A Formalization of the Naive Type Theory

Agnieszka Kozubek

University of Warsaw

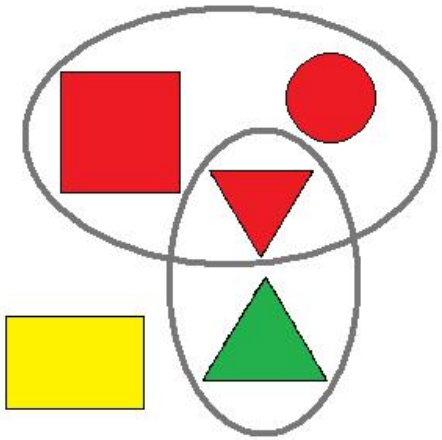
31 May 2012

Set theory: foundations of contemporary mathematics

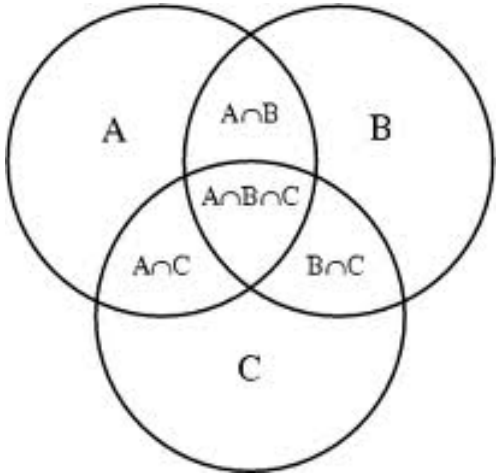
Set theory: foundations of contemporary mathematics

We teach it at all levels

kindergarten



school



university

$\{\langle x, y\rangle \in \mathbb{R}^2 \mid x^2 + y^2 > 1 \to y > 0\}$

$\{z \in \mathbb{R} \mid \exists x \forall x (x = 1)\}$

$\{\langle x, y\rangle \in \mathbb{R}^2 \mid \forall z (z < 0) \to y < 0\}$

$\{z \in \mathbb{R} \mid \forall x \exists x (x = 1)\}$

$\{\langle x, y\rangle \in \mathbb{R}^2 \mid \exists z (z^2 > 1 \to y^2 \leqslant \frac{1}{4})\}$

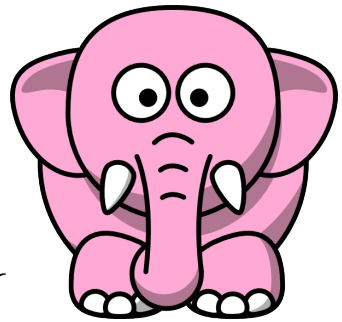
$\{x \in \mathbb{R} \mid \forall x \exists x (x = 1)\}$

$\{\langle x, y\rangle \in \mathbb{R}^2 \mid \exists z (z^2 > 1 \wedge y^2 \leqslant 42)\}$

$\{x \in \mathbb{R} \mid \forall x \exists x (x \neq 1)\}$

Extensionality

Extensionality

$$\{x \in \mathbb{Q} \mid x^2 = 2\} = \{$$  in this room $\}$

Universes and predicates

Set theory identifies two notions:

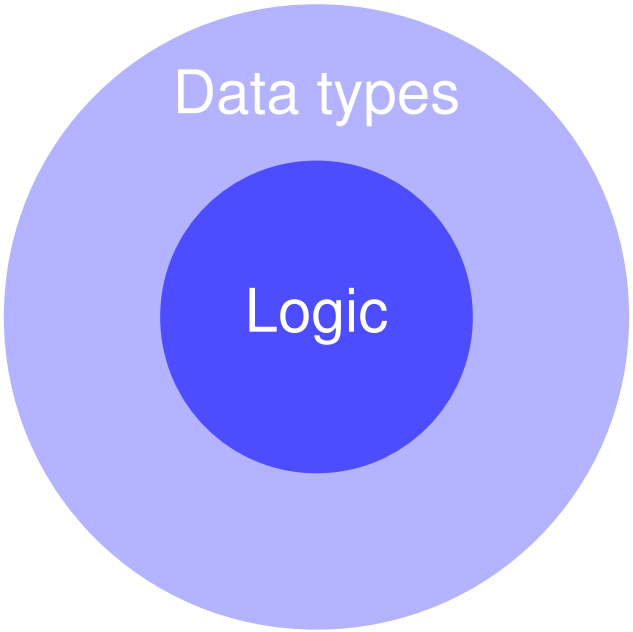
- Set as a domain (universe);
- Set as a (materialized) predicate.

The goal

A formal system which

- distinguishes between universes and predicates
- powersets are types, subsets are first-class objects
- intensional

The structure of the system: two layers



Logic: Less Naive Type Theory

$$\mathcal{S} = *^t, *^p, \Box^t, \Box^p \qquad \mathcal{A} = *^t : \Box^t, *^p : \Box^p$$

$\mathcal{R} = \{$

$(*^t, *^t, *^t)$	function space	$\tau \to \sigma$
$(*^p, *^p, *^p)$	implication	$\varphi \to \psi$
$(*^t, *^p, *^p)$	first-order logic	$\forall x : \tau.\varphi$
$(*^t, \Box^t, \Box^t)$	dependent types	$\forall n : \tau.List(n)$
$(\Box^p, *^p, *^p)$	second-order logic	$\forall p : *^p.\varphi(p)$
$(*^t, \Box^p, *^t)$	powersets are types	$\tau \to *^p : *^t$
	subsets are objects	$\{x : \tau \mid \varphi(x)\}$

$\}$

Layer 2: data types

- natural numbers,
- lists,
- graphs,
- pairs,
- empty type,
- ...

Layer 2: data types

- natural numbers,
- lists,
- graphs,
- pairs,
- empty type,
- ...

All defined by induction!

Inductive types

Example (Lists)

$$\begin{aligned} &\mathit{Inductive\ List} : *^t := \\ &\qquad \mathit{nil} : \mathit{List} \mid \\ &\qquad \mathit{cons} : A \to \mathit{List} \to \mathit{List}. \end{aligned}$$

Inductive types

Example (Lists)

$$\begin{aligned} &\mathit{Inductive\ List} : *^t := \\ &\qquad \mathit{nil} : \mathit{List} \mid \\ &\qquad \mathit{cons} : A \to \mathit{List} \to \mathit{List}. \end{aligned}$$

Also if $A = \tau \to *^p$!

Inductive types: extend the language

Types

Inductive types and predicates: $\mathrm{Ind}(X : A)\{\vec{C}\}$,

Terms

- inductive objects: $\mathrm{Constr}(n, I)$,
- eliminators: $\mathrm{Elim}(I, Q, M)\{\vec{f}\}$.

Iota reduction (example)

$\mathrm{Elim}(\mathit{List}, Q, \mathit{nil})\{f_1 \mid f_2\} \to_\iota f_1$
$\mathrm{Elim}(\mathit{List}, Q, \mathit{cons}\ h\ t)\{f_1 \mid f_2\} \to_\iota f_2\ h\ t\ \mathrm{Elim}(\mathit{List}, Q, t)\{f_1 \mid f_2\}$

Inductive types: typing rules

$$(Ind)\ \frac{\Gamma, X : A \vdash C_i : A}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}\ \text{(positivity condition)}$$

$$(Intro)\ \frac{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}{\Gamma \vdash \mathrm{Constr}(n, \mathrm{Ind}(X : A)\{\vec{C}\}) : C_n(\mathrm{Ind}(X : A)\{\vec{C}\})}$$

$$(Elim)\ \frac{\Gamma \vdash t : \mathrm{Ind}(X : A)\{\vec{C}\} \qquad \Gamma \vdash Q : \mathrm{Ind}(X : A)\{\vec{C}\} \to s \qquad \Gamma \vdash f_n : \triangle\{C_n(\mathrm{Ind}(X : A)\{\vec{C}\}), Q, \mathrm{Constr}(n, \mathrm{Ind}(X : A)\{\vec{C}\})\}}{\Gamma \vdash \mathrm{Elim}(\mathrm{Ind}(X : A)\{\vec{C}\}, Q, t)\{\vec{f}\} : Qt}$$

Inductive types: typing rules

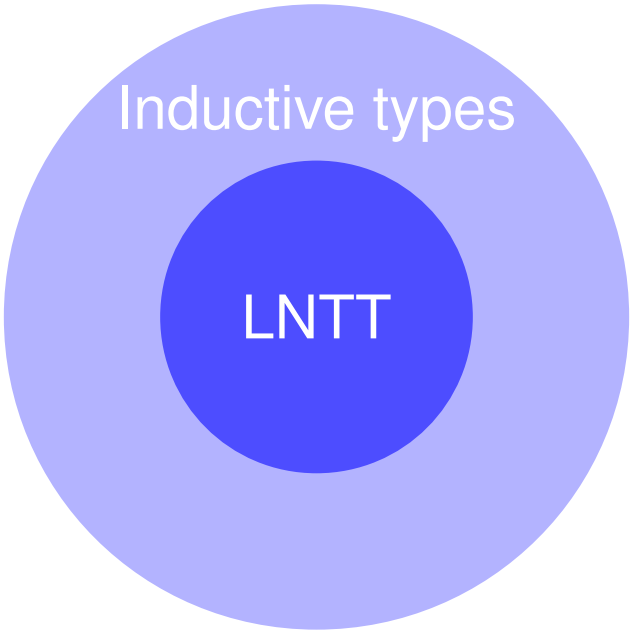
$$(Ind)\ \frac{\Gamma, X : A \vdash C_i : A}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}\ \text{(positivity condition)}$$

$$(Intro)\ \frac{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}{\Gamma \vdash \mathrm{Constr}(n, \mathrm{Ind}(X : A)\{\vec{C}\}) : C_n(\mathrm{Ind}(X : A)\{\vec{C}\})}$$

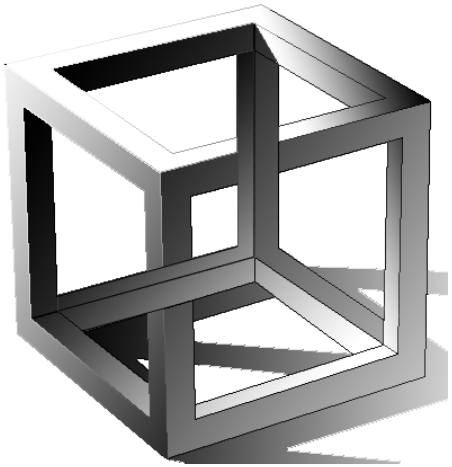
$$(Elim)\ \frac{\Gamma \vdash t : \mathrm{Ind}(X : A)\{\vec{C}\} \qquad \Gamma \vdash Q : \mathrm{Ind}(X : A)\{\vec{C}\} \to s \qquad \Gamma \vdash f_n : \triangle\{C_n(\mathrm{Ind}(X : A)\{\vec{C}\}), Q, \mathrm{Constr}(n, \mathrm{Ind}(X : A)\{\vec{C}\})\}}{\Gamma \vdash \mathrm{Elim}(\mathrm{Ind}(X : A)\{\vec{C}\}, Q, t)\{\vec{f}\} : Qt}$$

$$(\mathrm{Conv})\ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_{\beta\iota} B'}{\Gamma \vdash A : B'}$$

The two layers



The main question

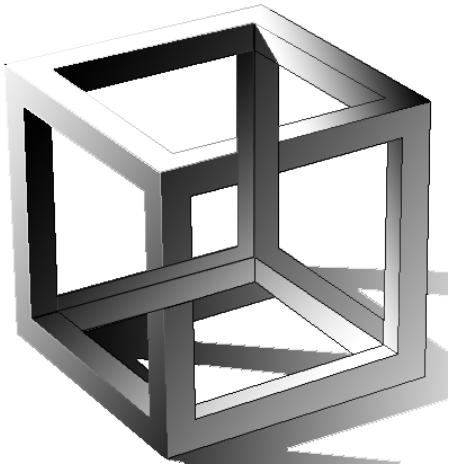


Is the system consistent?

First attempt: Naive Type Theory

Naive Type Theory is inconsistent. (Geuvers)

The main question



Is the system consistent?

First attempt: Naive Type Theory

Naive Type Theory is inconsistent. (Geuvers)

How do you prove consistency?

Theorem

Every strongly normalizing system is consistent.

Strong normalization

Every reduction sequence in the system is finite.

You can establish consistency by proving the strong normalization property.

The main theorem

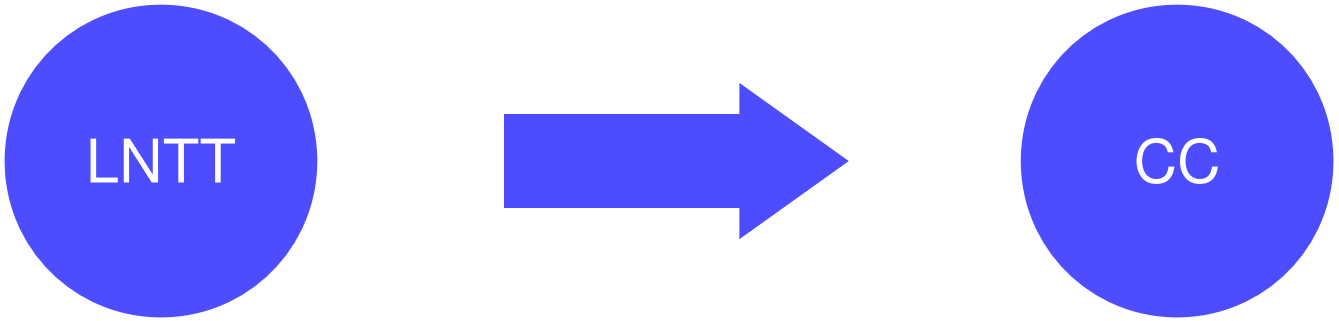
Theorem

LNTT with inductive types is strongly normalizing.

Two parts of the proof

1. SN for LNTT
2. SN for the full system

Part 1: Strong Normalization for LNTT



well-known
system with SN

Part 2: Girard's candidates

type τ $\longrightarrow$ $[\tau]_{\Delta,\xi,\rho} \subseteq SN$
$\{M \mid ...\}$

Main lemma

If $\Gamma \vdash M : \tau$ then $\rho(M) \in [\tau]_{\Delta,\xi,\rho}$.

Part 2: Girard's candidates

$$\text{type } \tau \longrightarrow [\Gamma \vdash \tau]_{\Delta,\xi,\rho} \subseteq SN^{\Delta}_{\rho(\tau)}$$
$$\{\Delta \vdash M \mid \ldots\}$$

Main lemma

If $\Gamma \vdash M : \tau$ then $(\Delta \vdash \rho(M)) \in [\Gamma \vdash \tau]_{\Delta,\xi,\rho}$.

Part 2: Girard's candidates

$$\text{type } \tau \longrightarrow [\Gamma \vdash \tau]_{\Delta,\xi,\rho} \subseteq SN^{\Delta}_{\rho(\tau)}$$
$$\{\Delta \vdash M \mid \ldots\}$$

Main lemma

If $\Gamma \vdash M : \tau$ then $(\Delta \vdash \rho(M)) \in [\Gamma \vdash \tau]_{\Delta,\xi,\rho}$.

We provide interpretation for

- type-like terms (types, formulas, kinds, sorts)
- type constructors
- large inductive objects and eliminations

Part 2: Girard's candidates

Intepretation for sorts, kinds, products, type constructors is standard.

Part 2: Girard's candidates

- interpretation for inductive type

$$[\vdash \mathit{List}] = \mathit{lfp}(F)$$
$$F(S) = \mathit{Base} \cup \{\Delta \vdash M \mid M \in \mathit{SN}, M \to^* \mathit{nil} \vee$$
$$M \to^* \mathrm{cons}\ h\ t \wedge h \in [\tau] \wedge t \in S\}$$

- interpretation for inductive predicate

$$[\vdash \mathit{Even}] = \mathit{lfp}(H)$$
$$H(S) = \mathit{Base} \cup \{\Delta \vdash M \mid M \in \mathit{SN},$$
$$\text{for each type } q \text{ and its interpretation } Q$$
$$\text{for each pattern matching branch } f_i \in [\Delta\{Q, X\}]_{X:=s}$$
$$\mathrm{Elim}(\mathit{Even}, q, M)\{\vec{f}\} \in Q\}$$

Part 2: Girard's candidates

- interpretation for inductive objects

$$[\mathit{cons}\ X\ l] = \langle [X], [l] \rangle$$

- interpretation for elimination terms

$$[\vdash \mathrm{Elim}(\mathit{List}, Q, (\mathit{cons}\ h\ t))\{f_1 \mid f_2\}] = [\vdash f_2\ h\ t\ \mathrm{Elim}(\mathit{List}, Q, t)\{f_1 \mid f_2\}]$$

Part 2: Girard's candidates

- interpretation for inductive objects

$$[\mathit{cons}\ X\ l] = \langle [X], [l] \rangle$$

- interpretation for elimination terms

$$\begin{aligned}[\vdash \mathrm{Elim}(\mathit{List}, Q, (\mathit{cons}\ h\ t))\{f_1 \mid f_2\}] \\ = [\vdash f_2\ h\ t\ \mathrm{Elim}(\mathit{List}, Q, t)\{f_1 \mid f_2\}] \\ = [\vdash f_2]\ [h]\ [t]\ [\vdash \mathrm{Elim}(\mathit{List}, Q, t)\{f_1 \mid f_2\}]\end{aligned}$$

The main result

Theorem

LNTT with inductive types is strongly normalizing.

Corollary

LNTT with inductive types is consistent.

Thank you!