Proof Theory, Logic and Algebra

Amir Akbar Tabatabai
Bernoulli institute, University of Groningen

TACL 2022, Praia de Mira

"It is equally stupid and simple to consider mathematics to be just an axiom system as it is to see a tree as nothing but a quantity of planks." L.E.J. Brouwer

What is a proof?

- In the first part, we saw three different formalizations of the intuitionistic proofs. They were all syntactical objects constructed in syntactical calculi. Which one is a proof?

What is a proof?

- In the first part, we saw three different formalizations of the intuitionistic proofs. They were all syntactical objects constructed in syntactical calculi. Which one is a proof?
- We have the same problem with computation? Which realization should we choose? Or maybe they are just different realizations of one abstract notion. We don't know that notion! Right? Why?

What is a proof?

- In the first part, we saw three different formalizations of the intuitionistic proofs. They were all syntactical objects constructed in syntactical calculi. Which one is a proof?
- We have the same problem with computation? Which realization should we choose? Or maybe they are just different realizations of one abstract notion. We don't know that notion! Right? Why?
- Proofs have never been the main citizens in logic, even in the proof theory itself.

What is a proof?

- In the first part, we saw three different formalizations of the intuitionistic proofs. They were all syntactical objects constructed in syntactical calculi. Which one is a proof?
- We have the same problem with computation? Which realization should we choose? Or maybe they are just different realizations of one abstract notion. We don't know that notion! Right? Why?
- Proofs have never been the main citizens in logic, even in the proof theory itself. (It is not quite true, but still).

What is a proof?

- In the first part, we saw three different formalizations of the intuitionistic proofs. They were all syntactical objects constructed in syntactical calculi. Which one is a proof?
- We have the same problem with computation? Which realization should we choose? Or maybe they are just different realizations of one abstract notion. We don't know that notion! Right? Why?
- Proofs have never been the main citizens in logic, even in the proof theory itself. (It is not quite true, but still). We are usually interested in provability and not the proofs. We need proofs just to help us to prove some theorems about the theories, e.g. the consistency, the interpolation, the admissible rules (DP), etc. They are not like groups in mathematics. The situation is similar to the state of the art in computability theory before the interest in algorithm design. What was important was computability not the computation.

Proof Equivalence

- As a consequence of this lack of interest, we do not have a reasonable answer to even the trivial questions! For instance, think about the proof equivalence. Are the following proofs equivalent?

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D} & \mathcal{D}' \\ A & B\end{array}}{A \wedge B}\wedge I}{A}\wedge E_1 \qquad\qquad \begin{array}{c}\mathcal{D} \\ A\end{array}$$

Proof Equivalence

- As a consequence of this lack of interest, we do not have a reasonable answer to even the trivial questions! For instance, think about the proof equivalence. Are the following proofs equivalent?

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D} & \mathcal{D}' \\ A & B\end{array}}{A \wedge B}\,\wedge I}{A}\,\wedge E_1 \qquad\qquad \begin{array}{c}\mathcal{D} \\ A\end{array}$$

- What about the following trees?

$$\dfrac{A \wedge A}{A}\,\wedge_1 E \qquad \dfrac{A \wedge A}{A}\,\wedge_2 E \qquad \dfrac{\dfrac{}{A, A \Rightarrow A}\,axiom}{A \wedge A \Rightarrow A}\,L\wedge$$

Note that it is not clear how to compare proofs in different systems.

Proof Equivalence

- As a consequence of this lack of interest, we do not have a reasonable answer to even the trivial questions! For instance, think about the proof equivalence. Are the following proofs equivalent?

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D} & \mathcal{D}' \\ A & B\end{array}}{A \wedge B}\wedge I}{A}\wedge E_1 \qquad\qquad \begin{array}{c}\mathcal{D} \\ A\end{array}$$

- What about the following trees?

$$\dfrac{A \wedge A}{A}\wedge_1 E \qquad \dfrac{A \wedge A}{A}\wedge_2 E \qquad \dfrac{\dfrac{}{A, A \Rightarrow A}\,axiom}{A \wedge A \Rightarrow A}\,L\wedge$$

  Note that it is not clear how to compare proofs in different systems.

- When are two statements equivalent? For instance, whether $A \wedge B$ and $B \wedge A$ are equivalent? What about A and $A \wedge A$?

Equivalence between Formulas

- Usually people say that A and B are equivalent if each one implies the other. This means that A and B are equivalent iff there is a proof π of B from A and also a proof σ of A from B.

Equivalence between Formulas

- Usually people say that A and B are equivalent if each one implies the other. This means that A and B are equivalent iff there is a proof π of B from A and also a proof σ of A from B. Shouldn't these π and σ be somehow related?

Equivalence between Formulas

- Usually people say that A and B are equivalent if each one implies the other. This means that A and B are equivalent iff there is a proof π of B from A and also a proof σ of A from B. Shouldn't these π and σ be somehow related?
- Compare it to the following situation. When two sets are equivalent? The existence of one map in each direction between the two, or the existence of a bijection? Now, what if the sets are the sets of proofs for A and B. If A and B are equivalent, shouldn't these sets be in one-to-one correspondence? In this sense, in addition to the existence of π and σ, we also need them to be the inverse of each other, i.e., $\pi\sigma = id$ and $\sigma\pi = id$. Hence, we need to understand the equality!

Equivalence between Formulas

- Usually people say that A and B are equivalent if each one implies the other. This means that A and B are equivalent iff there is a proof π of B from A and also a proof σ of A from B. Shouldn't these π and σ be somehow related?
- Compare it to the following situation. When two sets are equivalent? The existence of one map in each direction between the two, or the existence of a bijection? Now, what if the sets are the sets of proofs for A and B. If A and B are equivalent, shouldn't these sets be in one-to-one correspondence? In this sense, in addition to the existence of π and σ, we also need them to be the inverse of each other, i.e., $\pi\sigma = id$ and $\sigma\pi = id$. Hence, we need to understand the equality!
- Again, are $A \wedge B$ and $B \wedge A$ equivalent? What about A and $A \wedge A$?

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations.

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions).

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions). Identifying these relative structures, we can claim:

- A realization of a proof is whatever behaves like a proof, i.e., follows the relative structure. It can be anything, a continuous map, a computation, etc.

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions). Identifying these relative structures, we can claim:

- A realization of a proof is whatever behaves like a proof, i.e., follows the relative structure. It can be anything, a continuous map, a computation, etc.
- What is the abstract proof beyond all the realizations? It is a morphism living in the corresponding free category.

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions). Identifying these relative structures, we can claim:

- A realization of a proof is whatever behaves like a proof, i.e., follows the relative structure. It can be anything, a continuous map, a computation, etc.
- What is the abstract proof beyond all the realizations? It is a morphism living in the corresponding free category.
- What is the equality of proofs? It is the equality of maps in the category.

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions). Identifying these relative structures, we can claim:

- A realization of a proof is whatever behaves like a proof, i.e., follows the relative structure. It can be anything, a continuous map, a computation, etc.
- What is the abstract proof beyond all the realizations? It is a morphism living in the corresponding free category.
- What is the equality of proofs? It is the equality of maps in the category.
- What is an equivalence between the formulas? It is the existence of an isomorphism between the objects in the category.

A Way Out?

To answer these questions, we must know what are those abstract objects that usual proofs are the realization of. For that purpose, we need a structural setting in which one can describe the relative behavior of an entity rather than the details of its implementations. This framework of structural relative thinking is called category theory where morphisms (proofs) are more important than the objects (propositions). Identifying these relative structures, we can claim:

- A realization of a proof is whatever behaves like a proof, i.e., follows the relative structure. It can be anything, a continuous map, a computation, etc.
- What is the abstract proof beyond all the realizations? It is a morphism living in the corresponding free category.
- What is the equality of proofs? It is the equality of maps in the category.
- What is an equivalence between the formulas? It is the existence of an isomorphism between the objects in the category. Brouwer's quote!

Definition

A category $\mathcal{C}$ is the following data:

- a colection of objects, denoted by $ob(\mathcal{C})$,
- a collection of morphisms, denoted by $Mor(\mathcal{C})$,
- for any morphism $f \in Mor(\mathcal{C})$, an object $s(f)$ called the source of f,
- for any morphism $f \in Mor(\mathcal{C})$, an object $t(f)$ called the target of f,
- for any object $A \in ob(\mathcal{C})$, a morphism id_A,
- for any two morphisms $f, g \in Mor(\mathcal{C})$ such that $s(f) = t(g)$, a morphism $f \circ g$,

satisfying the following properties:

- $s(id_A) = t(id_A) = A$,
- $s(f \circ g) = s(g)$ and $t(f \circ g) = t(f)$,
- $f \circ id_A = f = id_B \circ f$, if $s(f) = A$ and $t(f) = B$,
- $f \circ (g \circ h) = (f \circ g) \circ h$.

For any $f \in Mor(\mathcal{C})$, we summarize the data $s(f) = A$ and $t(f) = B$ by $f : A \to B$. For any two objects $A, B \in ob(\mathcal{C})$ by $\mathcal{C}(A, B)$ or $Hom_{\mathcal{C}}(A, B)$, we mean the collection of all morphisms $f : A \to B$.

For any $f \in Mor(\mathcal{C})$, we summarize the data $s(f) = A$ and $t(f) = B$ by $f : A \to B$. For any two objects $A, B \in ob(\mathcal{C})$ by $\mathcal{C}(A, B)$ or $Hom_{\mathcal{C}}(A, B)$, we mean the collection of all morphisms $f : A \to B$.

Example

The collection of all sets as the objects and the usual functions as the morphisms with the usual composition and identity constitutes a category. This category is denoted by **Set**.

For any $f \in Mor(\mathcal{C})$, we summarize the data $s(f) = A$ and $t(f) = B$ by $f : A \to B$. For any two objects $A, B \in ob(\mathcal{C})$ by $\mathcal{C}(A, B)$ or $Hom_{\mathcal{C}}(A, B)$, we mean the collection of all morphisms $f : A \to B$.

Example

The collection of all sets as the objects and the usual functions as the morphisms with the usual composition and identity constitutes a category. This category is denoted by **Set**.

Example

A category $\mathcal{C}$ is called a preorder if for any two objects $A, B \in ob(\mathcal{C})$, the collection $Hom_{\mathcal{C}}(A, B)$ has at most one element. Spelling out the definition of a category in this special case, a preorder is actually a set, usually denoted by P with a binary relation $\leq \subseteq P \times P$ such that $x \leq x$, for any $x \in P$ and if $x \leq y$ and $y \leq z$ then $x \leq z$. The prototype example of preorders is a set of subsets of some set X with inclusion as the order.

Categories as Proof Systems

Objects in a category can be interpreted as propositions and morphisms $f : A \to B$ as proofs of B from A. This idea goes back to Joachim Lambek, inspired by Lawvere's thesis. Lambek called studying the proof systems via categorical means the categorical proof theory. This is a generalization of the usual Curry-Howard correspondence.

Categories as Proof Systems

Objects in a category can be interpreted as propositions and morphisms $f : A \to B$ as proofs of B from A. This idea goes back to Joachim Lambek, inspired by Lawvere's thesis. Lambek called studying the proof systems via categorical means the categorical proof theory. This is a generalization of the usual Curry-Howard correspondence.

- For instance, in **Set**, we can think of the set A as the set of all evidences for a proposition A. Then, a proof is a function to transform any evidence of A to an evidence for B.

Categories as Proof Systems

Objects in a category can be interpreted as propositions and morphisms $f : A \to B$ as proofs of B from A. This idea goes back to Joachim Lambek, inspired by Lawvere's thesis. Lambek called studying the proof systems via categorical means the categorical proof theory. This is a generalization of the usual Curry-Howard correspondence.

- For instance, in **Set**, we can think of the set A as the set of all evidences for a proposition A. Then, a proof is a function to transform any evidence of A to an evidence for B.
- In this sense, the preorders are the proof systems in which all proofs between two statements are collapsed to a single proof and hence the only information the category stores is just the provability.

Categories as Proof Systems

Objects in a category can be interpreted as propositions and morphisms $f : A \to B$ as proofs of B from A. This idea goes back to Joachim Lambek, inspired by Lawvere's thesis. Lambek called studying the proof systems via categorical means the categorical proof theory. This is a generalization of the usual Curry-Howard correspondence.

- For instance, in **Set**, we can think of the set A as the set of all evidences for a proposition A. Then, a proof is a function to transform any evidence of A to an evidence for B.
- In this sense, the preorders are the proof systems in which all proofs between two statements are collapsed to a single proof and hence the only information the category stores is just the provability. Therefore, it is no surprise that preorders provide the models of logic where we only care about the provability relation.

Categories as Proof Systems

Objects in a category can be interpreted as propositions and morphisms $f : A \to B$ as proofs of B from A. This idea goes back to Joachim Lambek, inspired by Lawvere's thesis. Lambek called studying the proof systems via categorical means the categorical proof theory. This is a generalization of the usual Curry-Howard correspondence.

- For instance, in **Set**, we can think of the set A as the set of all evidences for a proposition A. Then, a proof is a function to transform any evidence of A to an evidence for B.
- In this sense, the preorders are the proof systems in which all proofs between two statements are collapsed to a single proof and hence the only information the category stores is just the provability. Therefore, it is no surprise that preorders provide the models of logic where we only care about the provability relation.
- Using Brouwer's quote provocatively, mathematics is about categories not preoorders.

Some Examples

Example

The collection of pairs (A, σ_A), where A is a set and σ_A is a bijection over A with equivariant maps is a category. An equivariant map from (A, σ_A) to (B, σ_B) is a function $f : A \to B$ such that $f \circ \sigma_A = \sigma_B \circ f$. This category is denoted by $\mathbf{Set}^{\mathbb{Z}}$.

Some Examples

Example

The collection of pairs (A, σ_A), where A is a set and σ_A is a bijection over A with equivariant maps is a category. An equivariant map from (A, σ_A) to (B, σ_B) is a function $f : A \to B$ such that $f \circ \sigma_A = \sigma_B \circ f$. This category is denoted by $\mathbf{Set}^{\mathbb{Z}}$.

Philosophical Comment

Any object of this category can be interpreted as a *reversible dynamical system* consisting of a set A and a function $f : A \to A$, encoding the dynamism of the system. Of course, any map between the dynamic systems must be a function from the base sets preserving the dynamism. It is also possible to think of the set A as the set of all evidences for a proposition A and σ_A as a transformation up to which we consider the evidences. Then, a proof is a function to transform any evidence of A to an evidence for B, respecting the transformations.

Some Examples

Example

Consider the collection of pairs $(A, \sim_A)$, where A is a subset of $\mathbb{N}$ and $\sim_A$ is an equivalence relation on A together with maps $f : (A, \sim_A) \to (B, \sim_B)$ as the computable functions $f : \mathbb{N} \to \mathbb{N}$ that map A into B and preserve the relation. This is a category. For some reasons, it would be more convenient to identify any two functions $f, g : A \to B$ such that $f(a) \sim_B g(a)$, for any $a \in A$. We will denote this category by **Rec**.

Some Examples

Example

Consider the collection of pairs $(A, \sim_A)$, where A is a subset of $\mathbb{N}$ and $\sim_A$ is an equivalence relation on A together with maps $f : (A, \sim_A) \to (B, \sim_B)$ as the computable functions $f : \mathbb{N} \to \mathbb{N}$ that map A into B and preserve the relation. This is a category. For some reasons, it would be more convenient to identify any two functions $f, g : A \to B$ such that $f(a) \sim_B g(a)$, for any $a \in A$. We will denote this category by **Rec**.

Philosophical Comment

This is a rewriting of the previous example, sticking only to the equivalence relation between the evidences and forgetting the transformations. We also restricted ourselves to all computable functions as the evidence-transformations. After all, it is reasonable to consider the proofs as the algorithmic procedures.

The Category of Proofs

Convention. In this lecture, whenever we refer to natural deduction, we always mean the proofs in **NJ** up to $\beta\eta$-equivalences.

The Category of Proofs

Convention. In this lecture, whenever we refer to natural deduction, we always mean the proofs in **NJ** up to $\beta\eta$-equivalences.

Example

The collection of propositional formulas in the language $\{\top, \bot, \wedge, \vee, \rightarrow\}$ together with natural deduction proofs of B from the assumption A as the maps from A to B is a category. The composition is putting the proofs after one another and the identity id_A is just the axiom that A proves itself. We will denote this category by **Prf**.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

Example

- In **Set**, the terminal object is $\{*\}$.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

Example

- In **Set**, the terminal object is $\{*\}$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the terminal object is the pair $(\{*\}, id_{\{*\}})$.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

Example

- In **Set**, the terminal object is $\{*\}$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the terminal object is the pair $(\{*\}, id_{\{*\}})$.
- In a poset $(P, \leq)$, the terminal object is by definition an element $a \in P$ such that for any $b \in P$, we have $b \leq a$. Hence, the terminal object is the greatest element of the poset.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

Example

- In **Set**, the terminal object is $\{*\}$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the terminal object is the pair $(\{*\}, id_{\{*\}})$.
- In a poset $(P, \leq)$, the terminal object is by definition an element $a \in P$ such that for any $b \in P$, we have $b \leq a$. Hence, the terminal object is the greatest element of the poset.
- In **Rec**, the terminal object is $(\{*\}, =_{\{*\}})$.

The Terminal Object

Definition

An object A is called terminal if for any object B, there exists a unique map from B to A. This map is denoted by $! : B \to A$.

Example

- In **Set**, the terminal object is $\{*\}$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the terminal object is the pair $(\{*\}, id_{\{*\}})$.
- In a poset $(P, \leq)$, the terminal object is by definition an element $a \in P$ such that for any $b \in P$, we have $b \leq a$. Hence, the terminal object is the greatest element of the poset.
- In **Rec**, the terminal object is $(\{*\}, =_{\{*\}})$.
- In **Prf**, the terminal object is $\top$. Notice the effect of the η-equality to ensure the uniqueness.

The Terminal Object as the Trivial Truth

Philosophical Comment

- Reading any category as a proof system, the terminal object is $\top$ which is the trivial truth with no non-trivial proof.

The Terminal Object as the Trivial Truth

Philosophical Comment

- Reading any category as a proof system, the terminal object is $\top$ which is the trivial truth with no non-trivial proof.
- One may argue that although the intuition behind the uniqueness condition is clear for the singletons in **Set**, the use of the η-rule to ensure this condition for the proof is a bit artificial. To convince you, let me emphasize that $\top$ is in the language to formalize the trivial truth, meaning something that is provable with just one proof. $\top$ is not equivalent to any provable statement. You can see the clash with the proof-irrelevant approach here, where all provable statements are considered to be equivalent to $\top$.

The Terminal Object as the Trivial Truth

Philosophical Comment

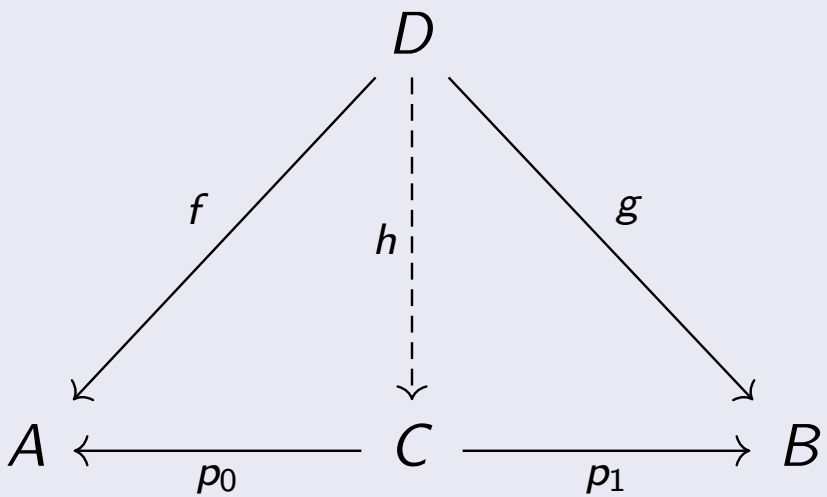
- Reading any category as a proof system, the terminal object is $\top$ which is the trivial truth with no non-trivial proof.
- One may argue that although the intuition behind the uniqueness condition is clear for the singletons in **Set**, the use of the η-rule to ensure this condition for the proof is a bit artificial. To convince you, let me emphasize that $\top$ is in the language to formalize the trivial truth, meaning something that is provable with just one proof. $\top$ is not equivalent to any provable statement. You can see the clash with the proof-irrelevant approach here, where all provable statements are considered to be equivalent to $\top$.

The terminal object is not unique (all singletons in **Set**). However, as it is easy to see that the terminal object is unique up to isomorphism we can denote it by a reserved name 1.

The Product

Definition

Let A and B be two objects. An object C together with two morphisms $p_0 : C \to A$ and $p_1 : C \to B$ is called a product if for any object D and any morphisms $f : D \to A$ and $g : D \to B$, there exists a unique map $h : D \to C$ such that:



This h is denoted by $\langle f, g \rangle$. The product of A and B is denoted by $A \times B$.

Note that $p_0 \circ \langle f, g \rangle = f$ and $p_1 \circ \langle f, g \rangle = g$. Also, $\langle p_0, p_1 \rangle = id_{A \times B}$. A category is called cartesian if it has a terminal object and all binary products.

Note that $p_0 \circ \langle f, g \rangle = f$ and $p_1 \circ \langle f, g \rangle = g$. Also, $\langle p_0, p_1 \rangle = id_{A \times B}$. A category is called cartesian if it has a terminal object and all binary products.

Example

- In **Set**, the product is the usual cartesian product together with the projections.

Note that $p_0 \circ \langle f, g \rangle = f$ and $p_1 \circ \langle f, g \rangle = g$. Also, $\langle p_0, p_1 \rangle = id_{A \times B}$. A category is called cartesian if it has a terminal object and all binary products.

Example

- In **Set**, the product is the usual cartesian product together with the projections.
- In $\mathbf{Set}^{\mathbb{Z}}$, the product of (A, σ_A) and (B, σ_B) is $(A \times B, \sigma_A \times \sigma_B)$ together with the projections, where $[\sigma_A \times \sigma_B](a, b) = (\sigma_A(a), \sigma_B(b))$.

Note that $p_0 \circ \langle f, g \rangle = f$ and $p_1 \circ \langle f, g \rangle = g$. Also, $\langle p_0, p_1 \rangle = id_{A \times B}$. A category is called cartesian if it has a terminal object and all binary products.

Example

- In **Set**, the product is the usual cartesian product together with the projections.
- In $\mathbf{Set}^{\mathbb{Z}}$, the product of (A, σ_A) and (B, σ_B) is $(A \times B, \sigma_A \times \sigma_B)$ together with the projections, where $[\sigma_A \times \sigma_B](a, b) = (\sigma_A(a), \sigma_B(b))$.
- In a poset $(P, \leq)$, the product of $a, b \in P$ is by definition the greatest lower bound of the subset $\{a, b\}$ i.e., an element c such that $c \leq a$ and $c \leq b$ and for any $d \in P$, if $d \leq a$ and $d \leq b$, then $d \leq c$.

Note that $p_0 \circ \langle f, g \rangle = f$ and $p_1 \circ \langle f, g \rangle = g$. Also, $\langle p_0, p_1 \rangle = id_{A \times B}$. A category is called cartesian if it has a terminal object and all binary products.

Example

- In **Set**, the product is the usual cartesian product together with the projections.
- In $\mathbf{Set}^{\mathbb{Z}}$, the product of (A, σ_A) and (B, σ_B) is $(A \times B, \sigma_A \times \sigma_B)$ together with the projections, where $[\sigma_A \times \sigma_B](a, b) = (\sigma_A(a), \sigma_B(b))$.
- In a poset $(P, \leq)$, the product of $a, b \in P$ is by definition the greatest lower bound of the subset $\{a, b\}$ i.e., an element c such that $c \leq a$ and $c \leq b$ and for any $d \in P$, if $d \leq a$ and $d \leq b$, then $d \leq c$.
- In **Rec**, the product of $(A, \sim_A)$ and $(B, \sim_B)$ is $(C, \sim_C)$, together with the maps $p_0 : C \to A$ and $p_1 : C \to B$, where $C = \{2^a(2b+1) \mid a \in A, b \in B\}$, $2^a(2b+1) \sim_C 2^c(2d+1)$ if $a \sim_A c$ and $b \sim_B d$ and $p_0(2^n(2m+1)) = n$ and $p_1(2^n(2m+1)) = m$.

The Product in **Prf**

In **Prf**, the product of A and B is the conjunction $A \wedge B$ together with the two proof trees in the left and for any two proof trees π, π, the proof tree $\langle \pi, \pi' \rangle$ is nothing but the proof tree in the right:

$$\frac{A \wedge B}{A}\,\wedge E_1 \qquad \frac{A \wedge B}{B}\,\wedge E_2 \qquad \frac{\begin{array}{cc}\pi & \pi' \\ A & B\end{array}}{A \wedge B}\,\wedge I$$

To prove $p_0(\langle \pi, \pi' \rangle) = \pi$, note that the left hand side is the left proof tree in the following

$$\frac{\dfrac{\begin{array}{cc}\pi & \pi' \\ A & B\end{array}}{A \wedge B}\,\wedge I}{A}\,\wedge E_1 \qquad \frac{\dfrac{\pi}{A \wedge B}\,\wedge E_1 \qquad \dfrac{\pi}{A \wedge B}\,\wedge E_2}{A \wedge B}\,\wedge I$$

which is β-equivalent to π. For uniqueness, it is enough to prove $\langle p_0\pi, p_1\pi \rangle = \pi$. Here, the left hand side is the right proof tree which is η-equivalent to π.

The Product as the Conjunction

Philosophical Comment

- Reading any category as a proof system, we can interpret the product as the conjunction.

The Product as the Conjunction

Philosophical Comment

- Reading any category as a proof system, we can interpret the product as the conjunction.
- One may naturally object that although the commutativity and the uniqueness for the cartesian product is natural, but the proof theoretical counterpart, i.e., the β- and η-equivalences are not and maybe the product is just too demanding. To motivate that, note that the product of A and B is just an object (together with a natural isomorphism) between $Hom(C, A \times B)$ and $Hom(C, A) \times Hom(C, B)$. This just says that any proof of $A \wedge B$ from C is in one-to-one correspondence (i.e., uniform in A, B and C) with the pairs of proofs of A and B from C. Think about the BHK interpretation.

The Product as the Conjunction

Philosophical Comment

- Reading any category as a proof system, we can interpret the product as the conjunction.
- One may naturally object that although the commutativity and the uniqueness for the cartesian product is natural, but the proof theoretical counterpart, i.e., the β- and η-equivalences are not and maybe the product is just too demanding. To motivate that, note that the product of A and B is just an object (together with a natural isomorphism) between $Hom(C, A \times B)$ and $Hom(C, A) \times Hom(C, B)$. This just says that any proof of $A \wedge B$ from C is in one-to-one correspondence (i.e., uniform in A, B and C) with the pairs of proofs of A and B from C. Think about the BHK interpretation.
- Therefore, the projections, the commutation and the uniqueness are not essential. They are just a presentation of a deep universal fact that characterizes the conjunction.

Some Isomorphisms

Using the universal property of the terminal and the products, it is easy to prove the following list of isomorphisms:

- $A \times 1 \cong A$ via the maps $p_0 : A \times 1 \to A$ and $\langle id_A, ! \rangle : A \to A \times 1$.

Some Isomorphisms

Using the universal property of the terminal and the products, it is easy to prove the following list of isomorphisms:

- $A \times 1 \cong A$ via the maps $p_0 : A \times 1 \to A$ and $\langle id_A, ! \rangle : A \to A \times 1$.
- $A \times B \cong B \times A$ via the maps $\langle p_1, p_0 \rangle : A \times B \to B \times A$.

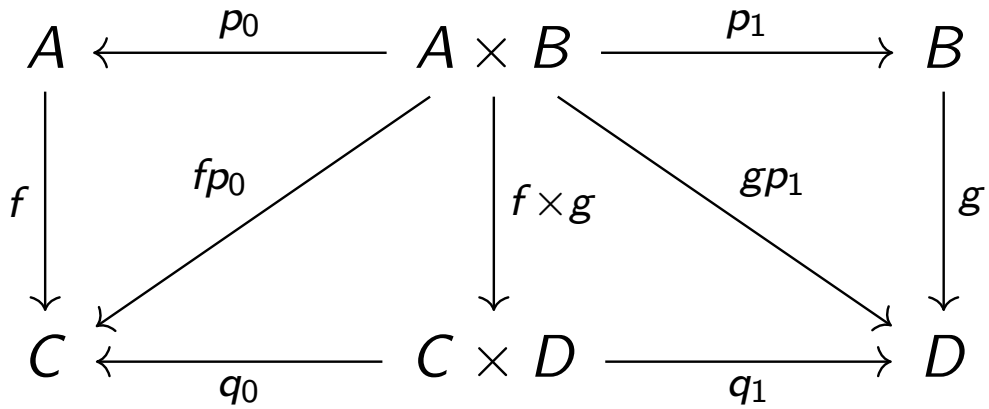
Some Isomorphisms

Using the universal property of the terminal and the products, it is easy to prove the following list of isomorphisms:

- $A \times 1 \cong A$ via the maps $p_0 : A \times 1 \to A$ and $\langle id_A, ! \rangle : A \to A \times 1$.
- $A \times B \cong B \times A$ via the maps $\langle p_1, p_0 \rangle : A \times B \to B \times A$.
- $(A \times B) \times C \cong A \times (B \times C)$.

Functoriality

The product as an operator not only acts on the objects, but also acts on the morphisms. Assume that both the product of A and B and the product of C and D exist. Now, assume that we are given $f : A \to C$ and $g : B \to D$. We intend to come up with a canonical map $f \times g : A \times B \to C \times D$. For that purpose, first note that $p_0 : A \times B \to A$ and $p_1 : A \times B \to B$. Therefore, $fp_0 : A \times B \to C$ and $gp_1 : A \times B \to D$. Therefore, $\langle fp_0, gp_1 \rangle : A \times B \to C \times D$:



The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

Example

- In **Set** the initial object is $\varnothing$.

The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

Example

- In **Set** the initial object is $\varnothing$.
- In $\mathbf{Set}^{\mathbb{Z}}$ the initial object is the pair $(\varnothing, id_{\varnothing})$.

The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

Example

- In **Set** the initial object is $\varnothing$.
- In $\mathbf{Set}^{\mathbb{Z}}$ the initial object is the pair $(\varnothing, id_{\varnothing})$.
- In a poset $(P, \leq)$, the initial object is by definition an element $a \in P$ such that for any $b \in P$, we have $a \leq b$. Hence, the initial object is the least element of the poset.

The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

Example

- In **Set** the initial object is $\varnothing$.
- In $\mathbf{Set}^{\mathbb{Z}}$ the initial object is the pair $(\varnothing, id_{\varnothing})$.
- In a poset $(P, \leq)$, the initial object is by definition an element $a \in P$ such that for any $b \in P$, we have $a \leq b$. Hence, the initial object is the least element of the poset.
- In **Rec**, the initial object is $(\varnothing, =_{\varnothing})$.

The Initial Object

Definition

An object A is called initial if for any object B, there exists a unique map from A to B. We denote the initial object by 0 and the unique map by $! : A \to B$.

Example

- In **Set** the initial object is $\varnothing$.
- In $\mathbf{Set}^{\mathbb{Z}}$ the initial object is the pair $(\varnothing, id_{\varnothing})$.
- In a poset $(P, \leq)$, the initial object is by definition an element $a \in P$ such that for any $b \in P$, we have $a \leq b$. Hence, the initial object is the least element of the poset.
- In **Rec**, the initial object is $(\varnothing, =_{\varnothing})$.
- In **Prf**, the initial object is $\bot$. Notice the effect of the η-equality to ensure the uniqueness.

The Initial Object as Inconsistency

Philosophical Comment

- Reading any category as a proof system, we can interpret the initial as the inconsistency.

The Initial Object as Inconsistency

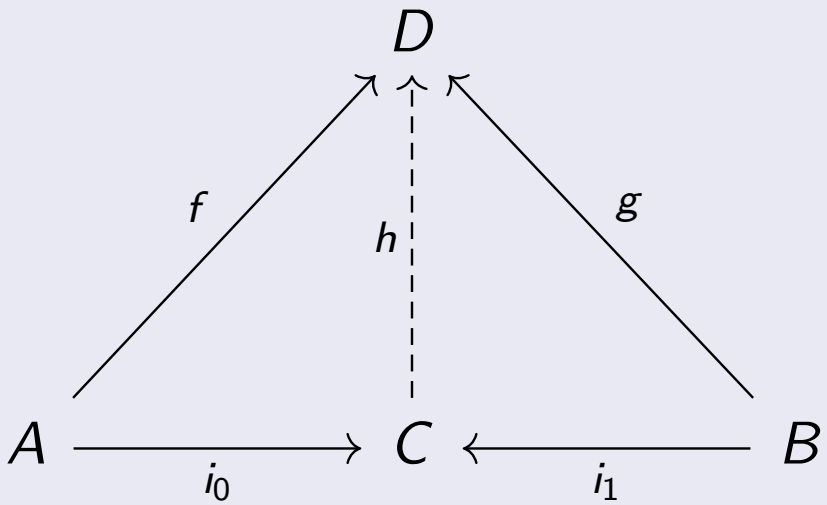
Philosophical Comment

- Reading any category as a proof system, we can interpret the initial as the inconsistency.
- The existence of a map from $\bot$ to A is a consequence of the BHK definition of $\bot$ that says $\bot$ has no proof. But it is weaker, right?

The Coproduct

Definition

Let A and B be two objects. An object C together with two morphisms $i_0 : A \to C$ and $i_1 : B \to C$ is called a coproduct if for any object D and any morphisms $f : A \to D$ and $g : B \to D$, there exists a unique map $h : C \to D$ such that:



This h is denoted by $[f, g]$. The coproduct of A and B is denoted by $A + B$.

Some Examples

A category is called cocartesian if it has the initial object and all binary coproducts.

Some Examples

A category is called cocartesian if it has the initial object and all binary coproducts.

Example

- In **Set**, the coproduct of A and B is the disjoint union $A + B$ together with the injections $i_0 : A \to A + B$ and $i_1 : B \to A + B$ defined by $i_0(a) = (0, a)$ and $i_1(b) = (1, b)$.

Some Examples

A category is called cocartesian if it has the initial object and all binary coproducts.

Example

- In **Set**, the coproduct of A and B is the disjoint union $A + B$ together with the injections $i_0 : A \to A + B$ and $i_1 : B \to A + B$ defined by $i_0(a) = (0, a)$ and $i_1(b) = (1, b)$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the coproduct of (A, σ_A) and (B, σ_B) is $(A + B, \sigma_A + \sigma_B)$ together with the injections, where $[\sigma_A + \sigma_B](0, a) = (0, \sigma_A(a))$ and $[\sigma_A + \sigma_B](1, b) = (1, \sigma_B(b))$.

Some Examples

A category is called cocartesian if it has the initial object and all binary coproducts.

Example

- In **Set**, the coproduct of A and B is the disjoint union $A + B$ together with the injections $i_0 : A \to A + B$ and $i_1 : B \to A + B$ defined by $i_0(a) = (0, a)$ and $i_1(b) = (1, b)$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the coproduct of (A, σ_A) and (B, σ_B) is $(A + B, \sigma_A + \sigma_B)$ together with the injections, where $[\sigma_A + \sigma_B](0, a) = (0, \sigma_A(a))$ and $[\sigma_A + \sigma_B](1, b) = (1, \sigma_B(b))$.
- In a poset $(P, \leq)$, the coproduct of $a, b \in P$ is by definition the least upper bound of the subset $\{a, b\}$ i.e., an element c such that $a \leq c$ and $b \leq c$ and for any $d \in P$, if $a \leq d$ and $b \leq d$, then $c \leq d$.

Some Examples

Example

- In **Rec**, the coproduct of $(A, \sim_A)$ and $(B, \sim_B)$ is $(C, \sim_C)$, together with the maps $i_0 : A \to C$ and $i_1 : B \to C$, where $C = \{2^a(2b+1) \mid (a = 0 \text{ and } b \in A) \text{ or } (a = 1 \text{ and } b \in B)\}$, $2^a(2b+1) \sim_C 2^c(2d+1)$ if $(a = c = 0 \text{ and } b \sim_A d)$ or $(a = c = 1 \text{ and } b \sim_B d)$ and $i_0(n) = 2n+1$ and $i_1(n) = 2(2n+1)$.

Some Examples

Example

- In **Rec**, the coproduct of $(A, \sim_A)$ and $(B, \sim_B)$ is $(C, \sim_C)$, together with the maps $i_0 : A \to C$ and $i_1 : B \to C$, where $C = \{2^a(2b+1) \mid (a = 0 \text{ and } b \in A) \text{ or } (a = 1 \text{ and } b \in B)\}$, $2^a(2b+1) \sim_C 2^c(2d+1)$ if $(a = c = 0 \text{ and } b \sim_A d)$ or $(a = c = 1$ and $b \sim_B d)$ and $i_0(n) = 2n+1$ and $i_1(n) = 2(2n+1)$.
- In **Prf**, the coproduct of A and B is the disjunction $A \vee B$ together with the two proof trees in the left and $[\pi, \sigma]$ is defined as the proof tree in the left:

$$\frac{A}{A \vee B} \vee I_1 \qquad \frac{B}{A \vee B} \vee I_2 \qquad \frac{A \vee B \qquad \begin{matrix}[A] \\ \pi \\ C\end{matrix} \qquad \begin{matrix}[B] \\ \sigma \\ C\end{matrix}}{C} \vee E$$

The Coproduct as Disjunction

Philosophical Comment

- Reading any category as a proof system, we can interpret the coproduct as the disjunction.

The Coproduct as Disjunction

Philosophical Comment

- Reading any category as a proof system, we can interpret the coproduct as the disjunction.
- To motivate, note that the coproduct of A and B is just an object (together with a natural isomorphism) between $Hom(A + B, C)$ and $Hom(A, C) \times Hom(B, C)$. This just says that any proof of C from $A \vee B$ is in one-to-one correspondence (i.e., uniform in A, B and C) with the pairs of proofs of C from A and B. If we think about the BHK interpretation, it is not exactly that. But it is its consequence.

The Coproduct as Disjunction

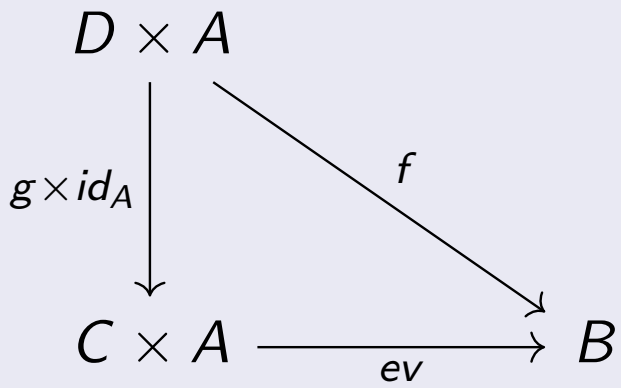
Philosophical Comment

- Reading any category as a proof system, we can interpret the coproduct as the disjunction.
- To motivate, note that the coproduct of A and B is just an object (together with a natural isomorphism) between $Hom(A + B, C)$ and $Hom(A, C) \times Hom(B, C)$. This just says that any proof of C from $A \vee B$ is in one-to-one correspondence (i.e., uniform in A, B and C) with the pairs of proofs of C from A and B. If we think about the BHK interpretation, it is not exactly that. But it is its consequence.
- Therefore, the injections, the commutation and the uniqueness are not essential. They are just a presentation of a deep universal fact that characterizes the disjunction.

Exponential Object

Definition

Let $\mathcal{C}$ be a category with products and A and B be two objects. An object C together with a morphism $ev : C \times A \to B$ is called an exponential object if for any $f : D \times A \to B$, there exists a unique $g : D \to C$ such that:



The exponentiation is denoted by B^A. A cartesian category with all exponentials is called cartesian closed. A cartesian closed category which is cocartesian is called bicartesian closed.

Some Examples

Example

- In the category **Set**, the exponential is $B^A = \{f | f : A \to B\}$ with the morphism $ev : B^A \times A \to B$ defined by $ev(f, a) = f(a)$. Note that for any map $f : C \times A \to B$, the map $g : C \to B^A$ is defined by $g(c)(a) = f(c, a)$.

Some Examples

Example

- In the category **Set**, the exponential is $B^A = \{f | f : A \to B\}$ with the morphism $ev : B^A \times A \to B$ defined by $ev(f, a) = f(a)$. Note that for any map $f : C \times A \to B$, the map $g : C \to B^A$ is defined by $g(c)(a) = f(c, a)$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the exponential of (B, σ_B) by (A, σ_A) is (B^A, σ_{B^A}) together with the evaluation map, where B^A is the set of *all* functions and $\sigma_{B^A}(f) = \sigma_B f \sigma_A^{-1}$.

Some Examples

Example

- In the category **Set**, the exponential is $B^A = \{f | f : A \to B\}$ with the morphism $ev : B^A \times A \to B$ defined by $ev(f, a) = f(a)$. Note that for any map $f : C \times A \to B$, the map $g : C \to B^A$ is defined by $g(c)(a) = f(c, a)$.
- In $\mathbf{Set}^{\mathbb{Z}}$, the exponential of (B, σ_B) by (A, σ_A) is (B^A, σ_{B^A}) together with the evaluation map, where B^A is the set of *all* functions and $\sigma_{B^A}(f) = \sigma_B f \sigma_A^{-1}$.
- In a poset $(P, \leq)$, the exponentiation is by definition the least element c such that $c \wedge a \leq b$ i.e., an element c such that $c \wedge a \leq b$ and for any $d \in P$ if $d \wedge a \leq b$ then $d \leq c$. Exponential objects in posets are called Heyting implications and denoted by $\to$.

Example

- A prototype example of preorder bicartesian closed categories is the frames. We can say that the finitely verifiable propositions provide a natural setting for provability. Is it possible to use finite verifiability to also address proofs and not just provability?

Example

- A prototype example of preorder bicartesian closed categories is the frames. We can say that the finitely verifiable propositions provide a natural setting for provability. Is it possible to use finite verifiability to also address proofs and not just provability? Yes! These categories are called Grothendieck toposes.

Example

- A prototype example of preorder bicartesian closed categories is the frames. We can say that the finitely verifiable propositions provide a natural setting for provability. Is it possible to use finite verifiability to also address proofs and not just provability? Yes! These categories are called Grothendieck toposes.
- In **Rec**, the exponentiation of $(B, \sim_B)$ by $(A, \sim_A)$ is $(C, \sim_C)$, together with the map $ev : C \times A \to B$, where C is the set of all numbers $e \in \mathbb{N}$ such that $\forall n \in A\ e \cdot n \in B$ and for all $m, n \in A$, if $m \sim_A n$ then $e \cdot m \sim_B e \cdot n\}$. Also, define $e \sim_C f$ if $e \cdot n \sim_B f \cdot n$, for any $n \in A$ and $ev(e, a) = e \cdot a$.

Example

- A prototype example of preorder bicartesian closed categories is the frames. We can say that the finitely verifiable propositions provide a natural setting for provability. Is it possible to use finite verifiability to also address proofs and not just provability? Yes! These categories are called Grothendieck toposes.
- In **Rec**, the exponentiation of $(B, \sim_B)$ by $(A, \sim_A)$ is $(C, \sim_C)$, together with the map $ev : C \times A \to B$, where C is the set of all numbers $e \in \mathbb{N}$ such that $\forall n \in A\ e \cdot n \in B$ and for all $m, n \in A$, if $m \sim_A n$ then $e \cdot m \sim_B e \cdot n\}$. Also, define $e \sim_C f$ if $e \cdot n \sim_B f \cdot n$, for any $n \in A$ and $ev(e, a) = e \cdot a$. Note that ev is actually the universal machine and for finding the map that the definition of the exponentiation demands, we need the S_{mn}-theorem. The idea is simply that we need a computable function that maps an algorithm $f : C \times A \to B$ to an algorithm $g : C \to B^A$ such that $(g \cdot c) \cdot a = f \cdot (2^c(2a + 1))$.

Some Examples

Example

In **Prf**, the exponential of B by A is the implication $A \to B$ together with the left proof tree as the evaluation map. The right proof tree also helps to introduce the map the definition demands for π:

$$\frac{A \qquad A \to B}{B} \to E \qquad \frac{\begin{array}{c}[A] \\ \pi \\ B\end{array}}{A \to B} \to I$$

The Exponential as the Implication

Philosophical Comment

- Reading any category as a proof system, we can interpret the exponential as the implication.

The Exponential as the Implication

Philosophical Comment

- Reading any category as a proof system, we can interpret the exponential as the implication.
- To motivate, note that the exponential B^A is just an object (together with a natural isomorphism) between $Hom(C \times A, B)$ and $Hom(C, A \to B)$. This just says that any proof of B from $C \times A$ is in one-to-one correspondence (i.e., uniform in A, B and C) with the proofs of $A \to B$ from C.

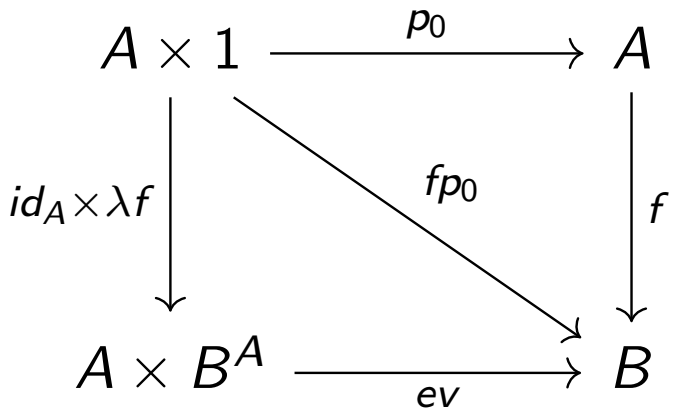
The Exponential as the Implication

Philosophical Comment

- Reading any category as a proof system, we can interpret the exponential as the implication.
- To motivate, note that the exponential B^A is just an object (together with a natural isomorphism) between $Hom(C \times A, B)$ and $Hom(C, A \to B)$. This just says that any proof of B from $C \times A$ is in one-to-one correspondence (i.e., uniform in A, B and C) with the proofs of $A \to B$ from C.
- Therefore, the evaluation, the commutation and the uniqueness are not essential. They are just a presentation of a deep universal fact that characterizes the implication.

A Comment on Internalization

There is a one-to-one correspondence between the maps $A \to B$ and the maps $1 \to B^A$. More precisely, for any $f : A \to B$, using $p_0 : A \times 1 \to A$, we have the map $fp_0 : A \times 1 \to B$. By the definition of exponentials, there is a unique map from 1 to B^A, denoted by λf, such that:



A Comment on Internalization

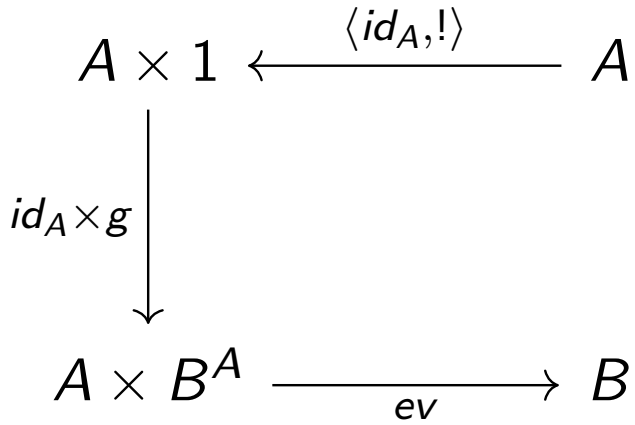
Conversely, having a map $g : 1 \to B^A$, we have

$$\begin{array}{ccc} A \times 1 & \xleftarrow{\langle id_A, ! \rangle} & A \\ {\scriptstyle id_A \times g}\downarrow & & \\ A \times B^A & \xrightarrow[ev]{} & B \end{array}$$

as a map from A to B. For any map $a : X \to A$, we denote the composition of the previous map by a, by $g \cdot a : X \to B$.

A Comment on Internalization

Conversely, having a map $g : 1 \to B^A$, we have

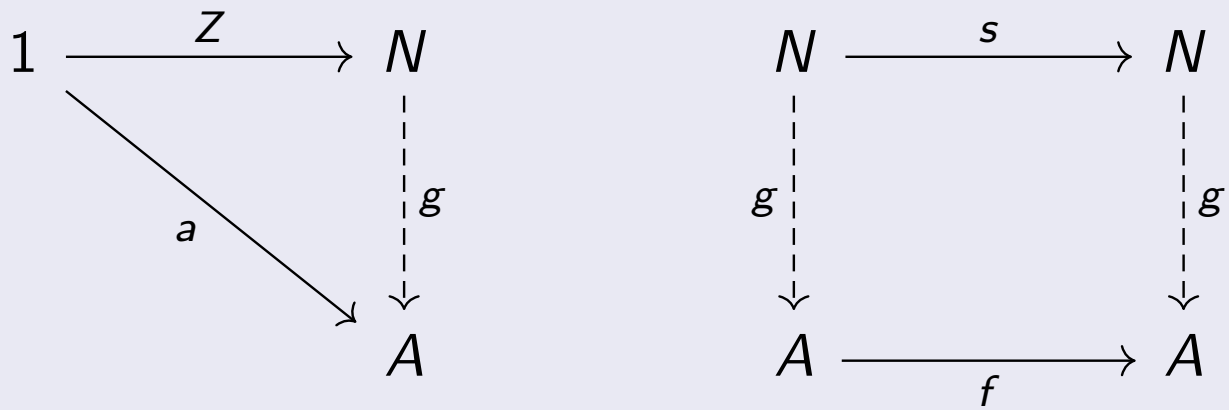


as a map from A to B. For any map $a : X \to A$, we denote the composition of the previous map by a, by $g \cdot a : X \to B$. Note that $(\lambda f) \cdot a = f \circ a$.

The Natural Number Object

Definition

A natural number object in a category with a terminal object is an object N together with maps $Z : 1 \to N$ and $s : N \to N$ such that for any object A and any maps $a : 1 \to A$ and $f : A \to A$, there exists a unique map $g : N \to A$ such that



Some Examples

Example

- In **Set**, the set of natural numbers together with the morphism $Z : \{*\} \to \mathbb{N}$ mapping $*$ to 0 and $s : \mathbb{N} \to \mathbb{N}$ mapping any number to its successor is a natural number object.

Some Examples

Example

- In **Set**, the set of natural numbers together with the morphism $Z : \{*\} \to \mathbb{N}$ mapping $*$ to 0 and $s : \mathbb{N} \to \mathbb{N}$ mapping any number to its successor is a natural number object.
- In $\mathbf{Set}^{\mathbb{Z}}$, the pair $(\mathbb{N}, id_{\mathbb{N}})$ together with the same data as before is a natural number object.

Some Examples

Example

- In **Set**, the set of natural numbers together with the morphism $Z : \{*\} \to \mathbb{N}$ mapping $*$ to 0 and $s : \mathbb{N} \to \mathbb{N}$ mapping any number to its successor is a natural number object.
- In $\mathbf{Set}^{\mathbb{Z}}$, the pair $(\mathbb{N}, id_{\mathbb{N}})$ together with the same data as before is a natural number object.
- In **Rec**, the pair $(\mathbb{N}, =_{\mathbb{N}})$ together with the usual morphisms is a natural number object.

Some Examples

Example

- In **Set**, the set of natural numbers together with the morphism $Z : \{*\} \to \mathbb{N}$ mapping $*$ to 0 and $s : \mathbb{N} \to \mathbb{N}$ mapping any number to its successor is a natural number object.
- In $\mathbf{Set}^{\mathbb{Z}}$, the pair $(\mathbb{N}, id_{\mathbb{N}})$ together with the same data as before is a natural number object.
- In **Rec**, the pair $(\mathbb{N}, =_{\mathbb{N}})$ together with the usual morphisms is a natural number object.
- In **Prf**, there is no natural number object and one may think that this breaks the connection between proofs and categories. However, if we extend the notion of proof as a constructive construction, then a natural number object can also be a type of constructions.

The Free Category

Recall that we aimed to identify the basic structure of proofs and now, we may say that it is the bicartesian closed structure. Therefore, any of the categories that we have seen can be interpreted as a proof system realizing intuitionistic proofs or if you like they can be the *models* of intuitionistic proofs. But what about the free category of abstract proofs? We do not want to define free bicartesian closed categories here. Let us just say that the free bicartesian closed category constructed from the object in a set X is the "smallest" bicartesian closed category containing X in its class of objects. Smallest here means that it has all the required structures (e.g. the terminal object, the product of the objects, the projections, the pairing of morphisms, etc) satisfying all required equations (e.g. $p_0(\langle f, g\rangle) = f$) but not anything more.

The Free Category

- For instance, the category **Prf** is the free bicartesian closed category constructed from the infinite set of objects $\{p_0, p_1, \cdots\}$. Therefore, the category of natural deductions is actually *the* category of intuitionistic propositional proofs with the atoms $\{p_0, p_1, \cdots\}$.

The Free Category

- For instance, the category **Prf** is the free bicartesian closed category constructed from the infinite set of objects $\{p_0, p_1, \cdots\}$. Therefore, the category of natural deductions is actually *the* category of intuitionistic propositional proofs with the atoms $\{p_0, p_1, \cdots\}$.
- In a similar way, it is possible to come up with the "smallest" cartesian (closed) category with a natural number object denoted by **T**. It has objects such as N, $N \times N$, $N^N \times N$. Every object of the category is constructed from N and 1, by product and exponentiation. The morphisms are all constructed from the very basic morphisms such as p_0 and p_1, by composition, pairing, etc.

Representable Functions

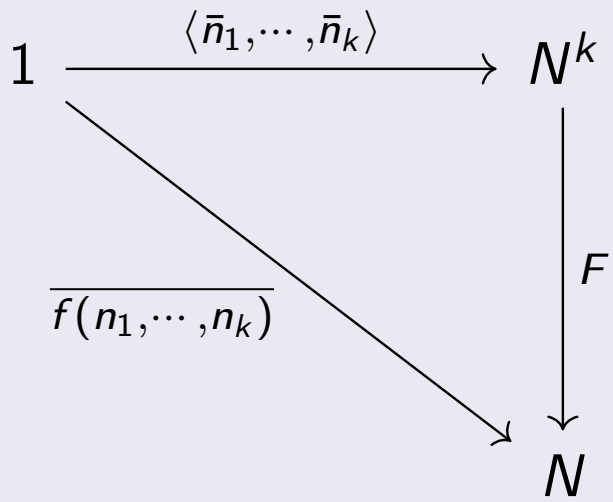
In any cartesian category with a natural number object, there is a canonical way to represent the natural number n by $\bar{n} = s^n \circ Z : 1 \to N$. Similarly, it is possible to represent numeral functions.

Representable Functions

In any cartesian category with a natural number object, there is a canonical way to represent the natural number n by $\bar{n} = s^n \circ Z : 1 \to N$. Similarly, it is possible to represent numeral functions.

Definition

A function $f : \mathbb{N}^k \to \mathbb{N}$ is called representable in the category $\mathcal{C}$ with a natural number object if there is a map $F : N^k \to N$ such that for any $(n_1, \cdots, n_k) \in \mathbb{N}^k$ we have



Some Examples

Example

- In **Set**, all numeral functions are representable.

Some Examples

Example

- In **Set**, all numeral functions are representable.
- In **Rec**, the representable functions are the total computable functions.

Some Examples

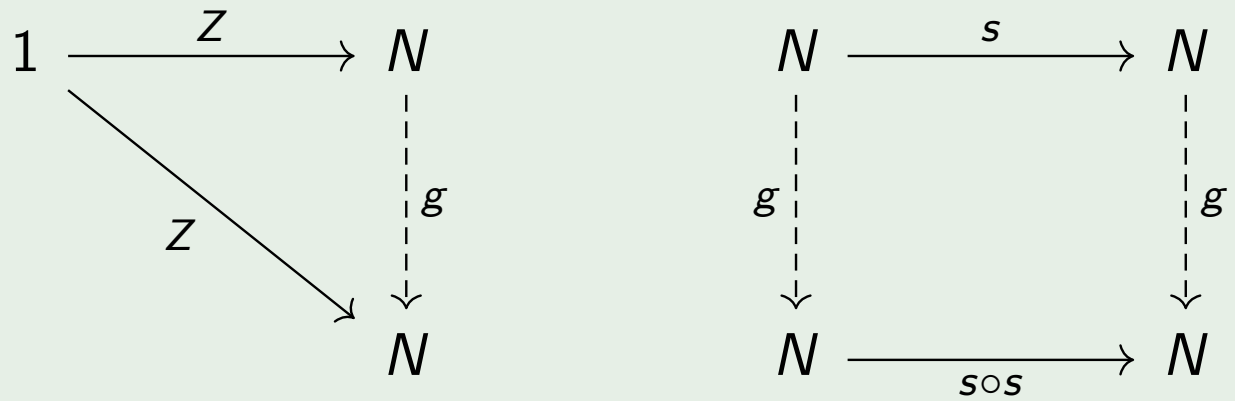
Example

- In **Set**, all numeral functions are representable.
- In **Rec**, the representable functions are the total computable functions.
- In the free cartesian closed category with a natural number object, the representable functions are called primitive recursive functionals. Of course any primitive recursive function is representable in **T**.

Some Examples

Example

- In **Set**, all numeral functions are representable.
- In **Rec**, the representable functions are the total computable functions.
- In the free cartesian closed category with a natural number object, the representable functions are called primitive recursive functionals. Of course any primitive recursive function is representable in **T**. For instance, we can represent the function $n \mapsto 2n$ by the map $g : N \to N$:

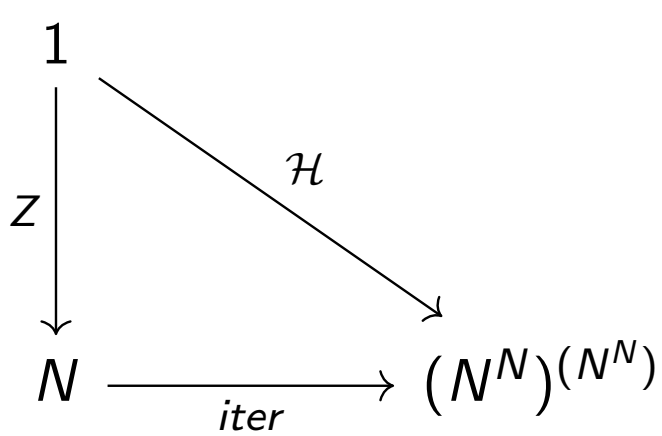


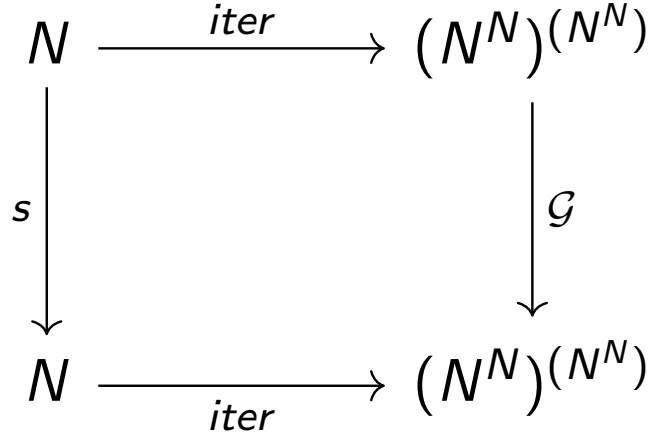
Some Examples

Example

- The power of **T** is far from the primitive recursion. For instance, it can represent the function $h(m, n)$ defined by $h(0, n) = n + 1$ and $h(m + 1, n) = h_m^n(n)$, where h_m is just $h(m, -)$. The idea simply is to use higher order recursion on m to define h_m by $h_0 = s$ and $h_{m+1} = \lambda n.h_m^n(n)$. The function h has the growth rate equivalent to that of the Ackermann function and hence is not primitive recursive.

Consider the map $\mathcal{F} : (N^N)^{(N^N)} \times N^N \times N \to N$ with the interpretation $ev(ev(F, f), n) = F(f)(n)$ in $\mathbf{T}$. Then, by exponentiality, we have a map $\mathcal{G} : (N^N)^{(N^N)} \to (N^N)^{(N^N)}$. Also, as we have a map $p_1 : N^N \times N \to N$, it gives rise to a map $\mathcal{H} : 1 \to (N^N)^{(N^N)}$. Now, by the definition of natural number object, we have a map $iter : N \to (N^N)^{(N^N)}$:





Now, set $h(n) = iter(n, \lambda s)$.

Thank you for your attention!