Lecture 2: The Curry-Howard isomorphism

Łukasz Czajka

Predicate logic: syntax

Inductive definition of predicate logic formulas (for a fixed implicit signature without function symbols).

Predicate logic: syntax

Inductive definition of predicate logic formulas (for a fixed implicit signature without function symbols).

- If $x_1, \ldots, x_n$ are individual variables and R is an n-ary predicate (relation) symbol, then the atom $R(x_1, \ldots, x_n)$ is a formula.

Predicate logic: syntax

Inductive definition of predicate logic formulas (for a fixed implicit signature without function symbols).

- If $x_1, \ldots, x_n$ are individual variables and R is an n-ary predicate (relation) symbol, then the atom $R(x_1, \ldots, x_n)$ is a formula.
- If φ and ψ are formulas, then $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$ and $\neg\varphi$ are formulas.

Predicate logic: syntax

Inductive definition of predicate logic formulas (for a fixed implicit signature without function symbols).

- If $x_1, \dots, x_n$ are individual variables and R is an n-ary predicate (relation) symbol, then the atom $R(x_1, \dots, x_n)$ is a formula.
- If φ and ψ are formulas, then $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ and $\neg\varphi$ are formulas.
- If φ is a formula and x is an individual variable, then $\forall x \varphi$ and $\exists x \varphi$ are formulas.

Predicate logic: syntax

Inductive definition of predicate logic formulas (for a fixed implicit signature without function symbols).

- If $x_1, \ldots, x_n$ are individual variables and R is an n-ary predicate (relation) symbol, then the atom $R(x_1, \ldots, x_n)$ is a formula.
- If φ and ψ are formulas, then $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$ and $\neg\varphi$ are formulas.
- If φ is a formula and x is an individual variable, then $\forall x\varphi$ and $\exists x\varphi$ are formulas.

NOTE: Propositional variables may be represented by nullary (0-ary) predicates.

Predicate logic: syntax

$$P \to Q \to R$$

Predicate logic: syntax

$$P \to (Q \to R)$$

Predicate logic: syntax

$$P \vee Q \rightarrow R$$

Predicate logic: syntax

$$(P \vee Q) \to R$$

Predicate logic: syntax

$$\forall x R(x) \vee \forall x \neg S(x)$$

Predicate logic: syntax

$$(\forall x R(x)) \vee (\forall x \neg S(x))$$

Predicate logic: the "dot" notation

$$\forall x.R(x) \vee S(x)$$

Predicate logic: the “dot” notation

$$\forall x(R(x) \vee S(x))$$

Predicate logic: free variables

$$Q(x) \vee \forall x \exists y R(x, y, z)$$

Predicate logic: free variables

$$Q(\mathbf{x}) \vee \forall x \exists y R(x, y, \mathbf{z})$$

Predicate logic: variable scopes

$$\forall x.\exists x R(x, x) \vee Q(x)$$

Predicate logic: variable scopes

$$\forall x.(\exists y R(y, y)) \lor Q(x)$$

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \dots, x_n)$ if $(\beta(x_1), \dots, \beta(x_n)) \in R^{\mathcal{A}}$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \dots, x_n)$ if $(\beta(x_1), \dots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.
- $I \models \varphi_1 \to \varphi_2$ iff $I \not\models \varphi_1$ or $I \models \varphi_2$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.
- $I \models \varphi_1 \to \varphi_2$ iff $I \not\models \varphi_1$ or $I \models \varphi_2$.
- $I \models \forall x \varphi$ iff for all $a \in A$ we have $I[a/x] \models \varphi$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.
- $I \models \varphi_1 \to \varphi_2$ iff $I \not\models \varphi_1$ or $I \models \varphi_2$.
- $I \models \forall x \varphi$ iff for all $a \in A$ we have $I[a/x] \models \varphi$.
- $I \models \exists x \varphi$ iff there exists $a \in A$ such that $I[a/x] \models \varphi$.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.
- $I \models \varphi_1 \to \varphi_2$ iff $I \not\models \varphi_1$ or $I \models \varphi_2$.
- $I \models \forall x\varphi$ iff for all $a \in A$ we have $I[a/x] \models \varphi$.
- $I \models \exists x\varphi$ iff there exists $a \in A$ such that $I[a/x] \models \varphi$.

A formula is (classically) valid (is a classical tautology) if it is satisfied by every interpretation.

Predicate logic: classical semantics

An interpretation $I = (\mathcal{A}, \beta)$ consists of a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ (over our fixed signature) and a valuation $\beta : \mathrm{Var} \to A$ of variables.

We define inductively when a formula φ is satisfied by an interpretation I, notation $I \models \varphi$.

- $I \models R(x_1, \ldots, x_n)$ if $(\beta(x_1), \ldots, \beta(x_n)) \in R^{\mathcal{A}}$.
- $I \models \neg\varphi$ iff $I \not\models \varphi$.
- $I \models \varphi_1 \wedge \varphi_2$ iff $I \models \varphi_1$ and $I \models \varphi_2$.
- $I \models \varphi_1 \vee \varphi_2$ iff $I \models \varphi_1$ or $I \models \varphi_2$.
- $I \models \varphi_1 \to \varphi_2$ iff $I \not\models \varphi_1$ or $I \models \varphi_2$.
- $I \models \forall x \varphi$ iff for all $a \in A$ we have $I[a/x] \models \varphi$.
- $I \models \exists x \varphi$ iff there exists $a \in A$ such that $I[a/x] \models \varphi$.

A formula is (classically) valid (is a classical tautology) if it is satisfied by every interpretation.

NOTE: For propositional formulas (without quantifiers and with only nullary predicates) the above semantics is the same as propositional truth-table semantics.

Intuitionistic logic: motivation

- In classical logic, the law of excluded middle

$$\varphi \vee \neg\varphi$$

is valid.

Intuitionistic logic: motivation

- In classical logic, the law of excluded middle

$$\varphi \vee \neg\varphi$$

  is valid.
- But in general we don't know which of $\varphi, \neg\varphi$ holds!

Intuitionistic logic: motivation

- In classical logic, the law of excluded middle

$$\varphi \vee \neg\varphi$$

  is valid.
- But in general we don't know which of $\varphi, \neg\varphi$ holds!
- Similarly, we may be able to classicaly prove $\exists x \varphi$ but still not be able to provide any concrete element for which φ holds.

Intuitionistic logic: motivation

Theorem

There exist irrational numbers a, b such that a^b is rational.

Intuitionistic logic: motivation

Theorem

There exist irrational numbers a, b such that a^b is rational.

Proof.

The number $\sqrt{2}^{\sqrt{2}}$ is either rational or irrational. If $\sqrt{2}^{\sqrt{2}}$ is rational then take $a = b = \sqrt{2}$. If $\sqrt{2}^{\sqrt{2}}$ is irrational then $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2$ is rational, so take $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$. $\square$

Intuitionistic logic

- Classical logic is about truth.

Intuitionistic logic

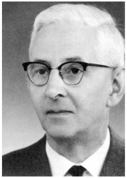
- Classical logic is about truth.
- Intuitionistic (constructive) logic is about constructibility.

The three constructivists: Brouwer, Heyting, Kolmogorov

"It does not make sense to think of truth or falsity of a mathematical statement independently of our knowledge concerning the statement. A statement is true if we have a proof of it, and false if we can show that the assumption that there is a proof for the statement leads to a contradiction." L.E.J. Brouwer

The three constructivists: Brouwer, Heyting, Kolmogorov

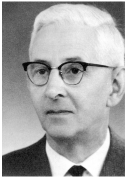
"It does not make sense to think of truth or falsity of a mathematical statement independently of our knowledge concerning the statement. A statement is true if we have a proof of it, and false if we can show that the assumption that there is a proof for the statement leads to a contradiction." L.E.J. Brouwer



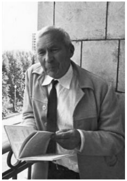
"The idea of an existence of the mathematical entities outside our mind should not enter into the demonstrations." A. Heyting

The three constructivists: Brouwer, Heyting, Kolmogorov

"It does not make sense to think of truth or falsity of a mathematical statement independently of our knowledge concerning the statement. A statement is true if we have a proof of it, and false if we can show that the assumption that there is a proof for the statement leads to a contradiction." L.E.J. Brouwer



"The idea of an existence of the mathematical entities outside our mind should not enter into the demonstrations." A. Heyting



"In addition to theoretical logic, which systematises proof schemata for theoretical truths, one can systematise proof schemata for solutions to problems. (...) Intuitionistic logic should rather be called the calculus of problems, since its objects are in reality problems, rather than theoretical propositions." A. Kolmogorov

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \land \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \land \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.
- A proof of $\varphi_1 \lor \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \wedge \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.
- A proof of $\varphi_1 \vee \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.
- There is no proof of $\perp$ (falsity).

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \wedge \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.
- A proof of $\varphi_1 \vee \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.
- There is no proof of $\bot$ (falsity).
- A proof of $\forall x \varphi$ is a method (constructive function) which transforms any object a into a proof of $\varphi(a)$.

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \wedge \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.
- A proof of $\varphi_1 \vee \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.
- There is no proof of $\perp$ (falsity).
- A proof of $\forall x \varphi$ is a method (constructive function) which transforms any object a into a proof of $\varphi(a)$.
- A proof of $\exists x \varphi$ consists of an object a and a proof of $\varphi(a)$

The Brouwer-Heyting-Kolmogorov (BHK) interpretation

The BHK interpretation is an informal semantics for intuitionistic logic.

- A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.
- A proof of $\varphi_1 \wedge \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.
- A proof of $\varphi_1 \vee \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.
- There is no proof of $\perp$ (falsity).
- A proof of $\forall x \varphi$ is a method (constructive function) which transforms any object a into a proof of $\varphi(a)$.
- A proof of $\exists x \varphi$ consists of an object a and a proof of $\varphi(a)$

$\neg\varphi$ is an abbreviation for $\varphi \to \perp$.

The BHK interpretation

$$P \to P$$

The identity function is a function which transforms a proof of P into a proof of P. It is thus a proof of the formula $P \to P$.

The BHK interpretation

$$P \to P$$

The identity function is a function which transforms a proof of P into a proof of P. It is thus a proof of the formula $P \to P$.

$$(\lambda p : P.p) : P \to P$$

The BHK interpretation

$$P \to Q \to P$$

One proof of this formula is the function which given a proof p of P returns a constant function which ignores its argument and returns p.

The BHK interpretation

$$P \to Q \to P$$

One proof of this formula is the function which given a proof p of P returns a constant function which ignores its argument and returns p.

$$(\lambda p : P.\lambda q : Q.p) : P \to Q \to P$$

The BHK interpretation

$$P \to (P \to Q) \to Q$$

One proof of this formula is the function which given a proof p of P returns a function which given a function f transforming any proof of P into a proof of Q, applies this function to p, returning $f(p)$.

The BHK interpretation

$$P \to (P \to Q) \to Q$$

One proof of this formula is the function which given a proof p of P returns a function which given a function f transforming any proof of P into a proof of Q, applies this function to p, returning $f(p)$.

$$(\lambda p : P.\lambda f : P \to Q.fp) : P \to (P \to Q) \to P$$

The BHK interpretation

Excluded middle:

$$P \vee \neg P$$

In general, we cannot provide for an arbitrary P either a proof of P or a function which transforms a proof of P into a proof of $\perp$. It seems then that $P \vee \neg P$ is not intuitionistically provable.

The BHK interpretation

Peirce's law:

$$((P \to Q) \to P) \to P$$

If we are given just a function which transforms a proof of $P \to Q$ into a proof of P, then there doesn't seem to be any way of using this function to obtain a proof of P. To use the function, we would need to construct a proof of $P \to Q$, i.e., a function which converts any proof of P into a proof of Q. This does not seem possible in general. Hence, it seems Peirce's law is not intuitionistically provable.

The BHK interpretation

NOTE: The BHK interpretation is informal, so the above arguments do not conclusively establish if the formulas are intuitionistically provable. Such arguments may nonetheless be helpful to quickly determine whether intuitionistic provability is plausible.

Curry-Howard isomorphism

The Curry-Howard isomorphism is a formalisation of the BHK interpretation in the lambda-calculus.

Curry-Howard isomorphism

The Curry-Howard isomorphism is a formalisation of the BHK interpretation in the lambda-calculus.

- Formulas are types.
- Proofs are lambda-terms (roughly: total functional programs).

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.
- Each individual variable x has a (globally) fixed domain. We write $x : A$ if the domain of x is A.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.
- Each individual variable x has a (globally) fixed domain. We write $x : A$ if the domain of x is A.
- Predicates now need a type signature, e.g., $R : A \to B \to \texttt{Prop}$ means that R takes two arguments in domains A and B respectively.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.
- Each individual variable x has a (globally) fixed domain. We write $x : A$ if the domain of x is A.
- Predicates now need a type signature, e.g., $R : A \to B \to \texttt{Prop}$ means that R takes two arguments in domains A and B respectively. Then $R(x, y)$ is a valid formula only if $x : A$ and $y : B$.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.
- Each individual variable x has a (globally) fixed domain. We write $x : A$ if the domain of x is A.
- Predicates now need a type signature, e.g., $R : A \to B \to \texttt{Prop}$ means that R takes two arguments in domains A and B respectively. Then $R(x, y)$ is a valid formula only if $x : A$ and $y : B$.
- $\bot$ is an atomic formula and we define $\neg\varphi \equiv \varphi \to \bot$.

Curry-Howard isomorphism: formulas are types

- In predicate logic there is a single implicit domain of quantification: it contains all individual objects. In contrast, in dependent type theory we always quantify over elements of an explicitly given type.
- Already for the lambda-calculus presentation of intuitionistic predicate logic, we allow different domains of quantification $A, B, \ldots$ and include them in the syntax of quantifiers: $\forall x : A.\varphi$, $\exists x : A.\varphi$.
- Each individual variable x has a (globally) fixed domain. We write $x : A$ if the domain of x is A.
- Predicates now need a type signature, e.g., $R : A \to B \to \texttt{Prop}$ means that R takes two arguments in domains A and B respectively. Then $R(x, y)$ is a valid formula only if $x : A$ and $y : B$.
- $\perp$ is an atomic formula and we define $\neg\varphi \equiv \varphi \to \perp$.
- Aside of these changes, the formulas (types) are exactly the formulas of first-order logic.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.
- An individual term t is an individual variable (no function symbols for simplicity).

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.
- An individual term t is an individual variable (no function symbols for simplicity). Each individual term belongs to a unique domain: we write $t : A$ if the domain of t is A.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of <u>proof variables</u> $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A <u>context</u> Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.
- An <u>individual term</u> t is an individual variable (no function symbols for simplicity). Each individual term belongs to a unique domain: we write $t : A$ if the domain of t is A.
- A <u>proof term</u> $M, N, \ldots$ is a proof variable X, a lambda abstraction $\lambda X : \varphi.M$ or $\lambda x : A.M$, an application $M_1 M_2$ or Mt, a pair (M_1, M_2) or $[t, M]$, an injection $\texttt{inl}\, M$ or $\texttt{inr}\, M$, or a case expression (details later).

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.
- An individual term t is an individual variable (no function symbols for simplicity). Each individual term belongs to a unique domain: we write $t : A$ if the domain of t is A.
- A proof term $M, N, \ldots$ is a proof variable X, a lambda abstraction $\lambda X : \varphi.M$ or $\lambda x : A.M$, an application $M_1 M_2$ or Mt, a pair (M_1, M_2) or $[t, M]$, an injection $\texttt{inl}\, M$ or $\texttt{inr}\, M$, or a case expression (details later).
- A judgement has the form $\Gamma \vdash M : \varphi$ where Γ is a context, M is a proof term, and φ is a formula.

Curry-Howard isomorphism: proofs are lambda-terms

- We assume an infinite set of proof variables $X_1, X_2, \ldots$ distinct from individual variables $x, y, z, \ldots$
- A context Γ is a finite set of unique variable declarations $X : \varphi$.
  - "unique" means: if $X : \varphi$ and $X : \psi$ are in Γ then $\varphi = \psi$.
  - $\Gamma, X : \varphi$ is a notation for $\Gamma \cup \{X : \varphi\}$.
- An individual term t is an individual variable (no function symbols for simplicity). Each individual term belongs to a unique domain: we write $t : A$ if the domain of t is A.
- A proof term $M, N, \ldots$ is a proof variable X, a lambda abstraction $\lambda X : \varphi.M$ or $\lambda x : A.M$, an application $M_1 M_2$ or Mt, a pair (M_1, M_2) or $[t, M]$, an injection $\texttt{inl}\, M$ or $\texttt{inr}\, M$, or a case expression (details later).
- A judgement has the form $\Gamma \vdash M : \varphi$ where Γ is a context, M is a proof term, and φ is a formula.
- We now present in detail the proof terms and the typing rules.

Intermission: derivation rules

$$\frac{J_1 \quad \ldots \quad J_n}{J}\ S$$

- If we have derived the judgements $J_1, \ldots, J_n$ and the side condition S holds, then we can derive the judgement J.

Intermission: derivation rules

$$\frac{J_1 \quad \dots \quad J_n}{J}\ S$$

- If we have derived the judgements $J_1, \dots, J_n$ and the side condition S holds, then we can derive the judgement J.
- Sometimes we write the side condition(s) above the line together with the judgements $J_1, \dots, J_n$.

Intermission: derivation trees

$$\dfrac{\dfrac{\dfrac{}{J_3} \qquad \dfrac{\dfrac{}{J_5}}{J_4}}{J_1} \qquad \dfrac{}{J_2}}{J}$$

- To derive a judgement J we build a derivation tree using the derivation rules: each node is a valid application of a derivation rule.

Intermission: derivation trees

$$\dfrac{\dfrac{\dfrac{}{J_3}\quad\dfrac{\dfrac{}{J_5}}{J_4}}{J_1}\qquad\dfrac{}{J_2}}{J}$$

- To derive a judgement J we build a derivation tree using the derivation rules: each node is a valid application of a derivation rule.
- At the leaves of the tree we need rules with no judgements above the line.

Curry-Howard isomorphism

A proof of $\varphi_1 \to \varphi_2$ is a method (constructive function) which transforms any proof of φ_1 into a proof of φ_2.

$$\frac{\Gamma, X : \varphi_1 \vdash M : \varphi_2}{\Gamma \vdash (\lambda X : \varphi_1 . M) : \varphi_1 \to \varphi_2}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M_1 : \varphi \to \psi \quad \Gamma \vdash M_2 : \varphi}{\Gamma \vdash M_1 M_2 : \psi}\ (\to\mathrm{E})$$

introduction (how to prove) elimination (how to use)

Curry-Howard isomorphism

A proof of $\varphi_1 \wedge \varphi_2$ consists of a proof of φ_1 and a proof of φ_2.

$$\frac{\Gamma \vdash M_1 : \varphi_1 \quad \Gamma \vdash M_2 : \varphi_2}{\Gamma \vdash (M_1, M_2) : \varphi_1 \wedge \varphi_2}\ (\wedge \mathrm{I})$$

$$\frac{\Gamma \vdash M : \varphi_1 \wedge \varphi_2 \quad \Gamma, X_1 : \varphi_1, X_2 : \varphi_2 \vdash N : \psi}{\Gamma \vdash (\texttt{case}\, M\, \texttt{of}\, (X_1, X_2) \Rightarrow N) : \psi}\ (\wedge \mathrm{E})$$

Curry-Howard isomorphism

A proof of $\varphi_1 \vee \varphi_2$ consists of an indicator $i \in \{1, 2\}$ and a proof of φ_i.

$$\frac{\Gamma \vdash M : \varphi_1}{\Gamma \vdash \texttt{inl}\, M : \varphi_1 \vee \varphi_2}\ (\vee\mathrm{I}_1) \qquad \frac{\Gamma \vdash M : \varphi_2}{\Gamma \vdash \texttt{inr}\, M : \varphi_1 \vee \varphi_2}\ (\vee\mathrm{I}_2)$$

$$\frac{\Gamma \vdash M : \varphi_1 \vee \varphi_2 \quad \Gamma, X_1 : \varphi_1 \vdash N_1 : \psi \quad \Gamma, X_2 : \varphi_2 \vdash N_2 : \psi}{\Gamma \vdash (\texttt{case}\, M\, \texttt{of}\, \texttt{inl}\, X_1 \Rightarrow N_1 \mid \texttt{inr}\, X_2 \Rightarrow N_2) : \psi}\ (\vee\mathrm{E})$$

Curry-Howard isomorphism

There is no proof of $\perp$.

$$\frac{\Gamma \vdash M : \perp}{\Gamma \vdash (\mathtt{case}_\psi\, M) : \psi}\ (\perp\mathrm{E})$$

Curry-Howard isomorphism

A proof of $\forall x : A.\varphi$ is a method (constructive function) which transforms any object t in A into a proof of $\varphi(t)$.

$$\frac{\Gamma \vdash M : \varphi \quad x : A \quad x \notin \mathrm{FV}(\Gamma)}{\Gamma \vdash (\lambda x : A.M) : \forall x : A.\varphi}\ (\forall\mathrm{I}) \qquad \frac{\Gamma \vdash M : \forall x : A.\varphi \quad t : A}{\Gamma \vdash Mt : \varphi[t/x]}\ (\forall\mathrm{E})$$

Note: $\mathrm{FV}(\Gamma)$ is the set of all object variables occuring free in one of the formulas declared in Γ

Curry-Howard isomorphism

A proof of $\exists x : A.\varphi$ consists of an object t in A and a proof of $\varphi(t)$.

$$\frac{\Gamma \vdash M : \varphi[t/x] \quad t : A \quad x : A}{\Gamma \vdash [t, M] : \exists x : A.\varphi}\ (\exists\mathrm{I})$$

$$\frac{\Gamma \vdash M : \exists x : A.\varphi \quad \Gamma, X : \varphi \vdash N : \psi \quad x \notin \mathrm{FV}(\Gamma, \psi)}{\Gamma \vdash (\mathtt{case}\, M\, \mathtt{of}\, [x, X] \Rightarrow N) : \psi}\ (\exists\mathrm{E})$$

Note: $\mathrm{FV}(\Gamma, \psi)$ is the set of all object variables occuring free in one of the formulas declared in Γ or in ψ

Predicate logic: intuitionistic natural deduction

$$\frac{}{\Gamma, X : \varphi \vdash X : \varphi}\ (\mathrm{Ax})$$

$$\frac{\Gamma, X : \varphi_1 \vdash M : \varphi_2}{\Gamma \vdash (\lambda X : \varphi_1 . M) : \varphi_1 \to \varphi_2}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M_1 : \varphi \to \psi \quad \Gamma \vdash M_2 : \varphi}{\Gamma \vdash M_1 M_2 : \psi}\ (\to\mathrm{E})$$

$$\frac{\Gamma \vdash M_1 : \varphi_1 \quad \Gamma \vdash M_2 : \varphi_2}{\Gamma \vdash (M_1, M_2) : \varphi_1 \wedge \varphi_2}\ (\wedge\mathrm{I}) \qquad \frac{\Gamma \vdash M : \varphi_1 \wedge \varphi_2 \quad \Gamma, X_1 : \varphi_1, X_2 : \varphi_2 \vdash N : \psi}{\Gamma \vdash (\mathsf{case}\ M\ \mathsf{of}\ (X_1, X_2) \Rightarrow N) : \psi}\ (\wedge\mathrm{E})$$

$$\frac{\Gamma \vdash M : \varphi_1}{\Gamma \vdash \mathsf{inl}\ M : \varphi_1 \vee \varphi_2}\ (\vee\mathrm{I}_1) \qquad \frac{\Gamma \vdash M : \varphi_2}{\Gamma \vdash \mathsf{inr}\ M : \varphi_1 \vee \varphi_2}\ (\vee\mathrm{I}_2)$$

$$\frac{\Gamma \vdash M : \varphi_1 \vee \varphi_2 \quad \Gamma, X_1 : \varphi_1 \vdash N_1 : \psi \quad \Gamma, X_2 : \varphi_2 \vdash N_2 : \psi}{\Gamma \vdash (\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ X_1 \Rightarrow N_1 \mid \mathsf{inr}\ X_2 \Rightarrow N_2) : \psi}\ (\vee\mathrm{E})$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash (\mathsf{case}_\psi\ M) : \psi}\ (\bot\mathrm{E})$$

$$\frac{\Gamma \vdash M : \varphi \quad x : A \quad x \notin \mathrm{FV}(\Gamma)}{\Gamma \vdash (\lambda x : A . M) : \forall x : A . \varphi}\ (\forall\mathrm{I}) \qquad \frac{\Gamma \vdash M : \forall x : A . \varphi \quad t : A}{\Gamma \vdash M t : \varphi[t/x]}\ (\forall\mathrm{E})$$

$$\frac{\Gamma \vdash M : \varphi[t/x] \quad t : A \quad x : A}{\Gamma \vdash [t, M] : \exists x : A . \varphi}\ (\exists\mathrm{I})$$

$$\frac{\Gamma \vdash M : \exists x : A . \varphi \quad \Gamma, X : \varphi \vdash N : \psi \quad x \notin \mathrm{FV}(\Gamma, \psi)}{\Gamma \vdash (\mathsf{case}\ M\ \mathsf{of}\ [x, X] \Rightarrow N) : \psi}\ (\exists\mathrm{E})$$

Predicate logic: classical natural deduction

Classical natural deduction is intuitionistic natural deduction extended with any one of the following rules

Predicate logic: classical natural deduction

Classical natural deduction is intuitionistic natural deduction extended with any one of the following rules:

- classical falsity elimination

$$\frac{\Gamma, X : \neg\varphi \vdash M : \bot}{\Gamma \vdash \langle X : \neg\varphi \rangle M : \varphi}$$

Predicate logic: classical natural deduction

Classical natural deduction is intuitionistic natural deduction extended with any one of the following rules:

- classical falsity elimination

$$\frac{\Gamma, X : \neg\varphi \vdash M : \bot}{\Gamma \vdash \langle X : \neg\varphi \rangle M : \varphi}$$

- excluded middle axiom

$$\frac{}{\Gamma \vdash \mathtt{exm}_{\varphi} : \varphi \vee \neg\varphi}$$

Predicate logic: classical natural deduction

Classical natural deduction is intuitionistic natural deduction extended with any one of the following rules:

- classical falsity elimination

$$\frac{\Gamma, X : \neg\varphi \vdash M : \bot}{\Gamma \vdash \langle X : \neg\varphi \rangle M : \varphi}$$

- excluded middle axiom

$$\frac{}{\Gamma \vdash \mathtt{exm}_\varphi : \varphi \vee \neg\varphi}$$

- double negation axiom

$$\frac{}{\Gamma \vdash \mathtt{dneg}_\varphi : \neg\neg\varphi \to \varphi}$$

Predicate logic: classical natural deduction

Classical natural deduction is intuitionistic natural deduction extended with any one of the following rules:

- classical falsity elimination

$$\frac{\Gamma, X : \neg\varphi \vdash M : \bot}{\Gamma \vdash \langle X : \neg\varphi \rangle M : \varphi}$$

- excluded middle axiom

$$\frac{}{\Gamma \vdash \mathtt{exm}_\varphi : \varphi \vee \neg\varphi}$$

- double negation axiom

$$\frac{}{\Gamma \vdash \mathtt{dneg}_\varphi : \neg\neg\varphi \to \varphi}$$

However, with such a “naive” extension we lose the correspondence between proofs and functional programs.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.
- A formula φ is a (classical) semantic consequence of a set of formulas Δ, denoted $\Delta \models_c \varphi$, if for every interpretation I such that $I \models \psi$ for each $\psi \in \Delta$, we have $I \models \varphi$.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.
- A formula φ is a (classical) semantic consequence of a set of formulas Δ, denoted $\Delta \models_c \varphi$, if for every interpretation I such that $I \models \psi$ for each $\psi \in \Delta$, we have $I \models \varphi$.
- Let $|\Gamma|$ denote the set of formulas declared in the context Γ.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.
- A formula φ is a (classical) semantic consequence of a set of formulas Δ, denoted $\Delta \models_c \varphi$, if for every interpretation I such that $I \models \psi$ for each $\psi \in \Delta$, we have $I \models \varphi$.
- Let $|\Gamma|$ denote the set of formulas declared in the context Γ.
- Write $\Gamma \vdash_c M : \varphi$ if $\Gamma \vdash M : \varphi$ may be derived in a classical natural deduction system.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.
- A formula φ is a (classical) semantic consequence of a set of formulas Δ, denoted $\Delta \models_c \varphi$, if for every interpretation I such that $I \models \psi$ for each $\psi \in \Delta$, we have $I \models \varphi$.
- Let $|\Gamma|$ denote the set of formulas declared in the context Γ.
- Write $\Gamma \vdash_c M : \varphi$ if $\Gamma \vdash M : \varphi$ may be derived in a classical natural deduction system.

Theorem

$\Gamma \vdash_c M : \varphi$ *for some* M *if and only if* $|\Gamma| \models_c \varphi$.

Predicate logic: classical natural deduction

- Classical natural deduction is sound and complete for classical first-order semantics.
- A formula φ is a (classical) <u>semantic consequence</u> of a set of formulas Δ, denoted $\Delta \models_c \varphi$, if for every interpretation I such that $I \models \psi$ for each $\psi \in \Delta$, we have $I \models \varphi$.
- Let $|\Gamma|$ denote the set of formulas declared in the context Γ.
- Write $\Gamma \vdash_c M : \varphi$ if $\Gamma \vdash M : \varphi$ may be derived in a classical natural deduction system.

Theorem

$\Gamma \vdash_c M : \varphi$ *for some* M *if and only if* $|\Gamma| \models_c \varphi$.

Conclusion: intuitionistic logic is a subsystem of classical logic with a "natural" computational interpretation.