# Type theory vs. Set theory

Alexandre Miquel

UNIVERSIDAD DE LA REPUBLICA URUGUAY
FACULTAD DE INGENIERIA
EQUIPO DE LÓGICA UDELAR

January 9th, 2019 – CASS 2020 – San José del Maipo

# Plan

1. A primer in set theory
2. Types in sets
3. Sets in types
4. Going further

# A bit of history

1878 While studying the properties of trigonometric series (derived sets), Georg **Cantor** (1845–1918) discovers ordinal numbers

Starting point of set theory: cardinal numbers, continuum hypothesis

1879 Gottlob **Frege**'s (1848–1925) *Begriffsschrift* ("concept-script")
Ancestor of the predicate calculus

1903 First attempt by **Frege** to formalize Cantor's set theory
Bertrand **Russell** (1872–1970) shows its inconsistency

1908 Ernst **Zermelo**'s (1871–1953) new axiomatization of set theory (Z)
Also introduces the axiom of choice (AC)

1922 Abraham **Fraenkel** (1891–1965) and Thoralf **Skolem** (1887–1963) independently introduce the replacement scheme (Z $\to$ ZF)

# What is set theory?

- Set theory describes a (nonempty) universe whose objects are sets
  Here: set = pure set = set whose elements are (pure) sets
- The set-theoretic universe is governed by two primitive relations
  - **Equality:** $x = y$ (where both $x$ and $y$ are sets)
  - **Membership:** $x \in y$ (where both $x$ and $y$ are sets)
- Sets are loose enough to encode most mathematical concepts: tuples, relations, functions, numbers... and of course: sets
- Many axiomatizations of set theory. Most notably:
  - Zermelo set theory (Z)
  - Zermelo-Fraenkel set theory (ZF) (= Zermelo + replacement)
- Many additional axioms:
  - Foundation axiom (FA), Axiom of choice (AC)
  - Continuum Hypothesis (CH), Generalized Cont. Hyp. (GCH)

# The language of set theory

- Set theory is traditionally presented using the language of first-order logic (with equality):

**Formulas** $\phi, \psi \ ::= \ x = y \ \mid \ x \in y \ \mid \ \neg\phi \ \mid \ \phi \Rightarrow \psi \ \mid \ \phi \wedge \psi \ \mid \ \phi \vee \psi \ \mid \ \forall x\, \phi \ \mid \ \exists x\, \phi$

No constant/function symbol; the only terms are variables

- Standard abbreviations:

$$\begin{array}{rcl} x \neq y & :\equiv & \neg(x = y) \\ x \notin y & :\equiv & \neg(x \in y) \\ \forall x \in a\ \phi(x) & :\equiv & \forall x\,(x \in a \Rightarrow \phi(x)) \\ \exists x \in a\ \phi(x) & :\equiv & \exists x\,(x \in a \wedge \phi(x)) \\ \exists! x\ \phi(x) & :\equiv & \exists x\, \phi(x) \ \wedge \ \forall x\, \forall x'\,(\phi(x) \wedge \phi(x') \Rightarrow x = x') \\ x \subseteq y & :\equiv & \forall z\,(z \in x \Rightarrow z \in y) \end{array}$$

# The axioms of Zermelo-Fraenkel set theory

| | |
|---|---|
| **Extensionality** | $\forall a\ \forall b\ \big(\forall x\ (x \in a \Leftrightarrow x \in b)\ \Rightarrow\ a = b\big)$ |
| **Pairing** | $\forall a\ \forall b\ \exists c\ \forall x\ (x \in c\ \Leftrightarrow\ x = a \vee x = b)$ |
| **Comprehension** | $\forall a\ \exists b\ \forall x\ (x \in b\ \Leftrightarrow\ x \in a \wedge \phi(x))$ for each formula $\phi(x)$ |
| **Union** | $\forall a\ \exists b\ \forall x\ (x \in b\ \Leftrightarrow\ \exists y \in a\ \ x \in y)$ |
| **Powerset** | $\forall a\ \exists b\ \forall x\ (x \in b\ \Leftrightarrow\ x \subseteq a)$ |
| **Infinity** | $\exists a\ \big(\exists x \in a\ \forall z\ (z \notin x)\ \wedge\ \forall x \in a\ \exists y \in a\ \forall z\ (z \in y \Leftrightarrow z \in x \vee z = x)\big)$ |
| **Replacement** | $\forall a\ \big(\forall x \in a\ \exists! y\ \psi(x,y)\ \Rightarrow\ \forall x \in a\ \exists y \in b\ \psi(x,y)\big)$ for each formula $\psi(x,y)$ |
| **Foundation** | $\forall a\ \big((\exists x\ x \in a)\ \Rightarrow\ \exists x \in a\ \forall y \in a\ y \notin x\big)$ |

# Introducing notations

The “official language” of set theory contains no constant/function symbol: the only terms are variables

This is the user’s job to introduce his/her own Skolem symbols

- For instance, replace the “official” pairing axiom by

  **Pairing** $\forall a\ \forall b\ \forall x\ (x \in \{a, b\} \ \Leftrightarrow\ x = a \vee x = b)$

  where $\{\_, \_\}$ is a new binary function symbol

- And similarly for $\bigcup \_$ (union), $\mathfrak{P}(\_)$ (powerset), $\Omega$ (infinity)

Such extensions are known to be conservative, in the sense that:

*If a formula of the official language is provable using Skolem symbols, then it is provable in the official formalism (i.e. without Skolem symbols)*

# Example: Skolemized Zermelo set theory ($Z^{sk}$)

$$\begin{array}{llllllllll}
\textbf{Terms} & t, u & ::= & x & \mid & \Omega & \mid & \mathfrak{P}(t) & \mid & \bigcup t \\
& & \mid & \{t_1, t_2\} & \mid & \{x \in t : \phi\} & & & & \\
\textbf{Formulas} & \phi, \psi & ::= & t = u & \mid & t \in u & \mid & \neg\phi & \mid & \phi \Rightarrow \psi \\
& & \mid & \phi \wedge \psi & \mid & \phi \vee \psi & \mid & \forall x\, \phi & \mid & \exists x\, \phi
\end{array}$$

**Extensionality** $\quad \forall a\, \forall b\, \big(\forall x\, (x \in a \Leftrightarrow x \in b) \;\Rightarrow\; a = b\big)$

**Pairing** $\quad \forall a\, \forall b\, \forall x\, (x \in \{a, b\} \;\Leftrightarrow\; x = a \vee x = b)$

**Comprehension** $\quad \forall a\, \forall x\, (x \in \{x \in a : \phi(x)\} \;\Leftrightarrow\; x \in a \wedge \phi(x))$

for each formula $\phi(x)$

**Union** $\quad \forall a\, \forall x\, (x \in \bigcup a \;\Leftrightarrow\; \exists y \in a\;\, x \in y)$

**Powerset** $\quad \forall a\, \forall x\, (x \in \mathfrak{P}(a) \;\Leftrightarrow\; x \subseteq a)$

**Infinity** $\quad \varnothing \in \Omega \;\;\wedge\;\; \forall x \in \Omega\;\; s(x) \in \Omega$

where $\;s(x) \;:\equiv\; \bigcup\{x, \{x, x\}\} \;\; (=\; x \cup \{x\})$

**Theorem [M. 2005]:** (I)$Z^{sk}$ is a conservative extension of (I)Z

# The expressiveness of set theory

**Intuition:** Sets are a clay to sculpt mathematical objects

$$a \cup b \;:=\; \bigcup\{a, b\} \qquad\qquad \varnothing \;:=\; \{x \in \Omega : x \neq x\}$$

$$a \cap b \;:=\; \{x \in a : x \in b\}$$

$$(x, y) \;:=\; \{\{x\}, \{x, y\}\}$$

$$A \times B \;:=\; \left\{p \in \mathfrak{P}(\mathfrak{P}(A \cup B)) \;:\; \exists x \in A\; \exists y \in B \;\; p = (x, y)\right\}$$

$$B^A \;:=\; \left\{f \in \mathfrak{P}(A \times B) \;:\; \forall x \in A\; \exists! y \in B \;\; (x, y) \in f\right\}$$

$$f(x) \;:=\; \bigcup\left\{y \in \bigcup\bigcup f \;:\; (x, y) \in f\right\}$$

$$A/\!\sim \;:=\; \left\{c \in \mathfrak{P}(A) \;:\; \exists x \in A\; \forall y \in A\; (y \in c \Leftrightarrow y \sim x)\right\}$$

$$0 \;:=\; \varnothing \qquad\qquad s(x) \;:=\; x \cup \{x\}$$

$$\mathbb{N} \;:=\; \left\{n \in \Omega \;:\; \forall Z\, (0 \in Z \wedge \forall x\, (x \in Z \Rightarrow s(x) \in Z) \Rightarrow n \in Z)\right\}$$

**Drawback:** $\sqrt{\pi} \;\cap\; \begin{pmatrix} 0 & \mathbb{R} \\ \cos & \mathcal{L}^2(\mathbb{R}) \end{pmatrix}$ is a well-formed set

# The modularity of set theory

Set theory is highly modular:

- It may be classical (Z, ZF) or intuitionistic (IZ, IZF)
- All set theories contain at least **Extensionality**, **Pairing**, **Union** and **Comprehension**... but the other axioms are optional
  - It may be impredicative or predicative (remove **Powerset**)
  - It may be infinitary or finitist (remove **Infinity**)
    **Note:** ZF − **Infinity** is equiconsistent to Peano arithmetic
- There are even non-extensional presentations of set theory
  (i.e. based on intensional membership $\varepsilon$, no primitive equality)

Even classical set theory (ZF) is highly customizable:

- Foundation or Antifoundation? Choice or Determinacy?
- + many axioms for large cardinals

**Motto:** Whatever your philosophy, there is a set theory for you!

# Plan

# Translating type theory into set theory

- **Idea:** Translate each type-theoretic construct into its obvious set-theoretic equivalent ("forget about typing constaints"):

| Type theory | Set theory |
|---|---|
| Functions (as algorithms) | Functions (as graphs) |
| Dependent products | Generalized Cartesian products |
| Predicative universes $\mathsf{Type}_i$ | Grothendieck universes |
| Inductively defined types | Inductively defined sets |
| Propositions | Booleans |
| Proofs | A single object (proof irrelevance) |

- Through this translation, we may regard type theory as a decidable fragment of set theory, with an explicit algorithmic contents
- However, this simple translation is incompatible with univalence

# Set-theoretic functions

- In set theory, a function is a set of pairs $f$ such that the binary relation "$(x, y) \in f$" is functional in $x$:

$$\begin{array}{lcl} f \text{ function} & :\equiv & \forall p \in f\ \exists x\ \exists y\ p = (x, y) \quad \wedge \\ & & \forall x\ \forall y\ \forall y'\ ((x, y) \in f \wedge (x, y') \in f \Rightarrow y = y') \end{array}$$

- Each function $f$ has a domain and an image:

$$\begin{array}{rcl} \mathrm{dom}(f) & := & \{x \in \bigcup\bigcup f \ :\ \exists y\ (x, y) \in f\} \\ \mathrm{img}(f) & := & \{y \in \bigcup\bigcup f \ :\ \exists x\ (x, y) \in f\} \end{array}$$

- Function application is defined by:

$$f(x) \ := \ \bigcup\{y \in \bigcup\bigcup f \ :\ (x, y) \in f\}$$

For all $x \in \mathrm{dom}(f)$, we have: $\quad f(x) = y \quad \text{iff} \quad (x, y) \in f$

# Interpreting abstraction and application

- Given a set $A$ and a set expression $b(x)$ depending on $x \in A$, we let

$$\lambda x \in A \,.\, b(x) \quad := \quad \{(x, b(x)) \;:\; x \in A\}$$

- Application is defined by:

$$f(a) \;:=\; \bigcup \Big\{ y \in \bigcup\bigcup f \;:\; (a, y) \in f \Big\}$$

**Fact:** If $a \in A$, then: $\big(\lambda x \in A \,.\, b(x)\big)(a) \;=\; b(a)$

# Dependent products as generalized Cartesian products

- Given sets $A$ and $B$, we let:

$$f : A \to B \;:\equiv\; f \text{ function} \;\wedge\; \mathrm{dom}(f) = A \;\wedge\; \mathrm{img}(f) \subseteq B$$
$$B^A \;:=\; \big\{ f \in \mathfrak{P}(A \times B) \;:\; (f : A \to B) \big\}$$

- More generally, if $(B_x)_{x \in A}$ is a family of sets indexed by $A$:

$$\prod_{x \in A} B_x \;:=\; \left\{ f \in \Big( \bigcup_{x \in A} B_x \Big)^A \;:\; \forall x \in A \;\; f(x) \in B_x \right\}$$

**Note:** Family of sets indexed by $A$ = function of domain $A$

**Fact:** If $f \in \prod_{x \in A} B_x$ and $a \in A$, then $f(a) \in B_a$

- Particular case where $B_x = B$ for all $x \in A$: $\prod_{x \in A} B_x \;=\; B^A$

# Grothendieck universes (1/3)

**Definition (Grothendieck universe)**

A set $\mathcal{U}$ is a Grothendieck universe when:

1. If $A \in \mathcal{U}$, then $A \subseteq \mathcal{U}$ (i.e. $\mathcal{U}$ is transitive)
2. $\mathbb{N} \in \mathcal{U}$
3. If $A \in \mathcal{U}$, then $\mathfrak{P}(A) \in \mathcal{U}$
4. If $A \in \mathcal{U}$ and $B_x \in \mathcal{U}$ for all $x \in A$, then $\bigcup_{x \in A} B_x \in \mathcal{U}$

In particular:

- If $A, B \in \mathcal{U}$, then $\{A, B\} \in \mathcal{U}$, $\bigcup A \in \mathcal{U}$ and $\mathfrak{P}(A) \in \mathcal{U}$
- If $A \in \mathcal{U}$ and $B \subseteq A$, then $B \in \mathcal{U}$
- If $A \in \mathcal{U}$ and $B_x \in \mathcal{U}$ for all $x \in A$, then $\prod_{x \in A} B_x \in \mathcal{U}$

# Grothendieck universes (2/3)

- Intuitively, a Grothendieck universe is a set that behaves as a set-theoretic universe inside the set-theoretic universe

**Theorem**

Each Grothendieck universe $\mathcal{U}$ is closed under all the set-theoretic constructions that are definable in ZF. In particular:

$$(\mathcal{U}, \in_{|\mathcal{U}}) \models \mathsf{ZF}$$

- From Gödel's second incompleteness theorem, the existence of Grothendieck universes cannot be proved in ZF
  (unless ZF is inconsistent)

- In what follows, we shall assume the existence of Grothendieck universes (with a suitable axiom)

# Grothendieck universes (3/3)

Grothendieck universes are related to strongly inaccessible cardinals

**Definition (Strongly inaccessible cardinal)**

A cardinal $\lambda$ is strongly inaccessible if:

1. $\lambda > \aleph_0$
2. If $\kappa < \lambda$, then $2^\kappa < \lambda$
3. If $\kappa < \lambda$ and $\mu_\alpha < \lambda$ for all $\alpha < \kappa$, then $\sup_{\alpha<\kappa} \mu_\alpha < \lambda$

**Proposition**

$$\mathcal{U} \text{ Grothendieck universe} \quad \Rightarrow \quad \mathrm{Card}(\mathcal{U}) \text{ strongly inaccessible}$$

$$\lambda \text{ strongly inaccessible} \quad \Rightarrow \quad V_\lambda \text{ Grothendieck universe}$$

**Recall:** The cumulative hierarchy $(V_\alpha)_\alpha$ is defined by

$$V_0 := \varnothing, \qquad V_{\alpha+1} := \mathfrak{P}(V_\alpha), \qquad V_\alpha := \bigcup_{\beta<\alpha} V_\beta \quad (\text{if } \alpha \text{ limit})$$

# Interpreting predicative universes $\mathsf{Type}_i$

- We now work in $\mathsf{ZF} + \mathsf{SI}^\omega$, where $\mathsf{SI}^\omega$ is the axiom:

  *There exist infinitely many strongly inaccessible cardinals*

- Let $(\lambda_i)_{i\in\omega}$ be the $\omega$ first strongly inaccessible cardinals, and write
$$\mathcal{U}_i \ := \ V_{\lambda_i} \qquad \text{(for each } i \in \omega\text{)}$$

- By construction, for all $i \in \omega$ we have:
  1. $\mathcal{U}_i \ \in \ \mathcal{U}_{i+1}$
  2. If $A \in \mathcal{U}_i$ and $B_x \in \mathcal{U}_i$ for all $x \in A$, then $\prod_{x\in A} B_x \ \in \ U_i$

$\Rightarrow$ Interpret each $\mathsf{Type}_i$ by $\mathcal{U}_i$

# How to interpret the sort Prop of propositions?

- The sort Prop of propositions enjoys two properties:
  1. Prop : $\text{Type}_0$
  2. If $U(x) : \text{Prop}$ for all $x : T$ ($T$ any), then $\Pi x : T \,.\, U(x) \;:\; \text{Prop}$
- Hence we need a set $\mathcal{U}_*$ such that:
  1. $\mathcal{U}_* \in \mathcal{U}_0$
  2. If $B_x \in \mathcal{U}_*$ for all $x \in A$ ($A$ any), then $\prod_{x \in A} B_x \in \mathcal{U}_*$
- **Intuition:** Elements of $\mathcal{U}_*$ should be small enough to be closed under arbitrary dependent products
  - $\mathcal{U}_* =$ class of all compact sets?
    Does not work, since $\mathcal{U}_*$ is not a set
  - $\mathcal{U}_* = V_\omega =$ set of all hereditarily finite sets?
    Does not work, since $V_\omega$ is not closed under dependent products

# The semantics of proof-irrelevance (1/2)

**Solution:** $B$ is a proposition iff $\mathrm{Card}(B) < 2$ (0 or 1 element)

**Proposition (Closure under dependent product)**

If $\mathrm{Card}(B_x) < 2$ for all $x \in A$ ($A$ any), then $\mathrm{Card}\left(\prod_{x\in A} B_x\right) < 2$:

$$\prod_{x\in A} B_x = \begin{cases} \varnothing & \text{if } B_x = \varnothing \text{ for some } x \in A \\ \{(x \mapsto b_x)\} & \text{if } B_x = \{b_x\} \text{ for all } x \in A \end{cases}$$

**Intuitions:**

- $B_x = \varnothing$ is the false proposition
- $B_x = \{b_x\}$ is a true proposition, with only one proof $b_x$

$\Rightarrow$ Set-theoretic $\prod_{x\in A} B_x$ mimics the Tarski semantics of $\forall$

# The semantics of proof-irrelevance (2/2)

- We take: $\mathcal{U}_* \;:=\; \{\varnothing, \{\bullet\}\} \;\; (= \;\mathfrak{P}(\{\bullet\}))$ ($\bullet$ = canonical proof)
- **Beware!** To ensure that $\mathcal{U}_*$ is closed under dependent products, we need to identify each constant function $(x \in A \mapsto \bullet)$ with $\bullet$
- For that we let $\bullet := \varnothing$ and replace each graph $f$ of a function by its trace $\mathrm{Tr}(f)$ (notion that comes from domain theory)

$$\mathrm{Tr}(f) \;=\; \{(x, z) \;:\; x \in \mathrm{dom}(f) \;\wedge\; z \in f(x)\}$$

(So that $\mathrm{Tr}(x \mapsto \bullet) = \varnothing = \bullet$)

- Interpretation of $\lambda$, app. and $\Pi$ has to be modified accordingly
- Thanks to this trick, we get:

**Proposition (Closure under dependent product)**

If $B_x \in \mathcal{U}_*$ for all $x \in A$ ($A$ any), then $\prod_{x \in A} B_x \;\in\; \mathcal{U}_*$

# Inductive definitions

Inductive definitions work the same way in set theory as in type theory

- Each inductive definition (in a Grothendieck universe $\mathcal{U}$) consists to define a monotonic operator $\Phi : \mathcal{U} \to \mathcal{U}$ ("the constructors") and then to take its least fixed point in $\mathcal{U}$
- For instance, the type of binary trees with leaves in $A$

```
Inductive bintree (A : Type) : Type :=
| Leaf : A → bintree A
| Node : bintree A → bintree A → bintree A
```

  is represented by the family of operators $\Phi_A : \mathcal{U} \to \mathcal{U}$ defined by

$$\Phi(X) \;=\; A + (X \times X)$$

  indexed by $A \in \mathcal{U}$ (here: $+$ is disjoint union)

- Positivity conditions (on the types of the constructors' arguments) ensure that $\Phi$ is monotonic and has a least fixed point in $\mathcal{U}$

# Glueing everything together (1/2)

Let $\mathscr{M} := \bigcup_{i\in\omega} \mathcal{U}_i$ (transitive set)

- A valuation in $\mathscr{M}$ is any function $\rho \in \mathscr{M}^{\mathcal{X}}$
  (where $\mathcal{X}$ is the set of all type-theoretic variables)

- To each raw term $M$, we associate its interpretation

$$(\rho \mapsto [\![M]\!]_\rho) \;:\; \mathscr{M}^{\mathcal{X}} \rightharpoonup \mathscr{M} \qquad \text{(partial function)}$$

  that is defined by structural induction on $M$:

$$\begin{array}{rcl@{\qquad}rcl}
[\![x]\!]_\rho & := & \rho(x) & [\![\lambda x : T\,.\,M]\!]_\rho & := & \mathrm{Tr}(v \in [\![T]\!]_\rho \mapsto [\![M]\!]_{\rho,x\leftarrow v}) \\
[\![\mathrm{Prop}]\!]_\rho & := & \mathcal{U}_* & [\![MN]\!]_\rho & := & \mathrm{TrApp}([\![M]\!]_\rho,\ [\![N]\!]_\rho) \\
[\![\mathrm{Type}_i]\!]_\rho & := & \mathcal{U}_i & [\![\Pi x : T\,.\,U]\!] & := & \prod_{v\in[\![T]\!]_\rho} [\![U]\!]_{\rho,x\leftarrow v} \\
 & \ldots & & & \ldots &
\end{array}$$

- **Note:** $[\![M]\!]_\rho$ (that is not always defined) only depends on the values of $\rho$ corresponding to the free variables of $M$

# Glueing everything together (2/2)

- The interpretation is extended to typing contexts:

$$[\![\Gamma]\!] \;:=\; \left\{\rho \in \mathscr{M}^{\mathcal{X}} \;:\; \begin{array}{l}[\![T]\!]_\rho \text{ defined and } \rho(x) \in [\![T]\!]_\rho \\ \text{for each declaration } (x : T) \in \Gamma\end{array}\right\}$$

**Theorem (Soundness)**

If a typing judgment $\Gamma \vdash M : T$ is derivable in the Calculus of Inductive Constructions (CIC), then for all valuations $\rho \in [\![\Gamma]\!]$:

1. $[\![M]\!]_\rho$ and $[\![T]\!]_\rho$ are defined, and
2. $[\![M]\!]_\rho \;\in\; [\![T]\!]_\rho$

- Recall that $\quad \mathrm{False} \;:=\; \Pi X : \mathrm{Prop} \,.\, X \quad (: \mathrm{Prop})$

**Corollary (Consistency)**

There is no closed proof-term $M$ such that $\vdash M : \mathrm{False}$

**Proof:** Indeed, we have $[\![\mathrm{False}]\!] \;=\; [\![\Pi X : \mathrm{Prop} \,.\, X]\!] \;=\; \prod_{A \in \mathcal{U}_*} A \;=\; \varnothing$

# An absolute consistency proof

- In the former slides, we constructed the simplest set-theoretic model of CIC: the proof-irrelevant model
- From this, we got a proof of consistency of CIC within $\mathsf{ZF} + \mathsf{SI}^\omega$ (used as a "metatheory"):

$$\mathsf{ZF} + \mathsf{SI}^\omega \ \vdash \ \mathsf{Cons}(\mathsf{CIC})$$

- This is a result of absolute consistency, which is written

$$\mathsf{CIC} \ < \ \mathsf{ZF} + \mathsf{SI}^\omega$$

- More generally, we write $A < B$ when the consistency of $A$ is provable in $B$. This statement implicitly assumes that:
    1. The theory $A$ is recursive (to be formalizable in Heyting arithmetic)
    2. The theory $B$ contains Heyting arithmetic (so it can formalize $A$)

  The relation $A < B$ is known to be irreflexive (by the second incompleteness theorem) and transitive
- **Recall:** This construction is incompatible with univalence

# Plan

# Translating set theory into type theory

- Translating type theory into set theory is quite easy

  Translate typed constructions into their set-theoretic equivalents...
  ... and forget about type constraints!

- Translating set theory into type theory is much more difficult

  *How to embed a world with little constraints (set theory)*
  *into a world with many constraints (type theory)?*

- **Idea:** Define a universal type for representing sets

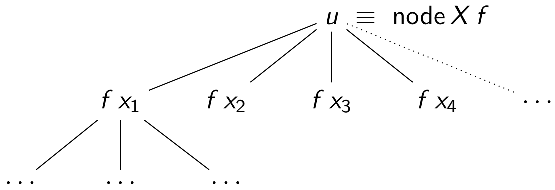
- Two known methods for achieving this:
  1. Sets as well-founded trees [Aczel '77, Werner '97]
  2. Sets as pointed graphs [M. 2000]

# Sets as well-founded trees [Aczel '77, Werner '98]

- Consider the inductive definition (Coq):

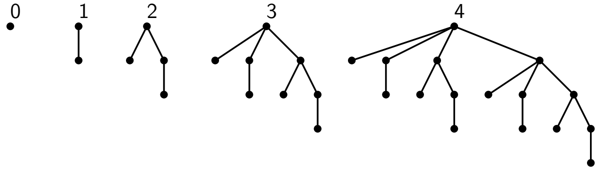
$$\begin{array}{l} \text{Inductive U} \ : \ \text{Type} \ := \\ \mid \quad \text{node} \ : \ \forall X : \text{Set},\ (X \to \text{U}) \to \text{U}. \end{array}$$

- Each object $u : \text{U}$ is of the form $u \equiv \text{node}\,X\,f$, where:
  - $X : \text{Set}$ is the type of branching (index type)
  - $f : X \to \text{U}$ is the family of children of $t$

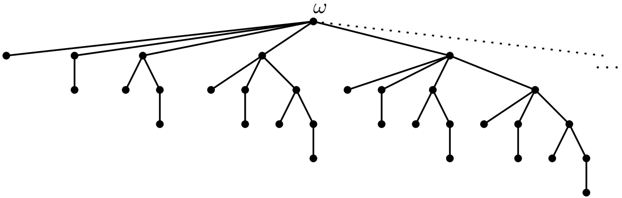


# Examples of sets as well-founded trees

- **Von Neumann numerals:** $0 = \varnothing$, $1 = \{0\} = \{\varnothing\}$, $2 = \{0, 1\} = \{\varnothing, \{\varnothing\}\}$, $3 = \{0, 1, 2\} = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$, etc.



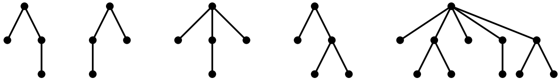

- **The set $\omega$ of natural numbers:** $\omega = \{0, 1, 2, 3, 4, \ldots\}$




- **Exercise:** Implement 0, 1, 2, ..., $\omega$ as objects of type U in Coq

# Identifying well-founded trees

- The same set can be represented by many well-founded trees:



$\Rightarrow$ Need an extensional collapse

- Recursive definition of extensional equality (Coq):

$$\begin{array}{l}\text{Fixpoint eqv } (u : \mathrm{U})\ (v : \mathrm{U})\ :\ \text{Prop}\ := \\ \quad \text{let } {}'\text{node}\, X\, f\ :=\ u \text{ in} \\ \quad \text{let } {}'\text{node}\, Y\, g\ :=\ v \text{ in} \\ \quad (\forall x : X,\ \exists y : Y,\ \text{eqv}\ (f\, x)\ (g\, y))\ \wedge \\ \quad (\forall y : Y,\ \exists x : X,\ \text{eqv}\ (f\, x)\ (g\, y)).\end{array}$$

- **Exercise:** Prove that 'eqv' is an equivalence relation on U.

# Defining membership

- Membership relation ($\in$) is defined by:

$$\begin{array}{l}\text{Definition mem } (u\ :\ \mathrm{U})\ (v\ :\ \mathrm{U})\ :\ \mathrm{Prop}\ := \\ \quad \text{let }{}'\text{node } Y\, g\ :=\ v \text{ in} \\ \quad \exists y : Y,\ \text{eqv}\ u\ (g\ y).\end{array}$$

- **Exercise:** Prove that 'mem' is compatible with 'eqv':

$$\forall u : \mathrm{U},\ \forall v : \mathrm{U},\ \forall u' : \mathrm{U},\ \text{mem}\ u\ v \to \text{eqv}\ u\ u' \to \text{mem}\ u'\ v$$
$$\forall u : \mathrm{U},\ \forall v : \mathrm{U},\ \forall v' : \mathrm{U},\ \text{mem}\ u\ v \to \text{eqv}\ v\ v' \to \text{mem}\ u\ v'$$

- **Remark:** The fact that 'mem' is compatible with 'eqv' implies that any first-order formula $\phi(u_1, \ldots, u_n)$ constructed from the only primitive predicates 'eqv' and 'mem' is compatible with 'eqv' in each argument $u_i$.

  Hence 'eqv' behaves as Leibniz equality w.r.t. the language of set theory

- **Exercise:** Prove the axiom of extensionality:

$$\forall u : \mathrm{U},\ \forall v : \mathrm{U},\ \ (\forall w : \mathrm{U},\ \text{mem}\ w\ u\ \leftrightarrow\ \text{mem}\ w\ v)\ \ \to\ \ \text{eqv}\ u\ v$$

# Proving the axioms of set theory (1/2)

- **Exercise:** Interpreting $=/\in$ as eqv/mem, prove in Coq:

1. The pairing axiom
$$\forall a : \mathrm{U},\ \forall b : \mathrm{U},\ \exists c : \mathrm{U},\ \forall x : \mathrm{U},\ \mathrm{mem}\, x\, c\ \leftrightarrow\ \mathrm{eqv}\, x\, a \lor \mathrm{eqv}\, x\, b$$
2. The union axiom:
$$\forall a : \mathrm{U},\ \exists b : \mathrm{U},\ \forall x : \mathrm{U},\ \mathrm{mem}\, x\, b\ \leftrightarrow\ \exists y : \mathrm{U},\ \mathrm{mem}\, y\, a \land \mathrm{mem}\, x\, y$$
3. The comprehension scheme:
$$\forall P : \mathrm{U} \to \mathrm{Prop},\ \mathrm{compat}\, P\ \to$$
$$\forall a : \mathrm{U},\ \exists b : \mathrm{U},\ \forall x : \mathrm{U},\ \mathrm{mem}\, x\, b\ \leftrightarrow\ \mathrm{mem}\, x\, a \land P\, x$$
where $\mathrm{compat}\, P\ := \ \forall x : \mathrm{U},\ \forall x' : \mathrm{U},\ P\, x \to \mathrm{eqv}\, x\, x' \to P\, x'$
4. The powerset axiom
$$\forall a : \mathrm{U},\ \exists b : \mathrm{U},\ \forall x : \mathrm{U},\ \mathrm{mem}\, x\, b\ \leftrightarrow\ \mathrm{sub}\, x\, a$$
where $\mathrm{sub}\, x\, y\ := \ \forall z : \mathrm{U},\ \mathrm{mem}\, z\, x \to \mathrm{mem}\, z\, y$
5. The infinity axiom

- **Exercise:** Prove that the relation ‘mem’ is well-founded on U

  This property is classically equivalent to the Foundation axiom

# Proving the axioms of set theory (2/2)

- The former results show that all axioms, and thus all theorems of Intuitionistic Zermelo set theory (IZ) are provable in Coq

$$\mathrm{IZ} \ < \ \mathrm{CIC}_2 \quad (= \text{CIC with 2 universes})$$ [Werner '97]

- The replacement scheme does not hold, but we have bits of it

- For instance, it is known that the set

$$X \ = \ \bigcup_{n\in\omega} \mathfrak{P}^n(\omega) \ = \ \bigcup_{n\in\omega} \underbrace{\mathfrak{P}(\cdots\mathfrak{P}}_{n}(\omega)\cdots)$$

  is not definable in (I)Z, and requires at least an instance of the replacement scheme to be constructed (in IZF)

- **Exercise:** Construct a well-founded tree that represents $X$

# Some results

- The representation of sets as well-founded trees was introduced by Peter Aczel (1977) in Martin-Löf type theory (MLTT).

  From this representation, Aczel extracted a new (constructive) axiomatization of set theory: Constructive Zermelo Fraenkel (CZF)

  $$\mathsf{CZF} \ < \ \mathsf{MLTT}\ \text{(with 1 universe)}$$

  **Note:** In CZF, the Powerset axiom is replaced by an Exponentiation axiom

- Using the same representation of sets in the Coq proof assistant, Benjamin Werner (1997) observed that:

  $$\mathsf{IZ} + \text{bits of replacement} \ < \ \mathsf{CIC}_2 \quad (= \mathsf{CIC}\ \text{with 2 universes})$$

- Proof-theoretically, we actually have

  $$\mathsf{CZF} \ < \ \mathsf{MLTT}\ \text{(with 1 universe)} \ < \ \mathsf{HA2} \ < \ \mathsf{HA}\omega \ < \ \mathsf{IZ} \ < \ \mathsf{CIC}_2$$

  where HA2 (resp. HA$\omega$) is second-order (resp. higher-order) Heyting arithmetic

# From well-founded trees to pointed graphs

- The representation of sets as well-founded trees is convenient to give set-theoretic lower bounds to type theories with inductive types
- Its main drawback is that it crucially relies on the presence of generalized inductive types in the target formalism

  *How to represent sets in a formalism without inductive types, for instance in a Pure Type System (PTS)?*

$\Rightarrow$ Representing sets as pointed graphs [M. 2000]

# Sets as pointed graphs

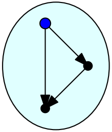
[M. 2000]

A pointed graph is a triple $(X, A, a)$ where:

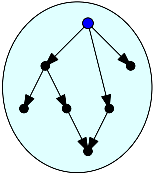
1. $X$ is a type (the type of vertices)
2. $A : X \to X \to \mathrm{Prop}$ is a binary relation on $X$ (the arc relation)
3. $a : X$ is a distinguished point (the root of the p. graph)

**Examples:**

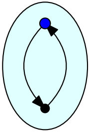
$2 = \{\varnothing, \{\varnothing\}\}$



$3 = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$



$x = \{\{x\}\}$

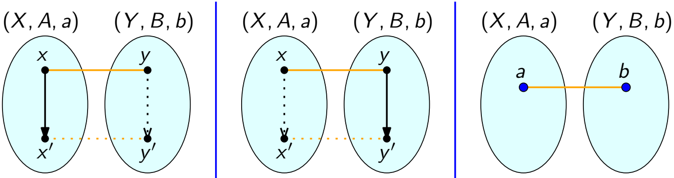


**Note:** Pointed graphs allow the representation of cyclic sets, or more generally: non-well-founded sets

# Equality as bisimilarity

- Again, a given set can be represented by many pointed graphs
- Extensional collapse is achieved via the relation of bisimilarity

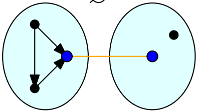
$$
\begin{array}{ll}
\multicolumn{2}{l}{(X, A, a) \approx (Y, B, b) \;\; :\equiv} \\
& \exists R : X \to Y \to \mathrm{Prop}, \\
(1) & \quad (\forall x\, x' : X,\ \forall y : Y,\ \ A\, x'\, x \wedge R\, x\, y \ \to\ \exists y' : Y,\ R\, x'\, y' \wedge B\, y'\, y) \ \wedge \\
(2) & \quad (\forall y\, y' : Y,\ \forall x : X,\ \ B\, y'\, y \wedge R\, x\, y \ \to\ \exists x' : X,\ R\, x'\, y' \wedge A\, x'\, x) \ \wedge \\
(3) & \quad R\, a\, b
\end{array}
$$



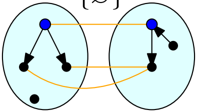

Condition (1) | Condition (2) | Condition (3)

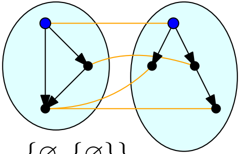
# Example of bisimulations

$\varnothing$

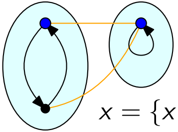


$\{\varnothing\}$





$\{\varnothing, \{\varnothing\}\}$

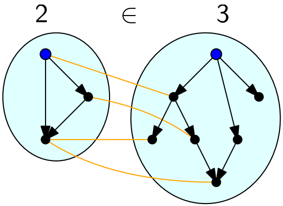


$x = \{x\}$

# Membership as shifted bisimilarity

- Extensional membership ($\in$) is interpreted as shifted bisimilarity

$$(X, A, a) \in (Y, B, b) \;:\equiv$$
$$\exists b' : Y,\; B\, b'\, b \wedge (X, A, a) \approx (Y, B, b')$$



# Compatibility with bisimilarity

- In what follows, we write:

$$\begin{array}{lcl}\forall(X,A,a),\ \cdots & :\equiv & \forall X:\mathrm{Type},\ \forall A:X\to X\to\mathrm{Prop},\ \forall a:X,\ \cdots\\ \exists(X,A,a),\ \cdots & :\equiv & \exists X:\mathrm{Type},\ \exists A:X\to X\to\mathrm{Prop},\ \exists a:X,\ \cdots\end{array}$$

- **Exercise:** Prove that $\in$ is compatible with $\approx$

$$\begin{array}{l}\forall(X,A,a),\ \forall(Y,B,b),\ \forall(X',A',a'),\\ \qquad (X,A,a)\in(Y,B,b)\to(X,A,a)\approx(X',A,a')\to(X',A',a')\in(Y,B,b)\end{array}$$

$$\begin{array}{l}\forall(X,A,a),\ \forall(Y,B,b),\ \forall(Y',B',b'),\\ \qquad (X,A,a)\in(Y,B,b)\to(Y,B,b)\approx(Y',B',b')\to(X,A,a)\in(Y',B',b')\end{array}$$

- **Exercise:** Prove the axiom of extensionality

$$\begin{array}{l}\forall(X,A,a),\ \forall(Y,B,b),\\ \quad (\forall(Z,C,c),\ (Z,C,c)\in(X,A,a)\leftrightarrow(Z,C,c)\in(Y,B,b))\\ \quad\ \to\quad (X,A,a)\approx(Y,B,b)\end{array}$$

- **Exercise:** Prove the other Zermelo axioms (without Foundation), without using any inductive datatype of Coq

# The Antifoundation axiom (AFA) [Aczel '88]

The sets-as-pointed-graphs representation is incompatible with the Foundation axiom, but it satisfies the Antifoundation axiom (AFA)

- (Going back to set theory) Given a digraph $G = (V, A)$, we call a reification of $G$ any family of sets $(x_i)_{i \in V}$ such that

$$x_i \;=\; \{x_j \;:\; (j, i) \in A\} \qquad\qquad \text{for all } i \in V$$

- Using Replacement, it is easy to see that each well-founded digraph has a unique reification. On the other hand, the Foundation axiom implies that non well-founded digraphs have no reification

  This naturally motivates the:

**Antifoundation axiom (AFA)**

Every digraph has a unique reification

- Using this axiom, we can prove (for instance) that there exists a unique set $x$ such that $x = \{x\}$

# Some results

- This representation of sets allows us to prove all axioms/theorems of Intuitionistic Zermelo set theory with Antifoundation (IZ + AFA) in the Calculus of Constructions with 3 universes ($CC_3$)

  $$\mathrm{IZ} + \mathrm{AFA} \ < \ \mathrm{CC}_3$$ [M. 2001]

  Recall that $CC_3$ has no inductive datatypes at all

  Uses a mixture of impredicative encodings (for defining the connectives and $\exists$) and predicative encodings (to define the carriers of pointed graphs)

- Why 3 universes?

  1. $\mathrm{Type}_0$ is the "bootstrap universe" (contains no provably infinite type)
     The bootstrap universe allows us to construct the first infinite type
     $N \ := \ \Pi X : \mathrm{Type}_0,\ X \to (X \to X) \to X \ : \ \mathrm{Type}_1$
  2. $\mathrm{Type}_1$ is the type of carriers of pointed graphs
  3. Top universe $\mathrm{Type}_2$ is only used for the rule $(\mathrm{Type}_2, \mathrm{Prop}, \mathrm{Prop})$, that allows to express quantifications over all sets

# Plan

# Going further...

A natural question:

*What is the smallest type theory that allows us to get* IZ (+ AFA) *via the sets-as-pointed-graphs representation?*

Such a type theory should contain:

- An impredicative sort Prop of propositions
- A predicative universe Type to build the carriers of pointed graphs
- A infinite datatype $\mathsf{Nat} : \mathsf{Type}$ (or a bootstrap universe $\mathsf{Type}_0 : \mathsf{Type}$)
- A top sort ($\mathsf{Type}'$) to allow quantifying over all $X : \mathsf{Type}$
  (via the PTS axiom $\mathsf{Type} : \mathsf{Type}'$ and rule $(\mathsf{Type}', \mathsf{Prop}, \mathsf{Prop})$)

# HOL$^{+}$: Syntax & typing

**Types** $\tau, \sigma \quad ::= \quad \alpha \quad | \quad \mathsf{Prop} \quad | \quad \mathsf{Nat} \quad | \quad \tau \to \sigma$

**Object-terms**

$$\begin{array}{rcl} M, N, A, B & ::= & x \quad | \quad \lambda x^{\tau} . M \quad | \quad M\,N \\ & | & A \Rightarrow B \quad | \quad \forall x^{\tau}.A \quad | \quad \forall \alpha.A \\ & | & \mathtt{0} \quad | \quad \mathtt{S} \quad | \quad \mathtt{rec}_{\tau} \end{array}$$

**Typing contexts** $\Sigma \quad ::= \quad x_1 : \tau_1, \ldots, x_n : \tau_n \qquad (x_1 \not\equiv x_i \text{ if } i \neq j)$

**Typing rules:**

$$\frac{}{\Sigma \vdash x : \tau}\,{\scriptstyle (x:\tau)\in\Sigma} \qquad \frac{\Sigma, x : \tau \vdash M : \sigma}{\Sigma \vdash \lambda x^{\tau} . M : \tau \to \sigma} \qquad \frac{\Sigma \vdash M : \tau \to \sigma \quad \Sigma \vdash N : \tau}{\Sigma \vdash MN : \sigma}$$

$$\frac{\Sigma \vdash A : \mathsf{Prop} \quad \Sigma \vdash B : \mathsf{Prop}}{\Sigma \vdash A \Rightarrow B : \mathsf{Prop}} \qquad \frac{\Sigma, x : \tau \vdash A : \mathsf{Prop}}{\Sigma \vdash \forall x^{\tau}.A : \mathsf{Prop}} \qquad \frac{\Sigma \vdash A : \mathsf{Prop}}{\Sigma \vdash \forall \alpha.A : \mathsf{Prop}}\,{\scriptstyle \alpha \notin TV(A)}$$

$$\frac{}{\Sigma \vdash \mathtt{0} : \iota} \qquad \frac{}{\Sigma \vdash \mathtt{S} : \iota \to \iota} \qquad \frac{}{\Sigma \vdash \mathtt{rec}_{\tau} : \tau \to (\iota \to \tau \to \tau) \to \iota \to \tau}$$

**Underlying PTS:** $F\omega + \mathsf{Nat} + (\mathsf{Type} : \mathsf{Type}') + (\mathsf{Type}', \mathsf{Prop}, \mathsf{Prop})$

# HOL$^{+}$: Reduction

- One step reduction is the congruence $\succ$ defined from the rules

$$\begin{array}{rcl} (\lambda x^{\tau} . M)\, N & \succ & M\{x := N\} \\ \mathtt{rec}_{\tau}\, M_0\, M_1\, 0 & \succ & M_0 \\ \mathtt{rec}_{\tau}\, M_0\, M_1\, (\mathtt{S}\, N) & \succ & M_1\, N\, (\mathtt{rec}_{\tau}\, M_0\, M_1\, N) \end{array}$$

  As usual, we write

  - $\succ^*$ the reflexive-transitive closure of $\succ$ (grand reduction)
  - $\cong$ the reflexive-symmetric-transitive closure of $\succ$ (conversion)

- **Church-Rosser + Subject reduction**

# $\text{HOL}^{+}$: Deduction

**Logical contexts:** $\Gamma \quad := \quad A_1, \ldots, A_n$

**Deduction rules:**

$$\frac{\Sigma \vdash A_i : \mathrm{Prop} \quad (1 \le i \le n)}{\langle\Sigma\rangle\ A_1, \ldots, A_n \vdash A_i} \qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \qquad \Sigma \vdash A' : \mathrm{Prop}}{\langle\Sigma\rangle\ \Gamma \vdash A'}\ A \cong A'$$

$$\frac{\langle\Sigma\rangle\ \Gamma, A \vdash B}{\langle\Sigma\rangle\ \Gamma \vdash A \Rightarrow B} \qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \Rightarrow B \qquad \langle\Sigma\rangle\ \Gamma \vdash A}{\langle\Sigma\rangle\ \Gamma \vdash B}$$

$$\frac{\langle\Sigma, x : \tau\rangle\ \Gamma \vdash A}{\langle\Sigma\rangle\ \Gamma \vdash \forall x^{\tau}.\, A} \qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash \forall x^{\tau}.\, A \qquad \Sigma \vdash N : \tau}{\langle\Sigma\rangle\ \Gamma \vdash A\{x := N\}}$$

$$\frac{\langle\Sigma\rangle\ \Gamma \vdash A}{\langle\Sigma\rangle\ \Gamma \vdash \forall \alpha.\, A}\ \alpha \notin TV(\Sigma, \Gamma) \qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash \forall \alpha.\, A}{\langle\Sigma\rangle\ \Gamma \vdash A\{\alpha := \tau\}}$$

# Zermelo set theory (recall)

$$\textbf{Formulas} \quad \phi, \psi \quad ::= \quad x = y \;\mid\; x \in y \;\mid\; \neg\phi \;\mid\; \phi \Rightarrow \psi \;\mid\; \phi \wedge \psi \;\mid\; \phi \vee \psi \;\mid\; \forall x\, \phi \;\mid\; \exists x\, \phi$$

**Extensionality** $\forall a\, \forall b\, \big(\forall x\, (x \in a \Leftrightarrow x \in b) \;\Rightarrow\; a = b\big)$

**Pairing** $\forall a\, \forall b\, \exists c\, \forall x\, (x \in c \;\Leftrightarrow\; x = a \vee x = b)$

**Comprehension** $\forall a\, \exists b\, \forall x\, (x \in b \;\Leftrightarrow\; x \in a \wedge \phi(x))$
for each formula $\phi(x)$

**Union** $\forall a\, \exists b\, \forall x\, (x \in b \;\Leftrightarrow\; \exists y \in a\;\, x \in y)$

**Powerset** $\forall a\, \exists b\, \forall x\, (x \in b \;\Leftrightarrow\; x \subseteq a)$

**Infinity** $\exists a\, \big(\exists x \in a\, \forall z\, (z \notin x) \;\wedge\; \forall x \in a\, \exists y \in a\, \forall z\, (z \in y \Leftrightarrow z \in x \vee z = x)\big)$

Z / IZ $=$ classical/intuitionistic Zermelo set theory

# $\mathrm{HOL}^+$ (recall)

**Types** $\tau, \sigma \quad ::= \quad \alpha \quad | \quad \mathsf{Prop} \quad | \quad \mathsf{Nat} \quad | \quad \tau \to \sigma$

**Object-terms**
$$\begin{array}{rcl} M, N, A, B & ::= & x \quad | \quad \lambda x^{\tau} . M \quad | \quad M\, N \\ & | & A \Rightarrow B \quad | \quad \forall x^{\tau}.A \quad | \quad \forall \alpha.A \\ & | & \mathtt{0} \quad | \quad \mathtt{S} \quad | \quad \mathtt{rec}_{\tau} \end{array}$$

**Typing contexts** $\Sigma \quad ::= \quad x_1 : \tau_1, \ldots, x_n : \tau_n \qquad (x_1 \not\equiv x_i \text{ if } i \neq j)$

**Typing rules:**

$$\frac{}{\Sigma \vdash x : \tau}\ (x{:}\tau) \in \Sigma \qquad \frac{\Sigma, x : \tau \vdash M : \sigma}{\Sigma \vdash \lambda x^{\tau} . M : \tau \to \sigma} \qquad \frac{\Sigma \vdash M : \tau \to \sigma \quad \Sigma \vdash N : \tau}{\Sigma \vdash MN : \sigma}$$

$$\frac{\Sigma \vdash A : \mathsf{Prop} \quad \Sigma \vdash B : \mathsf{Prop}}{\Sigma \vdash A \Rightarrow B : \mathsf{Prop}} \qquad \frac{\Sigma, x : \tau \vdash A : \mathsf{Prop}}{\Sigma \vdash \forall x^{\tau}.A : \mathsf{Prop}} \qquad \frac{\Sigma \vdash A : \mathsf{Prop}}{\Sigma \vdash \forall \alpha.A : \mathsf{Prop}}\ \alpha \notin TV(A)$$

$$\frac{}{\Sigma \vdash \mathtt{0} : \iota} \qquad \frac{}{\Sigma \vdash \mathtt{S} : \iota \to \iota} \qquad \frac{}{\Sigma \vdash \mathtt{rec}_{\tau} : \tau \to (\iota \to \tau \to \tau) \to \iota \to \tau}$$

**Underlying PTS:** $F\omega + \mathsf{Nat} + (\mathsf{Type} : \mathsf{Type}') + (\mathsf{Type}', \mathsf{Prop}, \mathsf{Prop})$

# Equiconsistency

**Theorem** [M. 2009]

- The theories

$$\mathsf{IZ}, \qquad \mathsf{IZ} + \mathsf{FA}, \qquad \mathsf{IZ} + \mathsf{AFA} \qquad \text{and} \qquad \mathsf{HOL}^+$$

  are equiconsistent
- Moreover, these theories prove the very same arithmetic formulas

**Remarks:**

- FA (Foundation axiom) is an axiom scheme in intuitionitic logic (i.e.: "the relation $\in$ is well-founded")
- The equiconsistency $\mathsf{Z} \approx \mathsf{Z} + \mathsf{FA} \approx \mathsf{Z} + \mathsf{AFA}$ (in classical logic) was already known before [Esser & Hinnion '99]
- The real novelty is: $\mathsf{IZ} \approx \mathsf{HOL}^+$ (set theory $\approx$ type theory)
- We can replace $\mathsf{HOL}^+$ by the Pure Type System $\lambda Z$ [M. 2005]

# Architecture of the proof

1. **Translate sets (IZ) into pointed graphs ($\mathrm{HOL}^+$)**
   - Each variable $x$ (IZ) is turned into three variables $\alpha_x$, $A_x$, $a_x$ ($\mathrm{HOL}^+$)
   - Each formula $\phi$ (IZ) is turned into a proposition $\phi^*$ ($\mathrm{HOL}^+$)
   - **Soundness:** If $\mathrm{IZ} + \mathrm{AFA} \vdash \phi$, then $\mathrm{HOL}^+ \vdash \phi^*$
   - **Corollary:** If $\mathrm{IZ} + \mathrm{AFA} \vdash \bot$, then $\mathrm{HOL}^+ \vdash \bot$

   Therefore: $\mathrm{IZ} + \mathrm{AFA} \ \leq \ \mathrm{HOL}^+$ (relative consistency)

2. **Translate types ($\mathrm{HOL}^+$) into sets ($\mathrm{IZ}^{\mathrm{sk}}$)**
   - Each type $\tau$ ($\mathrm{HOL}^+$) is turned into a set $\tau^\dagger$ ($\mathrm{IZ}^{\mathrm{sk}}$)
   - Each term object $M : \tau$ ($\mathrm{HOL}^+$) is turned into a set $M^\dagger \in \tau^\dagger$ ($\mathrm{IZ}^{\mathrm{sk}}$)
     In particular, each proposition $A$ is turned into a subset $A^\dagger \subseteq \{\bullet\}$
   - **Soundness:** If $\mathrm{HOL}^+ \vdash A$, then $\mathrm{IZ}^{\mathrm{sk}} \vdash \bullet \in A^\dagger$
   - **Corollary:** If $\mathrm{IZ} + \mathrm{AFA} \vdash \bot$, then $\mathrm{HOL}^+ \vdash \bot$

   Therefore: $\mathrm{HOL}^+ \ \leq \ \mathrm{IZ}^{\mathrm{sk}}$ (relative consistency)

3. **Compose both translations** $\phi \mapsto \phi^*$ and $A \mapsto A^\dagger$

# The proof diagram

$$
\begin{tikzcd}
\mathrm{IZ}+\mathrm{AFA} \arrow[r, "\phi \mapsto \phi^*"] & \mathrm{HOL}^+ \arrow[r, "A \mapsto A^\dagger"] & \mathrm{IZ}^{\mathrm{sk}} \arrow[r, "\text{deskolemization}"] & \mathrm{IZ} \arrow[d, phantom, "\cap"] \\
 & & & \mathrm{IZ}+\mathrm{AFA} \arrow[ulll, bend left, "\text{reification (AFA)}"]
\end{tikzcd}
$$

From this, it follows that:

1. $\mathrm{HOL}^+$ is a conservative extension of $\mathrm{IZ}+\mathrm{AFA}$ (via the map $\phi \mapsto \phi^*$)
2. $\mathrm{IZ}$, $\mathrm{IZ}+\mathrm{AFA}$ and $\mathrm{HOL}^+$ are equiconsistent
3. $\mathrm{IZ}$, $\mathrm{IZ}+\mathrm{AFA}$ and $\mathrm{HOL}^+$ prove the same arithmetic formulas

The case of $\mathrm{IZ}+\mathrm{FA}$ is treated separately

# What about classical systems?

Using Friedman's $A$-translation (in set theory), we have:

- $\mathrm{ZF} \approx \mathrm{IZF}_C$ [Friedman '73]

With the same method, we also get:

- $\mathrm{Z} \approx \mathrm{IZ}$
- $\mathrm{Z}+\mathrm{FA} \approx \mathrm{IZ}+\mathrm{FA}$
- $\mathrm{Z}+\mathrm{AFA} \approx \mathrm{IZ}+\mathrm{AFA}$

Therefore:

**Theorem** [M. 2005, 2009]

The following theories are equiconsistent:

$$\begin{array}{ccccccccc} \mathrm{Z} & & \mathrm{Z}+\mathrm{FA} & & \mathrm{Z}+\mathrm{AFA} & & & & \\ \wr\wr & & \wr\wr & & \wr\wr & & & & \\ \mathrm{IZ} & \approx & \mathrm{IZ}+\mathrm{FA} & \approx & \mathrm{IZ}+\mathrm{AFA} & \approx & \mathrm{HOL}^+ & \approx & \lambda Z \end{array}$$

# What about replacement?

A long quest for cut elimination:

- PA $\approx$ HA $\rightsquigarrow$ System T [Gödel '58, Tait '67]
- PA2 $\approx$ HA2 $\rightsquigarrow$ System F [Girard '69]
- PA$\omega$ $\approx$ HA$\omega$ $\rightsquigarrow$ System F$\omega$ [Girard '72]
- Z $\approx$ IZ $\approx$ HOL$^+$ $\rightsquigarrow$ $\lambda$HOL$^+$ [M. 2009]
- ZF $\approx$ IZF$_C$ $\approx$ HOL$^+ + D$ $\rightsquigarrow$ $\lambda(\mathrm{HOL}^+ + D)$ [M. 2009]

where $D$ is the domination scheme:

$$(\forall x : \tau \,.\, \mathrm{mon}\,\beta \,.\, R(x, \beta)) \;\Rightarrow\; (\forall x : \tau \,.\, P(x) \Rightarrow \exists \beta \,.\, R(x, \beta)) \;\Rightarrow\; \exists \beta \,.\, \forall x : \tau \,.\, P(x) \Rightarrow R(x, \beta)$$

where $\quad \mathrm{mon}\,\beta \,.\, A(\beta) \;\equiv\; \forall \beta, \beta' \,.\, \forall f : (\beta \to \beta') \,.\, \mathrm{inj}(\beta, \beta', f) \Rightarrow A(\beta) \Rightarrow A(\beta')$

$$\lambda(\mathrm{HOL}^+ + D) \;=\; \lambda\mathrm{HOL}^+ \;+\; \text{proof term } \; \lambda\xi_1\xi_2\psi \,.\, \psi\,(\lambda\rho \,.\, \xi_2\,\rho\,(\xi_1\,\mathbf{I})) \;:\; D$$

(keeps SN)