Simple type theory

I. What we have seen so far

Basic notions: proof, theory, model

Examples of theories: Arithmetic
All mathematical statements: naive set theory, set theory

In naive set theory

- every predicate is an object
- every predicate can be applied to every object

Abandon the first principle: set theory no full comprehension scheme, but always $t \in u$
Abandon the second: simple type theory

II. Simple type theory

A many-sorted theory

Objects classified according to their degree of functionality: base objects, propositional contents, functions mapping base objects to base objects, functions mapping functions to functions, etc.

An infinite number of sorts: simple types

- ι and o are simple types,
- if A and B are simple types, then $A \rightarrow B$ is a simple type.

ι: base objects, o propositional contents, $A \rightarrow B$: functions from A to B

The language of Simple type theory

For each pair A, B, a $\alpha_{A,B}$ of arity $\langle A \to B, A, B\rangle$
A predicate symbol ε of arity $\langle o \rangle$
For each t, $x_1, ..., x_n \mapsto t$, for each A, $\{x_1, ..., x_n \mid A\}$

But sufficient to take
$K_{A,B} = x, y \mapsto x$
$S_{A,B,C} = x, y, z \mapsto (x\ z\ (y\ z))$
$\dot{\wedge} = \{x, y \mid \varepsilon(x) \wedge \varepsilon(y)\}$
$\dot{\forall}_A = \{x \mid \forall y\ \varepsilon(x\ y)\}$, etc.

$$K_{A,B} : A \to B \to A$$
$$S_{A,B,C} : (A \to B \to C) \to (A \to B) \to A \to C$$
$$\dot{\top} : o$$
$$\dot{\bot} : o$$
$$\dot{\wedge} : o \to o \to o$$
$$\dot{\vee} : o \to o \to o$$
$$\dot{\Rightarrow} : o \to o \to o$$
$$\dot{\forall}_A : (A \to o) \to o$$
$$\dot{\exists}_A : (A \to o) \to o$$

A purely computational theory

$$(K_{A,B}\ x\ y) \longrightarrow x$$

$$(S_{A,B,C}\ x\ y\ z) \longrightarrow (x\ z\ (y\ z))$$

$$\varepsilon(\dot{\top}) \longrightarrow \top \qquad\qquad \varepsilon(\dot{\bot}) \longrightarrow \bot$$

$$\varepsilon(\dot{\wedge}\ x\ y) \longrightarrow (\varepsilon(x) \wedge \varepsilon(y))$$

$$\varepsilon(\dot{\vee}\ x\ y) \longrightarrow (\varepsilon(x) \vee \varepsilon(y))$$

$$\varepsilon(\dot{\Rightarrow}\ x\ y) \longrightarrow (\varepsilon(x) \Rightarrow \varepsilon(y))$$

$$\varepsilon(\dot{\forall}_A\ x) \longrightarrow \forall y\ \varepsilon(x\ y)$$

$$\varepsilon(\dot{\exists}_A\ x) \longrightarrow \exists y\ \varepsilon(x\ y)$$

Comprehension

Term t, there exists u such that

$$(u\ y) \longrightarrow^* t$$

$u = \overline{\lambda} y\ t$

Proposition A, there exists u such that

$$\varepsilon(u) \longrightarrow^* A$$

Impredicativity

Propositional content of the proposition

$$\forall p\ (\varepsilon(p) \Rightarrow \varepsilon(p))$$

$$\dot{\forall}\overline{\lambda} p\ (p \mathbin{\dot{\Rightarrow}} p)$$

A propositional content built by quantifying over all propositional contents: impredicativity

In Arithmetic symbols $f_{x_1,\ldots,x_n,y,A}$ restricted to A with no ϵ: predicative

If we drop this restriction: impredicative Arithmetic

As all propositions have a propositional content
The proposition

$$\forall p \; (\varepsilon(p) \Rightarrow \varepsilon(p))$$

expresses that all propositions imply themselves
In particular itself
Partial auto-reference due to impredicativity

III. Models of Simple type theory

A model valued in $\{0, 1\}$

$\mathcal{M}_\iota$ any non empty set (e.g. $\{7\}$)
$\mathcal{M}_o = \{0, 1\}$
$\mathcal{M}_{A \to B}$ set of all functions from $\mathcal{M}_A$ to $\mathcal{M}_B$
$\hat{K}_{A,B}$ function mapping a and b to a
$\hat{S}_{A,B,C}$ function mapping a, b, and c to $(a\ c\ (b\ c))$
$\hat{\dot{\top}} = \tilde{\top} = 1$, $\hat{\dot{\bot}} = \tilde{\bot} = 0$
$\hat{\dot{\wedge}} = \tilde{\wedge}$, $\hat{\dot{\vee}} = \tilde{\vee}$, $\hat{\dot{\Rightarrow}} = \tilde{\Rightarrow}$

$\hat{\dot{\forall}}_A$ function mapping f to $\tilde{\forall}\ \{f(x) \mid x \in \mathcal{M}_A\}$
$\hat{\dot{\exists}}_A$ function mapping f to $\tilde{\exists}\ \{f(x) \mid x \in \mathcal{M}_A\}$
$\hat{\alpha}_{A,B}$ function mapping f and a to $f(a)$
$\hat{\varepsilon}$ identity

All reduction rules of Simple type theory are valid in this model

Super-consistency

Same construction, except $\mathcal{M}_o = \mathcal{B}$

IV. Elements of mathematics

Equality

Like in Arithmetic

A constant $=$ of type $\iota \to \iota \to o$ and the rule

$$x = y \longrightarrow (\dot{\forall}_{\iota \to o}\ (\overline{\lambda} c\ ((c\ x) \dot{\Rightarrow} (c\ y))))$$

Note

$$\varepsilon(x = y) \longrightarrow \forall c\ (\varepsilon(c\ x) \Rightarrow \varepsilon(c\ y))$$

Reflexivity, symmetry, transitivity, substitutivity

$$\forall x \forall y\ (\varepsilon(x = y) \Rightarrow (x/z)A \Rightarrow (y/z)A)$$

Peano numbers

Need to construct natural numbers
Without axioms or reduction rules no infinite set

Models of Simple type theory where all types are finite

The axiom of infinity

An infinity of elements of type ι

A non surjective injection f

g left inverse, E contains no element of the image of f, a element of E

$$(g\ (f\ x)) \longrightarrow x$$

$$(E\ a) \longrightarrow \dot{\top}$$

$$(E\ (f\ x)) \longrightarrow \dot{\bot}$$

Peano numbers

Call f S, call g $Pred$, call E $Null$ and call a 0

$$(Pred\ (S\ x)) \longrightarrow x$$

$$(Null\ 0) \longrightarrow \dot{\top}$$

$$(Null\ (S\ x)) \longrightarrow \dot{\bot}$$

Define the set of natural numbers, as the smallest set containing 0 and closed by the successor function

Simple type theory with Peano numbers

Models of Simple type theory with Peano numbers

A model valued in $\{0, 1\}$: same construction as for Simple type theory, but take $\mathcal{M}_\iota = \mathbb{N}$

Super-consistency: $\mathcal{M}_\iota = \mathbb{N}$, $\mathcal{M}_o = \mathcal{B}$

An alternative definition of natural numbers: Cantor numbers

Finite cardinals
An infinity of elements of type ι
Equinumerosity on sets of elements of ι

Cardinals: equivalence classes for this relation (type $(\iota \to o) \to o$)
0, successor, prove successor injective and not surjective

Smallest set containing zero and closed by successor

Another alternative definition: Church numbers

Iterators

Type $\iota \to (\iota \to \iota) \to \iota$

n is $\overline{\lambda} x \overline{\lambda} f \ (f \ (f \ \ldots \ (f \ x)))$

0, successor, prove successor injective and not surjective

Natural numbers as the smallest set containing zero and closed by successor

But no way to define Von Neumann numbers

V. The existence of functions in Simple type theory

To prove $\exists x\ A$ prove $(t/x)A$ and use $\exists$-intro

Can we prove

$$\exists f\ \forall x\ (\varepsilon(N\ x) \Rightarrow ((\varepsilon(x = 0) \Rightarrow \varepsilon((f\ x) = 0)) \\ \wedge(\neg\varepsilon(x = 0) \Rightarrow \varepsilon((f\ x) = 1))))\ ?$$

No term expressing the function χ such that $\chi 0 = 0$ and $\chi n = 1$ otherwise

Proposition not provable in Simple type theory (counter model)

But easy to build relation R s.t. $\varepsilon(R\ x\ y)$ expresses that y is the image of x by χ

$$R = \overline{\lambda}x\overline{\lambda}y\ ((x = 0 \mathrel{\dot{\Rightarrow}} y = 0) \mathrel{\dot{\wedge}} (\dot{\neg}(x = 0) \mathrel{\dot{\Rightarrow}} y = 1))$$

$R : \iota \rightarrow \iota \rightarrow o$

And to prove, by induction on x, the proposition

$$\forall x\ (\varepsilon(N\ x) \Rightarrow \exists y\ (\varepsilon(N\ y) \wedge \varepsilon(R\ x\ y)))$$

R functional relation

Transform this relation into a function: a new axiom of choice
Constant C of type $(\iota \to o) \to \iota$

$$\forall E\ (\exists y\ \varepsilon(E\ y)) \Rightarrow \varepsilon(E\ (C\ E))$$

Transform any relation into a function

$$\phi = \overline{\lambda} x\ (C\ \overline{\lambda} y\ (R\ x\ y))$$

If $\forall x\ (\varepsilon(N\ x) \Rightarrow \exists y\ \varepsilon(R\ x\ y))$ provable then so is
$\forall x\ (\varepsilon(N\ x) \Rightarrow \varepsilon(R\ x\ (\phi\ x)))$.
No need for such an axiom in set theory: functions are relations

The axiom of choice to prove the existence of χ?

Enough to have the axiom of descriptions

$$\forall E\ (\exists_1 y\ \varepsilon(E\ y)) \Rightarrow \varepsilon(E\ (C\ E))$$

where $\exists_1 x\ A$ is an abbreviation for
$\exists x\ (A \wedge (\forall y\ (y/x)A \Rightarrow \varepsilon(y = x)))$

VI. Alternative formulations of Simple type theory

Abstracting a variable

For each x and t a term u s.t.

$$(u\ x) \longrightarrow^* t$$

Much bigger than t

Instead of S and K, a combinator $x_1, ..., x_n \mapsto t$ for each t
Still uncomfortable: to abstract y in $(x\ y)$ use the constant $x, y \mapsto (x\ y)$ and apply it back to x

The λ-calculus

$x \mapsto t$ not a constant
obtained by applying $\mapsto$ to the term t
$\lambda x : A\ t$

More comfortable, but more complex
λ binds a variable
Outside Predicate logic

Rules

$$((\lambda x : A\ t)\ x) \longrightarrow t$$

$$\varepsilon(\dot{\top}) \longrightarrow \top$$

$$\varepsilon(\dot{\bot}) \longrightarrow \bot$$

$$\varepsilon(\dot{\wedge}\ x\ y) \longrightarrow (\varepsilon(x) \wedge \varepsilon(y))$$

etc. The first rule *the β-reduction rule*
If not use the same name x for bound and free variable

$$((\lambda x : A\ t)\ u) \longrightarrow (u/x)t$$

Termination: exercise

Propositional contents

Perfect correspondence between propositions and propositional contents

Define sequents of propositional contents $t_1, ..., t_n \vdash u$ and drop ε
$t_1, ..., t_n \vdash u$ provable iff $\varepsilon(t_1), ..., \varepsilon(t_n) \vdash \varepsilon(u)$ is

Deduction rules directly on such sequents, for instance

$$\frac{\Gamma \vdash (\dot{\wedge}\ t\ u)}{\Gamma \vdash t}$$

After the break

An exercise: the termination of proof reduction