Arnon Avron

A Logical Framework for Developing and Mechanizing Set Theories

IJCAR 2016

Why Set Theory?

- The basic notions of (naive) set theory are used in any branch and textbook of modern mathematics.
- Set theory is almost universally accepted as the foundational theory in which the whole of mathematics can (and should) be developed.

The Problems with Set Theory

- The language(s) used in official formalizations of set theories, like ZF, are rather poor and inconvenient.

The Problems with Set Theory

- The language(s) used in official formalizations of set theories, like ZF, are rather poor and inconvenient.

- *ZF* treats all sets on a par, and so hid the computational significance of many of them.

- Scientifically applicable mathematics practically deals only with a fraction of the set-theoretical "universe" of *ZF*. Therefore easier to mechanize subsystems, corresponding to subuniverses which are safer and better suited for computations, should do.

ZF

Official language: $\{=, \in\}$

Examples of problematic axioms:

Powerset:

$$\exists Z \forall x. x \in Z \leftrightarrow (\forall z. z \in x \rightarrow z \in y)$$

Replacement:

$$(\forall y \in w \exists v \forall x(\varphi \leftrightarrow x = v)) \rightarrow \exists Z \forall x. x \in Z \leftrightarrow (\exists y. y \in w \wedge \varphi)$$

Infinity:

$$\exists Z \forall x. x \in Z \leftrightarrow \forall y \in x(\forall z(z \notin y) \vee \exists w \in x \forall z. z \in y \leftrightarrow (z = w \vee z \in w))$$

ZF

Official language: $\{=, \in\}$

Examples of problematic axioms:

Powerset:

$$\exists Z \forall x. x \in Z \leftrightarrow (\forall z. z \in x \rightarrow z \in y)$$

Replacement:

$$(\forall y \in w \exists v \forall x(\varphi \leftrightarrow x = v)) \rightarrow \exists Z \forall x. x \in Z \leftrightarrow (\exists y. y \in w \wedge \varphi)$$

Infinity:

$$\exists Z \forall x. x \in Z \leftrightarrow \forall y \in x(\forall z(z \notin y) \vee \exists w \in x \forall z. z \in y \leftrightarrow (z = w \vee z \in w))$$

$$\exists Z \forall x. x \in Z \leftrightarrow \forall y \in x(y = \emptyset \vee \exists w \in x. y = w \cup \{w\})$$

The Standard Method of Extensions by Definitions

Let $\mathcal{T}$ be a theory in a language $\mathcal{L}$.

- New predicate symbols are introduced statically, as abbreviations: If $Fv(\varphi) = \{x_1, \ldots, x_n\}$ then one may add to $\mathcal{L}$ a new n-ary relation symbol P, and to $\mathcal{T}$ the axiom:

$$\forall x_1 \ldots \forall x_n . P(x_1, \ldots, x_n) \leftrightarrow \varphi$$

The Standard Method of Extensions by Definitions

Let $\mathcal{T}$ be a theory in a language $\mathcal{L}$.

- New predicate symbols are introduced statically, as abbreviations: If $Fv(\varphi) = \{x_1, \ldots, x_n\}$ then one may add to $\mathcal{L}$ a new n-ary relation symbol P, and to $\mathcal{T}$ the axiom:

$$\forall x_1 \ldots \forall x_n . P(x_1, \ldots, x_n) \leftrightarrow \varphi$$

- New constants and operation symbols are introduced dynamically: If $Fv(\varphi) = \{x_1, \ldots, x_n, y\}$ and $\mathcal{T} \vdash \forall x_1 \ldots \forall x_n \exists! y \varphi$, then one may add to $\mathcal{L}$ a new n-ary operation symbol f, and to $\mathcal{T}$ the axiom:

$$\forall x_1 \ldots \forall x_n . \varphi[f(x_1, \ldots, x_n)/y]$$

Abstraction Terms in *ZF*

Texts about sets make extensive use of terms of the form $\{x \mid \varphi\}$.

There are two known ways of using such abstraction terms in *ZF*:

- For convenience, as Class terms (that can be eliminated).
- As sugar for new operation symbols (including constants), in extensions by definitions of the basic system:

  If $\vdash \exists Z \forall x (x \in Z \leftrightarrow \varphi)$

  Then add as a new axiom: $\forall x (x \in \{x \mid \varphi\} \leftrightarrow \varphi)$.

Our Goal

To present a natural unified framework for formalizing axiomatic set theories of different strength, from rudimentary set theory to full *ZF* (and beyond), with the following properties:

1. It uses only static languages.
2. It reflects real mathematical practice in allowing the use of abstraction terms for sets.
3. It is suitable for making computations with sets.

The Ideal Language for Sets

Terms:

- Every variable is a term.
- $\{x \mid \varphi\}$ is a term if φ is a formula and $x \in Fv(\varphi)$.

Formulas:

- If t and s are terms then $t = s$ and $t \in s$ are formulas.
- If φ and ψ are formulas, and x is a variable, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $\forall x\varphi$, $\exists x\varphi$ are formulas.

The Ideal Calculus for Sets

Extensionality:

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y$$

$$(t = \{x \mid x \in t\})$$

The Comprehension Schema:

$$\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi)$$

$$(t \in \{x \mid \varphi\} \leftrightarrow \varphi[t/x])$$

The Regularity Schema ($\in$-induction):

$$(\forall x(\forall y(y \in x \rightarrow \varphi[y/x]) \rightarrow \varphi)) \rightarrow \forall x \varphi$$

The Ideal Calculus for Sets

Extensionality:

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y$$

$$(t = \{x \mid x \in t\})$$

The Comprehension Schema:

$$\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi)$$

$$(t \in \{x \mid \varphi\} \leftrightarrow \varphi[t/x])$$

The Regularity Schema ($\in$-induction):

$$(\forall x(\forall y(y \in x \rightarrow \varphi[y/x]) \rightarrow \varphi)) \rightarrow \forall x\varphi$$

Ideal, but inconsistent!

The Computational Perspective

$$(\alpha) \quad \lambda x.t = \lambda y.t[y/x] \qquad \{x \mid \varphi\} = \{y \mid \varphi[y/x]\}$$

$$(\beta) \quad (\lambda x.t)s = t[s/x] \qquad s \in \{x \mid \varphi\} \leftrightarrow \varphi[s/x]$$

$$(\eta) \quad (\lambda x.tx) = t \qquad t = \{x \mid x \in t\}$$

Conditions:

- (α) y is free for x in φ (in t).
- (β) s is free for x in φ (in t).
- (η) x is not free in t.

Giving up a Part of Our Ideal

Extensionality:

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y$$

The Comprehension Schema:

$$\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi), \text{if } \varphi \text{ is safe w.r.t. } \{x\}.$$

The Regularity Schema ($\in$-induction):

$$(\forall x(\forall y(y \in x \rightarrow \varphi[y/x]) \rightarrow \varphi)) \rightarrow \forall x \varphi$$

Our Demands Concerning Safety

- Safety is a relation between formulas and sets of variables.
- Differences between set theories should mainly be due to differences in their notions of safety.
- The safety relation underlying a set theory should be decidable, and defined syntactically, in a static way.

Our Demands Concerning Safety

- Safety is a relation between formulas and sets of variables.
- Differences between set theories should mainly be due to differences in their notions of safety.
- The safety relation underlying a set theory should be decidable, and defined syntactically, in a static way.

THE PROBLEM: What should be the syntactic properties of safety relations that would make it possible to develop an adequate framework for set theories, in which these demands are met?

Philosophical Ideas from Set Theory Itself

Definiteness

According to Zermelo's formulation of the **separation** axiom, the formula $x \in A \wedge \psi$ is safe, provided ψ defines a definite property.

In ZF, every formula is "definite".

Philosophical Ideas from Set Theory Itself

Definiteness

According to Zermelo's formulation of the **separation** axiom, the formula $x \in A \wedge \psi$ is safe, provided ψ defines a definite property.

In ZF, every formula is "definite".

Predicativity

- A set is the extension of a property which is defined by some predicative formula.
- A formula is predicative if the collection it defines is universe independent.

Technical Ideas from Set Theory Itself

Absoluteness of Formulas

A formula in the language of set theory is absolute if the truth value it gets by an assignment v in a universe M depends only on v, but not on M; that is: if its truth value is universe independent.

From a predicativist point of view, definite=absolute.

Technical Ideas from Set Theory Itself

Absoluteness of Formulas

A formula in the language of set theory is absolute if the truth value it gets by an assignment v in a universe M depends only on v, but not on M; that is: if its truth value is universe independent.

From a predicativist point of view, definite=absolute.

Constructible Sets

- Union, intersection: constructible
- Powerset: not constructible

"Universe Independence" of Collections

Intuitively, a collection $\{x \mid \varphi\}$ is universe independent if

$$\{x \in S_1 \mid S_1 \models \varphi\} = \{x \in S_2 \mid S_2 \models \varphi\}$$

whenever S_1 and S_2 are "Universes" of sets that include all the values that are assigned to the parameters of φ.

Examples: If S_1 and S_2 are "universes", and $a \in S_1 \cap S_2$, then:

$$\{x \in S_1 \mid x \in \cup a\} = \{x \in S_2 \mid x \in \cup a\}$$

but in general:

$$\{x \in S_1 \mid x \in P(a)\} \neq \{x \in S_2 \mid x \in P(a)\}$$

A Syntactic Approximation of Absoluteness: Δ_0

- Every atomic formula is in Δ_0.
- If φ and ψ are in Δ_0, then so are $\neg\varphi$, $\varphi \vee \psi$, and $\varphi \wedge \psi$.
- If x and y are two different variables, and φ is in Δ_0, then so are $\exists x \in y\varphi$ and $\forall x \in y\varphi$.

A very similar syntactic approximation of decidability of formulas is used in computability theory and formal arithmetics.

Ideas from Database Theory

- To provide an answer to a query in a relational database, a computation should be made in which:
  - The input is a finite set of finite sets of tuples.
  - The output should also be a finite set of tuples.

  In other words: the computation is done with (finite) sets.

Ideas from Database Theory

- To provide an answer to a query in a relational database, a computation should be made in which:
  - The input is a finite set of finite sets of tuples.
  - The output should also be a finite set of tuples.

  In other words: the computation is done with (finite) sets.
- In order for such a computation to be possible, only safe formulas should be used for queries.

Ideas from Database Theory

- To provide an answer to a query in a relational database, a computation should be made in which:
  - The input is a finite set of finite sets of tuples.
  - The output should also be a finite set of tuples.

  In other words: the computation is done with (finite) sets.
- In order for such a computation to be possible, only safe formulas should be used for queries.
- A formula φ such that $Fv(\varphi) = \{x_1, \ldots, x_n\}$ is taken to be safe if the identity of $\{\langle x_1, \ldots, x_n \rangle \mid \varphi\}$ is domain independent (d.i.) (given an interpretation of the basic relations — the "tables" of the database).

Syntactic Safety: the Class $SS(\overrightarrow{P})$

- $P_i(t_1, \ldots, t_{n_i}) \in SS(\overrightarrow{P})$ in case P_i is in $\overrightarrow{P}$.
- $x = c$ and $c = x$ are in $SS(\overrightarrow{P})$.
- $\varphi \vee \psi \in SS(\overrightarrow{P})$ if $\varphi \in SS(\overrightarrow{P})$, $\psi \in SS(\overrightarrow{P})$, and the two formulas have the same free variables.
- $\exists x\varphi \in SS(\overrightarrow{P})$ if $\varphi \in SS(\overrightarrow{P})$.
- If $\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_k$, then $\varphi \in SS(\overrightarrow{P})$ if the following conditions are met:
  1. For each i, either φ_i is atomic, or φ_i is in $SS(\overrightarrow{P})$, or φ_i is a negation of a formula of either type.
  2. Every free variable x of φ is limited in φ.

Generalizing Domain Independence and Absoluteness

Let $Fv(\varphi) = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$

- φ is d.i. (domain-independent) for S_1 and S_2 with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$) if for all $a_1 \ldots, a_m \in S_1 \cap S_2$:

$$\{\overrightarrow{x} \in S_2^n \mid S_2 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\} = \{\overrightarrow{x} \in S_1^n \mid S_1 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\}$$

Generalizing Domain Independence and Absoluteness

Let $Fv(\varphi) = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$

- φ is d.i. (domain-independent) for S_1 and S_2 with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$) if for all $a_1 \ldots, a_m \in S_1 \cap S_2$:

$$\{\overrightarrow{x} \in S_2^n \mid S_2 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\} = \{\overrightarrow{x} \in S_1^n \mid S_1 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\}$$

- φ is d.i. with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ \{x_1, \ldots, x_n\}$) if $\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$ for every admissible domains S_1 and S_2.

Generalizing Domain Independence and Absoluteness

Let $Fv(\varphi) = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$

- φ is d.i. (domain-independent) for S_1 and S_2 with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$) if for all $a_1 \ldots, a_m \in S_1 \cap S_2$:

$$\{\overrightarrow{x} \in S_2^n \mid S_2 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\} = \{\overrightarrow{x} \in S_1^n \mid S_1 \models \varphi(\overrightarrow{x}, \overrightarrow{a})\}$$

- φ is d.i. with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ \{x_1, \ldots, x_n\}$) if $\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$ for every admissible domains S_1 and S_2.

- φ is d.i. if $\varphi \succ Fv(\varphi)$. φ is absolute if $\varphi \succ \emptyset$.

Generalizing Domain Independence and Absoluteness

Let $Fv(\varphi) = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$

- φ is d.i. (domain-independent) for S_1 and S_2 with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$) if for all $a_1 \ldots, a_m \in S_1 \cap S_2$:

$$\{\vec{x} \in S_2^n \mid S_2 \models \varphi(\vec{x}, \vec{a})\} = \{\vec{x} \in S_1^n \mid S_1 \models \varphi(\vec{x}, \vec{a})\}$$

- φ is d.i. with respect to $\{x_1, \ldots, x_n\}$ ($\varphi \succ \{x_1, \ldots, x_n\}$) if $\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$ for every admissible domains S_1 and S_2.

- φ is d.i. if $\varphi \succ Fv(\varphi)$. φ is absolute if $\varphi \succ \emptyset$.

Obviously, if $\varphi \succ^{S_1;S_2} X$, and $Y \subseteq X$, then $\varphi \succ^{S_1;S_2} Y$. Similarly, if $\varphi \succ X$ and $Y \subseteq X$, then $\varphi \succ Y$.

Principles and Examples

- Languages (and their intended models) should be designed so that every atomic formula is absolute.
- For the equality relation we have:
  - $x = y \succ \{x\}$, $x = y \succ \{y\}$ (and $x = y \succ \emptyset$). However,
  - $x = y \not\succ \{x, y\}$.
  - $x \neq x \succ \{x\}$.
- In databases, $P(x_1, \ldots, x_n) \succ \{x_1, \ldots, x_n\}$ whenever P represents a table of the database.
- In set theories (assuming that universes are transitive):
  - $x \in y \succ \{x\}$.
  - $x \in y \not\succ \{y\}$.

Safety Relations

Let $\mathcal{L}$ be a (first-order) language. A relation $\succ$ between formulas φ of $\mathcal{L}$ and subsets of $Fv(\varphi)$ is a safety relation for $\mathcal{L}$ if it has the following properties of d.i.:

- If $\varphi \succ X$ and $Y \subseteq X$, then $\varphi \succ Y$.
- $\varphi \succ \emptyset$ if φ is atomic.
- $\varphi \succ \{x\}$ if $\varphi \in \{x \neq x, x = t, t = x\}$, and $x \notin Fv(t)$.
- $\neg\varphi \succ \emptyset$ if $\varphi \succ \emptyset$.
- $\varphi \vee \psi \succ X$ if $\varphi \succ X$ and $\psi \succ X$.
- $\varphi \wedge \psi \succ X \cup Y$ if $\varphi \succ X$, $\psi \succ Y$ and $Y \cap Fv(\varphi) = \emptyset$.
- $\exists y\varphi \succ X - \{y\}$ if $y \in X$ and $\varphi \succ X$.
- If $\varphi \succ \{x_1, ..., x_n\}$ and $\psi \succ \emptyset$, then $\forall x_1, ..., x_n(\varphi \rightarrow \psi) \succ \emptyset$.

Explaining the Condition for $\wedge$

Let $\theta(x, y, z) = \varphi(x, z) \wedge \psi(x, y, z)$, $\varphi \succ \{x\}$, $\psi \succ \{y\}$. Define:

- $Z(s) = \{x \mid \varphi(x, s)\}$
- $W(s, d) = \{y \mid \psi(d, y, s)\}$

Then for every object s:

$$\{\langle x, y\rangle \mid \theta(x, y, s)\} = \bigcup_{d \in Z(s)} \{d\} \times W(s, d)$$

Since $\varphi \succ \{x\}$, then $Z(s)$ is d.i. for every s.
Since $\psi \succ \{y\}$, then $W(s, d)$ is d.i. for every s and d.

Hence $\{\langle x, y\rangle \mid \theta(x, y, s)\}$ is d.i. for every s.

Completeness of the Syntactic Characterization (I)

Let σ be an ordinary first-order signature with equality.

- A d.i.-function for σ is a function which assigns to every n-ary predicate symbol from σ a set of subsets of $\{1, \ldots, n\}$.
- Let F be a d.i.-function, and let S_1 and S_2 be structures for σ. S_1 and S_2 are $F-$compatible if:
  - $p(x_1, \ldots, x_n) \succ^{S_1;S_2} \{x_{i_1}, \ldots, x_{i_k}\}$ in case p is n-ary, $x_1, \ldots, x_n$ are distinct, and $\{i_1, \ldots, i_k\} \in F(p)$.
  - $y = f(x_1, \ldots, x_n) \succ^{S_1;S_2} \{y\}$ in case f is n-ary and $y, x_1, \ldots, x_n$ are distinct.
- A formula φ of σ is called $F-$d.i. w.r.t. X ($\varphi \succ_F X$) if $\varphi \succ^{S_1;S_2} X$ whenever S_1 and S_2 are $F-$compatible.

Completeness of the Syntactic Characterization (II)

Theorem: Let σ be a first-order signature with equality, and let F be a d.i.-function for σ.
$\succ_F$ is the minimal safety relation for $\mathcal{L}(\sigma)$ which satisfies the following conditions:

- $p(x_1, \ldots, x_n) \succ \{x_{i_1}, \ldots, x_{i_k}\}$ in case p is n-ary, $x_1, \ldots, x_n$ are distinct, and $\{i_1, \ldots, i_k\} \subseteq X$ for some $X \in F(p)$.
- If $\varphi \succ X$, ψ is classically equivalent to φ, and $Fv(\psi) = Fv(\varphi)$, then $\psi \succ X$.

Our Basic Framework: Syntax

Terms:

- Every variable is a term.
- $\{x \mid \varphi\}$ is a term if φ is a formula and $\varphi \succ \{x\}$.

Formulas:

- If t and s are terms then $t = s$ and $t \in s$ are atomic formulas.
- If φ and ψ are formulas, and x is a variable, then $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\exists x\varphi$ ($\varphi \to \psi$, $\forall x\varphi$,...) are formulas.

The relation $\succ$:

- $\succ$ should be a decidable safety relation.
- $x \in t \succ \{x\}$ if t is a term and $x \notin Fv(t)$.

Our Basic Framework: Basic Axioms

Extensionality:

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y$$

$$(t = \{x \mid x \in t\})$$

The Comprehension Schema:

$$\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi)$$

$$(t \in \{x \mid \varphi\} \leftrightarrow \varphi[t/x])$$

The Regularity Schema ($\in$-induction):

$$(\forall x(\forall y(y \in x \rightarrow \varphi[y/x]) \rightarrow \varphi)) \rightarrow \forall x\varphi$$

Introducing New Symbols

Operations: Let t be a term. One may introduce a new n-ary operation symbol F_φ together with the axiom:

$$F_\varphi(x_1, \ldots, x_n) = t$$

Instead of a new symbol, one may use $\lambda x_1, \ldots, x_n.t$.

Introducing New Symbols

Operations: Let t be a term. One may introduce a new n-ary operation symbol F_φ together with the axiom:

$$F_\varphi(x_1, \ldots, x_n) = t$$

Instead of a new symbol, one may use $\lambda x_1, \ldots, x_n.t$.

Predicates: Suppose $Fv(\varphi) \subseteq \{x_1, \ldots, x_n\}$ and $\varphi \succ \emptyset$. Then one may introduce a new n-ary predicate symbol P_φ together with the axiom:

$$P_\varphi(x_1, \ldots, x_n) \leftrightarrow \varphi$$

Introducing New Symbols

Operations: Let t be a term. One may introduce a new n-ary operation symbol F_φ together with the axiom:

$$F_\varphi(x_1, \ldots, x_n) = t$$

Instead of a new symbol, one may use $\lambda x_1, \ldots, x_n.t$.

Predicates: Suppose $Fv(\varphi) \subseteq \{x_1, \ldots, x_n\}$ and $\varphi \succ \emptyset$. Then one may introduce a new n-ary predicate symbol P_φ together with the axiom:

$$P_\varphi(x_1, \ldots, x_n) \leftrightarrow \varphi$$

In the case of unary predicates it is often more convenient to write $t \in \{x|\varphi\}$ instead of $P_\varphi(t)$. The expression $\{x|\varphi\}$ (where $\varphi \succ \emptyset$) is called a class term, and it may contain parameters. (Note that a "class term" is not a term of the language!)

Rudimentary Set Theory

We denote by $\succ_{RST}$ the minimal safety relation $\succ$ allowed in our framework. It can inductively be defined as follows:

- $\varphi \succ_{RST} \emptyset$ if φ is atomic.
- $\varphi \succ_{RST} \{x\}$ if $\varphi \in \{x = t, t = x, x \neq x, x \in t\}$, and $x \notin Fv(t)$.
- $\neg\varphi \succ_{RST} \emptyset$ if $\varphi \succ_{RST} \emptyset$.
- $\varphi \vee \psi \succ_{RST} X$ if $\varphi \succ_{RST} X$ and $\psi \succ_{RST} X$.
- $\varphi \wedge \psi \succ_{RST} X \cup Y$ if $\varphi \succ_{RST} X$, $\psi \succ_{RST} Y$ and $Y \cap Fv(\varphi) = \emptyset$.
- $\exists y\varphi \succ_{RST} X - \{y\}$ if $y \in X$ and $\varphi \succ_{RST} X$.

RST (Rudimentary Set Theory) is the set theory which is induced in our framework by $\succ_{RST}$.

The Power of *RST*

- $s \subseteq t \leftrightarrow_{Df} \forall x(x \in s \rightarrow x \in t)$

  (because $\forall x(\varphi \rightarrow \psi) =_{Df} \neg\exists x(\varphi \wedge \neg\psi)$)

The Power of *RST*

- $s \subseteq t \leftrightarrow_{Df} \forall x(x \in s \rightarrow x \in t)$

  (because $\forall x(\varphi \rightarrow \psi) =_{Df} \neg\exists x(\varphi \wedge \neg\psi)$)

- $\emptyset =_{Df} \{x \mid x \neq x\}$.
- $\{t_1, \ldots, t_n\} =_{Df} \{x \mid x = t_1 \vee \ldots \vee x = t_n\}$
- $\langle t, s\rangle =_{Df} \{\{t\}, \{t, s\}\}$.
- $\{x \in t \mid \varphi\} =_{Df} \{x \mid x \in t \wedge \varphi\}$, provided $\varphi \succ \emptyset$.
- $\{t(x) \mid x \in s\} =_{Df} \{y \mid \exists x.x \in s \wedge y = t\}$

The Power of *RST*

- $s \subseteq t \leftrightarrow_{Df} \forall x(x \in s \rightarrow x \in t)$

  (because $\forall x(\varphi \rightarrow \psi) =_{Df} \neg\exists x(\varphi \wedge \neg\psi)$)

- $\emptyset =_{Df} \{x \mid x \neq x\}$.
- $\{t_1, \ldots, t_n\} =_{Df} \{x \mid x = t_1 \vee \ldots \vee x = t_n\}$
- $\langle t, s\rangle =_{Df} \{\{t\}, \{t, s\}\}$.
- $\{x \in t \mid \varphi\} =_{Df} \{x \mid x \in t \wedge \varphi\}$, provided $\varphi \succ \emptyset$.
- $\{t(x) \mid x \in s\} =_{Df} \{y \mid \exists x.x \in s \wedge y = t\}$

- $\bigcup t =_{Df} \{x \mid \exists y.y \in t \wedge x \in y\}$
- $s \times t =_{Df} \{x \mid \exists a \exists b.a \in s \wedge b \in t \wedge x = \langle a, b\rangle\}$

In general, an operation is definable in *RST* iff it is rudimentary.

An Explanation of $\times$

$a \in s \succ \{a\}$ (if $a \notin Fv(s)$)

$b \in t \succ \{b\}$ (if $b \notin Fv(t)$)

Since $b \notin Fv(a \in s)$, we get: $a \in s \wedge b \in t \succ \{a, b\}$

$x = \langle a, b \rangle \succ \{x\}$ (if $x \neq a, x \neq b$)

Since $x \notin Fv(a \in s \wedge b \in t)$, we get:

$a \in s \wedge b \in t \wedge x = \langle a, b \rangle \succ \{a, b, x\}$, and so:

$$\exists a \exists b. a \in s \wedge b \in t \wedge x = \langle a, b \rangle \succ \{x\}$$

The Power of *RST* - Continued

- $\iota x \varphi =_{Df} \bigcup \{x \mid \varphi\}$ (provided $\varphi \succ \{x\}$).
- $\lambda x \in s.t =_{Df} \{\langle x, t \rangle \mid x \in s\}$
- $f(x) =_{Df} \iota y. \exists z \exists v (z \in f \wedge v \in z \wedge y \in v \wedge z = \langle x, y \rangle)$

$$\vdash_{RST} a \in s \rightarrow (\lambda x \in s.t)(a) = t\{a/x\}$$

The Power of *RST* - Continued

- $\iota x \varphi =_{Df} \bigcup \{x \mid \varphi\}$ (provided $\varphi \succ \{x\}$).
- $\lambda x \in s.t =_{Df} \{\langle x, t\rangle \mid x \in s\}$
- $f(x) =_{Df} \iota y.\exists z \exists v(z \in f \wedge v \in z \wedge y \in v \wedge z = \langle x, y\rangle)$

$$\vdash_{RST} a \in s \rightarrow (\lambda x \in s.t)(a) = t\{a/x\}$$

- $V =_{Df} \{x|x = x\}$
- $P(y) =_{Df} \{x|x \subseteq y\}$
- $\omega =_{Df} \{x|(x = \emptyset \vee \exists w \in x.x = w \cup \{w\}) \wedge$
  $\wedge \forall y \in x(y = \emptyset \vee \exists w \in x.y = w \cup \{w\})\}$
- $HF = J_1$ is the minimal model of *RST*.

Handling The Impredicative Comprehension Axioms

Each of the impredicative comprehension axioms of ZF can be captured (in a modular way) by adding to the definition of the safety relation $\succ$ a corresponding syntactic condition:

Separation: $\varphi \succ \emptyset$ for every formula φ.

Powerset: $x \subseteq t \succ \{x\}$ if $x \notin Fv(t)$.

(Here $\subseteq$ should better be taken as a new primitive.)

Replacement: $\exists y \varphi \wedge \forall y(\varphi \to \psi) \succ X$

provided $\psi \succ X$, and $X \cap Fv(\varphi) = \emptyset$.

An Important Property of Replacement

Let $\mathcal{T}$ be a set theory in our framework whose safety relation $\succ_{\mathcal{T}}$ satisfies the condition for replacement. Then for every formula φ, if $Fv(\varphi) = \{y_1, \ldots, y_n, x\}$ then there exists a term t_φ such that:

1. $Fv(t_\varphi) = \{y_1, \ldots, y_n\}$
2. $\vdash_{\mathcal{T}} \forall y_1, \ldots, y_n \exists! x \varphi \rightarrow \forall y_1, \ldots, y_n (\varphi\{t_\varphi / x\})$

Handling Infinity by Adding a Constant

- Include in the language a new constant HF (interpreted as the collection of hereditarily finite sets.)
- Include in the set of axioms the following counterparts of Peano's axioms:
  1. $\emptyset \in HF$
  2. $\forall x \forall y . x \in HF \wedge y \in HF \rightarrow x \cup \{y\} \in HF$
  3. $\varphi(0) \wedge (\forall x \forall y . \varphi(x) \wedge \varphi(y) \rightarrow \varphi(x \cup \{y\}) \rightarrow \forall x \in HF . \varphi(x)$

$RST\omega$ is the theory which is obtained from RST in this way.

$RST\omega$ and J_2

- The minimal model of $RST\omega$ is J_2.

$RST\omega$ and J_2

- The minimal model of $RST\omega$ is J_2.
- Each $a \in J_2$ is defined by some closed term of $RST\omega$.
- It can be shown that J_2 (as a universe) and $RST\omega$ (as a theory) suffice for great parts (most of?) scientifically applicable mathematics.

$RST\omega$ and J_2

- The minimal model of $RST\omega$ is J_2.
- Each $a \in J_2$ is defined by some closed term of $RST\omega$.
- It can be shown that J_2 (as a universe) and $RST\omega$ (as a theory) suffice for great parts (most of?) scientifically applicable mathematics.

  However, this involves a lot of coding, as well as treating the collection of real numbers as a proper class.

A Better Solution: Using Ancestral Logic (**AL**)

Languages in **AL** are defined like first-order languages with equality, but with the following additional clause:

- If φ is a formula, x, y are distinct variables which are free in φ, and s, t are terms, then $(TC_{x,y}\varphi)(s, t)$ is a formula.

A Better Solution: Using Ancestral Logic (**AL**)

Languages in **AL** are defined like first-order languages with equality, but with the following additional clause:

- If φ is a formula, x, y are distinct variables which are free in φ, and s, t are terms, then $(TC_{x,y}\varphi)(s,t)$ is a formula.

The intended meaning of $(TC_{x,y}\varphi)(x,y)$ is:

$$\begin{aligned} & \varphi(x,y) \\ \vee & \exists w_1.\varphi(x,w_1) \wedge \varphi(w_1,y) \\ \vee & \exists w_1 \exists w_2.\varphi(x,w_1) \wedge \varphi(w_1,w_2) \wedge \varphi(w_2,y) \\ \vee & \ldots \end{aligned}$$

The meaning of $(TC_{x,y}\varphi)(s,t)$ is the same as that of $\exists u \exists v.u = s \wedge v = t \wedge (TC_{u,v}\varphi)(u,v)$, where u and v are fresh.

A Better Solution: Using Ancestral Logic (**AL**)

Languages in **AL** are defined like first-order languages with equality, but with the following additional clause:

- If φ is a formula, x, y are distinct variables which are free in φ, and s, t are terms, then $(TC_{x,y}\varphi)(s,t)$ is a formula.

The intended meaning of $(TC_{x,y}\varphi)(x,y)$ is:

$$\begin{aligned} & \varphi(x,y) \\ \vee & \exists w_1.\varphi(x,w_1) \wedge \varphi(w_1,y) \\ \vee & \exists w_1 \exists w_2.\varphi(x,w_1) \wedge \varphi(w_1,w_2) \wedge \varphi(w_2,y) \\ \vee & \ldots \end{aligned}$$

The meaning of $(TC_{x,y}\varphi)(s,t)$ is the same as that of $\exists u \exists v.u = s \wedge v = t \wedge (TC_{u,v}\varphi)(u,v)$, where u and v are fresh.

Unlike **SOL**, **AL** involves no new ontological commitments.

Example of the Naturalness of **AL**

Let V_0 be the closure of $\{0\}$ under pairing. Then a subset S of V_0 is r.e. iff it is definable by a formula of the language $\mathcal{PTC}^+$, where the latter has variables, 0; $\langle\ ,\ \rangle$; $=$; $\vee$; $\wedge$; and TC.

Example of the Naturalness of **AL**

Let V_0 be the closure of $\{0\}$ under pairing. Then a subset S of V_0 is r.e. iff it is definable by a formula of the language $\mathcal{PTC}^+$, where the latter has variables, 0; $\langle\ ,\ \rangle$; $=$; $\vee$; $\wedge$; and TC.

Terms of $\mathcal{PTC}^+$

1. The constant 0 is a term.
2. Every variable is a term.
3. If t and s are terms then so is $\langle t, s\rangle$.

Formulas of $\mathcal{PTC}^+$

1. If t and s are terms then $t = s$ is a formula.
2. If φ and ψ are formulas, x, y are distinct variables, and s, t are terms, then $\varphi \vee \psi$, $\varphi \wedge \psi$, and $(TC_{x,y}\varphi)(s, t)$ are formulas.

Logical Proof Systems

- **AL** has no sound and complete finitary proof system.
- **AL** does have a natural sound finitary proof systems. The easiest to mechanize are obtained by adding to some sequent calculi for first order logic with equality a few natural rules for TC. One of them is the following induction rule:

$$\frac{\Gamma, \psi(x), \varphi(x,y) \Rightarrow \psi(y), \Delta}{\Gamma, \psi(s), (TC_{x,y}\varphi)(s,t) \Rightarrow \psi(t), \Delta}$$

(provided x and y are not free in Γ, Δ, and y is not free in ψ)

- The Gentzen-type system for classical **AL** is complete for an appropriate Henkin-type semantics.

Using **AL** in Our Framework

Safety Relations in **AL** are defined like in the case of **FOL**, but with the following additional condition (that respects d.i.):

$$TC_{x,y}\varphi \succ X \quad \text{if} \quad \varphi \succ X \quad \text{and} \quad \{x, y\} \cap X \neq \emptyset$$

$\succ_{PZF}$ is the minimal safety relation in **AL**.

PZF (Predicative Set Theory) is the set theory induced by $\succ_{PZF}$.

Using **AL** in Our Framework

Safety Relations in **AL** are defined like in the case of **FOL**, but with the following additional condition (that respects d.i.):

$$TC_{x,y}\varphi \succ X \quad \text{if} \quad \varphi \succ X \quad \text{and} \quad \{x, y\} \cap X \neq \emptyset$$

$\succ_{PZF}$ is the minimal safety relation in **AL**.

PZF (Predicative Set Theory) is the set theory induced by $\succ_{PZF}$.

Expressive Power:

- All finitary inductive definitions are available in *PZF*.
- $\omega =_{Df} \{x \mid x = \emptyset \vee \exists y.y = \emptyset \wedge (TC_{x,y}x = y \cup \{y\})(x, y)\}$
- $TH(x) =_{Df} x \cup \{y \mid (TC_{x,y}y \in x)(x, y)\}$
- $HF =_{Df} \{x \mid \exists y \exists z.x \in y \wedge z = \{\emptyset\} \wedge$
  $\wedge (TC_{z,y}\exists u \in z \exists v \in z.y = z \cup \{u \cup \{v\}\})(z, y)\}$

Properties of *PZF*

- The minimal model of *PZF* is $J_{\omega^\omega} = L_{\omega^\omega}$.

Properties of *PZF*

- The minimal model of *PZF* is $J_{\omega^\omega} = L_{\omega^\omega}$.
- If t is a closed term of *PZF* then t defines an element of J_{ω^ω}. Conversely, every element of J_{ω^ω} is defined by some closed term of *PZF*.

  In particular, J_2, J_3, ... , J_ω, J_{ω^2}, J_{ω^3}, ... are defined by terms of *PZF*.

Properties of *PZF*

- The minimal model of *PZF* is $J_{\omega^\omega} = L_{\omega^\omega}$.
- If t is a closed term of *PZF* then t defines an element of J_{ω^ω}. Conversely, every element of J_{ω^ω} is defined by some closed term of *PZF*.

  In particular, J_2, J_3, ... , J_ω, J_{ω^2}, J_{ω^3}, ... are defined by terms of *PZF*.

- J_{ω^ω} (as a universe) and *PZF* (as a theory) suffice for (most of?) scientifically applicable mathematics. This involves no coding, and the collection of real numbers can be taken as a set (e.g. as an element of J_{ω^2}). (This set does not include, of course, "all" the real numbers.)

Computational Theories and Structures

We call a set theory T computational if:

1. The set of closed terms of T determines a transitive set, which is the minimal model of T.
2. If t is a term of T, and $Fv(t) = \{y_1, \ldots, y_n\}$, then:

$$\forall y_1 \ldots \forall y_n . y_1 \in \mathcal{M} \wedge \ldots \wedge y_n \in \mathcal{M} \rightarrow t_{\mathcal{M}} = t$$

RST, *RST*ω, and *PZF* are all computational.

The Axiom of Choice

The most natural way to incorporate the axiom of choice into our framework is by further extending the set of terms, using Hilbert's ε symbol, together with its usual characterizing axiom (which is equivalent to the axiom of global choice):

$$\exists x \varphi \to \varphi\{\varepsilon x \varphi / x\}$$