we can write the ultraproduct $\mathrm{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s : \mathsf{C} \to \mathsf{Set}$ of the models of C as the following composition

$$\mathsf{C} \xrightarrow{\{M_s\}_{s \in S}} \mathsf{Set}^S \xrightarrow{\underset{S_0 \in \mathcal{U}^{\mathrm{op}}}{\mathrm{colim}} \prod_{s \in S_0} (\bullet)} \mathsf{Set} \ .$$

Since each M_s is a model of C, that is, a pretopos functor, we have that $\{M_s\}_{s \in S}$ is a pretopos functor. Also, by the Lemma, we have the second map is a pretopos functor. Then we have that $\mathrm{colim}_{S_0 \in \mathcal{U}^{\mathrm{op}}} \prod_{s \in S_0} M_s$ is a pretopos functor and hence a model of C. $\square$

Classical Łoś Ultraproduct Theorem

Let T be a first-order theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Then the ultraproduct $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}}\prod_{s\in S_0} M_s$ is also a model of T.

Proof.

Apply Łoś Ultraproduct Theorem to the pretopos completed syntactic category $\mathrm{Syn}^{eq}(T)$. □

Classical Łoś Ultraproduct Theorem

Classical Łoś Ultraproduct Theorem

Let T be a first-order theory and let $\{M_s\}_{s\in S}$ be a collection of models of T. Then the ultraproduct $\operatorname{colim}_{S_0\in\mathcal{U}^{\mathrm{op}}} \prod_{s\in S_0} M_s$ is also a model of T.

Proof.

Apply Łoś Ultraproduct Theorem to the pretopos completed syntactic category $\mathrm{Syn}^{eq}(T)$. $\square$

Ultracategory

- What Łoś Ultraproduct Theorem shows is that $\mathsf{Mod}(\mathsf{C})$ has ultraproducts in $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathsf{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathsf{Fun}^{\mathsf{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

- What Łoś Ultraproduct Theorem shows is that $\mathsf{Mod}(\mathsf{C})$ has ultraproducts in $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathsf{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathsf{Fun}^{\mathsf{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

Ultracategory

- What Łoś Ultraproduct Theorem shows is that $\mathsf{Mod}(\mathsf{C})$ has ultraproducts in $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathsf{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathsf{Fun}^{\mathsf{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

Ultracategory

- What Łoś Ultraproduct Theorem shows is that $\mathsf{Mod}(\mathsf{C})$ has ultraproducts in $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathsf{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathsf{Fun}^{\mathsf{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

Ultracategory

- What Łoś Ultraproduct Theorem shows is that $\mathrm{Mod}(\mathsf{C})$ has ultraproducts in $\mathrm{Fun}(\mathsf{C}, \mathrm{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathrm{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathrm{Fun}^{\mathrm{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

Ultracategory

- What Łoś Ultraproduct Theorem shows is that $\mathsf{Mod}(\mathsf{C})$ has ultraproducts in $\mathsf{Fun}(\mathsf{C}, \mathsf{Set})$.
- Intuitively (and informally), an *ultracategory* is a category with an "ultraproduct" structure, that is, three collections of different types of ultraproduct functors satisfying some axioms.
- $\mathsf{Mod}(\mathsf{C})$ is an ultracategory.
- Set is both a pretopos and an ultracategory.
- An *ultrafunctor* is a functor between ultracategories which preserves the "ultraproduct" structure.
- We let $\mathsf{Fun}^{\mathsf{Ult}}(\mathsf{C}, \mathsf{D})$ denote the category of ultrafunctors between C and D.

Conceptual Completeness

- For each object X in a pretopos C, define the *evaluation functor* $\mathrm{ev}_X : \mathrm{Mod}(\mathsf{C}) \to \mathrm{Set}$ by $\mathrm{ev}_X(M) = M(X)$.
- We have that ev_X is an ultrafunctor and the map $X \mapsto \mathrm{ev}_X$ determines a functor $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$.

Makkai's Strong Conceptual Completeness

Let C be a small pretopos. Then the evaluation map $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$ is an equivalence of categories.

Makkai-Reyes Conceptual Completeness Theorem

Let C and D be small pretoposes and $\lambda : \mathsf{C} \to \mathsf{D}$ be a pretopos functor. Consider the map induced by precomposition with λ,

$$\circ\lambda : \mathrm{Mod}(\mathsf{D}) \to \mathrm{Mod}(\mathsf{C}) :: (F : \mathsf{D} \to \mathrm{Set}) \mapsto (F \circ \lambda : \mathsf{C} \to \mathrm{Set}),$$

If $\circ\lambda$ is an equivalence of categories, then so is λ.

Conceptual Completeness

- For each object X in a pretopos C, define the *evaluation functor* $\mathrm{ev}_X : \mathrm{Mod}(\mathsf{C}) \to \mathsf{Set}$ by $\mathrm{ev}_X(M) = M(X)$.
- We have that ev_X is an ultrafunctor and the map $X \mapsto \mathrm{ev}_X$ determines a functor $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathsf{Set})$.

Makkai's Strong Conceptual Completeness

Let C be a small pretopos. Then the evaluation map $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathsf{Set})$ is an equivalence of categories.

Makkai-Reyes Conceptual Completeness Theorem

Let C and D be small pretoposes and $\lambda : \mathsf{C} \to \mathsf{D}$ be a pretopos functor. Consider the map induced by precomposition with λ,

$$\circ\lambda : \mathrm{Mod}(\mathsf{D}) \to \mathrm{Mod}(\mathsf{C}) :: (F : \mathsf{D} \to \mathsf{Set}) \mapsto (F \circ \lambda : \mathsf{C} \to \mathsf{Set}),$$

If $\circ\lambda$ is an equivalence of categories, then so is λ.

Conceptual Completeness

- For each object X in a pretopos C, define the *evaluation functor* $\mathrm{ev}_X : \mathrm{Mod}(\mathsf{C}) \to \mathrm{Set}$ by $\mathrm{ev}_X(M) = M(X)$.
- We have that ev_X is an ultrafunctor and the map $X \mapsto \mathrm{ev}_X$ determines a functor $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$.

Makkai's Strong Conceptual Completeness

Let C be a small pretopos. Then the evaluation map $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$ is an equivalence of categories.

Makkai-Reyes Conceptual Completeness Theorem

Let C and D be small pretoposes and $\lambda : \mathsf{C} \to \mathsf{D}$ be a pretopos functor. Consider the map induced by precomposition with λ,

$$\circ\lambda : \mathrm{Mod}(\mathsf{D}) \to \mathrm{Mod}(\mathsf{C}) :: (F : \mathsf{D} \to \mathrm{Set}) \mapsto (F \circ \lambda : \mathsf{C} \to \mathrm{Set}),$$

If $\circ\lambda$ is an equivalence of categories, then so is λ.

Conceptual Completeness

- For each object X in a pretopos C, define the *evaluation functor* $\mathrm{ev}_X : \mathrm{Mod}(\mathsf{C}) \to \mathrm{Set}$ by $\mathrm{ev}_X(M) = M(X)$.
- We have that ev_X is an ultrafunctor and the map $X \mapsto \mathrm{ev}_X$ determines a functor $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$.

Makkai's Strong Conceptual Completeness

Let C be a small pretopos. Then the evaluation map $\mathrm{ev} : \mathsf{C} \to \mathrm{Fun}^{\mathrm{Ult}}(\mathrm{Mod}(\mathsf{C}), \mathrm{Set})$ is an equivalence of categories.

Makkai-Reyes Conceptual Completeness Theorem

Let C and D be small pretoposes and $\lambda : \mathsf{C} \to \mathsf{D}$ be a pretopos functor. Consider the map induced by precomposition with λ,

$$\circ\lambda : \mathrm{Mod}(\mathsf{D}) \to \mathrm{Mod}(\mathsf{C}) :: (F : \mathsf{D} \to \mathrm{Set}) \mapsto (F \circ \lambda : \mathsf{C} \to \mathrm{Set}),$$

If $\circ\lambda$ is an equivalence of categories, then so is λ.

Conclusion

- Pretoposes: syntax $\mathrm{Syn}(T)$
- Ultracategories: semantics $\mathrm{Mod}(\mathrm{Syn}(T))$.

The End

Thank You!

Questions?