Towards a general proof theory of term-forming operators

Andrzej Indrzejczak

Department of Logic, University of Lodz

ExtenDD Seminar, Łódź, March 21, 2023

Term-forming operators (variable-binding term operators):

Term-forming operators (variable-binding term operators):

Examples:

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x \varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x \varphi$ - a(n) x such that φ;

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x \varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x \varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x \varphi$ - the number of x such that φ;

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x \varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x \varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x \varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x \varphi$ - the property of being φ.

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x\varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x\varphi$ - the property of being φ.

Why do they matter?

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x\varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x\varphi$ - the property of being φ.

Why do they matter?

1. The role of complex terms is crucial in communication.

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x\varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x\varphi$ - the property of being φ.

Why do they matter?

1. The role of complex terms is crucial in communication.
2. The role of complex terms is totally neglected in modern logic.

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x\varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x\varphi$ - the property of being φ.

Why do they matter?

1. The role of complex terms is crucial in communication.
2. The role of complex terms is totally neglected in modern logic.

It's time to fill this gap

Term-forming operators (variable-binding term operators):

Examples:

- iota-operator (Peano): $\imath x\varphi$ - the (only) x such that φ;
- epsilon-operator (Hilbert): $\epsilon x\varphi$ - a(n) x such that φ;
- abstraction-operator: $\{x : \varphi\}$ - the set of (all) x satisfying φ;
- counting-operator (Frege): $\sharp x\varphi$ - the number of x such that φ;
- lambda-operator (Church): $\lambda x\varphi$ - the property of being φ.

Why do they matter?

1. The role of complex terms is crucial in communication.
2. The role of complex terms is totally neglected in modern logic.

It's time to fill this gap $\Longrightarrow$ ExtenDD project.

Term-forming operators (variable-binding term operators):

Term-forming operators (variable-binding term operators):

Is there a general theory of such operators?

Term-forming operators (variable-binding term operators):

Is there a general theory of such operators?

There are two attempts to develop such a theory.

Term-forming operators (variable-binding term operators):

Is there a general theory of such operators?

There are two attempts to develop such a theory.

1. A theory independently proposed by Scott, by Hatcher, Corcoran and Herring, and by Da Costa.

Term-forming operators (variable-binding term operators):

Is there a general theory of such operators?

There are two attempts to develop such a theory.

1. A theory independently proposed by Scott, by Hatcher, Corcoran and Herring, and by Da Costa.
   - D. Scott.
     Existence and description in formal logic, in B. Russell, Philosopher of the Century, Little, Brown and Co., Boston 1967.
   - W. S. Hatcher.
     1968. Foundations of Mathematics, Saunders, Philadelphia.
     1982. The logical foundations of Mathematics, Pergamon
   - J. Corcoran and J. Herring.
     1971.- Notes on a semantical analysis of variable-binding term operators, Logique et Analyse 55, pp. 644-657.
   - J. Corcoran, W. S. Hatcher and J. Herring.
     1972.- Variable-binding term operators, Zeitschr. f. math. Logik u. Grund. d. Math. 18, pp. 177-182.

Term-forming operators (variable-binding term operators):

Is there a general theory of such operators?

There are two attempts to develop such a theory.

1. A theory independently proposed by Scott, by Hatcher, Corcoran and Herring, and by Da Costa.
   - N. C. A. da Costa.
     1973.- Review of Corcoran, Hatcher and Herring 1972, Zentralblatt f. Math. 257, pp. 8-9.
     1980.- A model-theoretical approach to variable-binding term operators, in: Mathematical Logic in Latin America, pp. 133–162, North-Holland
2. An approach developed by Neil Tennant
   1978. Natural Logic, Edinburgh.
   1987. Anti-Realism and Logic, Oxford.
   2004.- A general theory of Abstraction Operators, The Philosophical Quaterly 54(214), pp. 105–133.
   2022. The Logic of Number, Oxford.

Term-forming operators (variable-binding term operators):

Term-forming operators (variable-binding term operators):

The first theory (Scott, Hatcher, Corcoran and Herring, Da Costa)

Term-forming operators (variable-binding term operators):

The first theory (Scott, Hatcher, Corcoran and Herring, Da Costa)

It is based on two general principles added to PFFOLI (positive free first-order logic with identity) [Scott] or to CFOLI (classical FOLI) [the remaining authors].

Term-forming operators (variable-binding term operators):

The first theory (Scott, Hatcher, Corcoran and Herring, Da Costa)

It is based on two general principles added to PFFOLI (positive free first-order logic with identity) [Scott] or to CFOLI (classical FOLI) [the remaining authors].

EXT: $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \tau x\varphi(x) = \tau x\psi(x)$
AV: $\tau x\varphi(x) = \tau y\varphi(y)$

Term-forming operators (variable-binding term operators):

The first theory (Scott, Hatcher, Corcoran and Herring, Da Costa)

It is based on two general principles added to PFFOLI (positive free first-order logic with identity) [Scott] or to CFOLI (classical FOLI) [the remaining authors].

EXT: $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \tau x\varphi(x) = \tau x\psi(x)$
AV: $\tau x\varphi(x) = \tau y\varphi(y)$

or, equivalently:

EXTAV: $\forall xy(x = y \rightarrow \varphi(x) \leftrightarrow \psi(y)) \rightarrow \tau x\varphi(x) = \tau y\psi(y)$

Term-forming operators (variable-binding term operators):

The first theory (Scott, Hatcher, Corcoran and Herring, Da Costa)

It is based on two general principles added to PFFOLI (positive free first-order logic with identity) [Scott] or to CFOLI (classical FOLI) [the remaining authors].

EXT: $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \tau x\varphi(x) = \tau x\psi(x)$
AV: $\tau x\varphi(x) = \tau y\varphi(y)$

or, equivalently:

EXTAV: $\forall xy(x = y \rightarrow \varphi(x) \leftrightarrow \psi(y)) \rightarrow \tau x\varphi(x) = \tau y\psi(y)$

It may be also developed in the setting of FOL (no identity) by means of:

EXT': $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow (\chi[\tau x\varphi(x)] \leftrightarrow \chi[\tau x\psi(x)])$
AV': $\chi[\tau x\varphi(x)] \leftrightarrow \chi[\tau y\psi(y)]$

Term-forming operators (variable-binding term operators):

Term-forming operators (variable-binding term operators):

The first theory – possible objections:

Term-forming operators (variable-binding term operators):

The first theory – possible objections:

1. In a sense it is too general and too weak. For several operators one needs additional principles. For example:

The first theory – possible objections:

1. In a sense it is too general and too weak. For several operators one needs additional principles. For example:

For ι Rosser adds to EXT and AV:
$\exists_1 x\varphi(x) \rightarrow \forall x(x = \iota x\varphi(x) \leftrightarrow \varphi(x))$

Term-forming operators (variable-binding term operators):

The first theory – possible objections:

1. In a sense it is too general and too weak. For several operators one needs additional principles. For example:

For ι Rosser adds to EXT and AV:
$\exists_1 x\varphi(x) \rightarrow \forall x(x = \iota x\varphi(x) \leftrightarrow \varphi(x))$

and Da Costa adds:
$\exists_1 x\varphi(x) \rightarrow \forall x(x = \iota x\varphi(x) \rightarrow \varphi(x))$
$\neg\exists_1 x\varphi(x) \rightarrow \iota x\varphi(x) = \iota x(x \neq x)$

Term-forming operators (variable-binding term operators):

The first theory – possible objections:

1. In a sense it is too general and too weak. For several operators one needs additional principles. For example:

For ι Rosser adds to EXT and AV:
$\exists_1 x\varphi(x) \to \forall x(x = \iota x\varphi(x) \leftrightarrow \varphi(x))$

and Da Costa adds:
$\exists_1 x\varphi(x) \to \forall x(x = \iota x\varphi(x) \to \varphi(x))$
$\neg\exists_1 x\varphi(x) \to \iota x\varphi(x) = \iota x(x \neq x)$

For ϵ we need to add:
$\exists x\varphi(x) \to \forall x(x = \epsilon x\varphi(x) \to \varphi(x))$

Term-forming operators (variable-binding term operators):

The first theory – possible objections:

1. In a sense it is too general and too weak. For several operators one needs additional principles. For example:

For ι Rosser adds to EXT and AV:
$\exists_1 x\varphi(x) \to \forall x(x = \iota x\varphi(x) \leftrightarrow \varphi(x))$

and Da Costa adds:
$\exists_1 x\varphi(x) \to \forall x(x = \iota x\varphi(x) \to \varphi(x))$
$\neg\exists_1 x\varphi(x) \to \iota x\varphi(x) = \iota x(x \neq x)$

For ϵ we need to add:
$\exists x\varphi(x) \to \forall x(x = \epsilon x\varphi(x) \to \varphi(x))$

2. For some theories of DD it is too strong, e.g. for the Russellian theory.

Term-forming operators (variable-binding term operators):

Term-forming operators (variable-binding term operators):

The second theory (Tennant)

Term-forming operators (variable-binding term operators):

The second theory (Tennant)

Developed in the setting of NFFOLI (negative free FOLI).

Term-forming operators (variable-binding term operators):

The second theory (Tennant)

Developed in the setting of NFFOLI (negative free FOLI).
Based on the following ND rules:

τI If $\varphi(a), Ea \vdash Rat$ and $Rat \vdash \varphi(a)$ and Et, then $t = \tau x\varphi(x)$;
$\tau E1$ If $t = \tau x\varphi(x)$ and $\varphi(b)$ and Eb, then Rbt
$\tau E2$ If $t = \tau x\varphi(x)$, then Et
$\tau E3$ If $t = \tau x\varphi(x)$ and Rbt, then $\varphi(b)$

where a is an eigenvariable, and R is the specific relation involved in the characterisation of τ; e.g. $=$ for ι, $\in$ for set builder.

Term-forming operators (variable-binding term operators):

The second theory (Tennant)

Developed in the setting of NFFOLI (negative free FOLI).
Based on the following ND rules:

τI If $\varphi(a), Ea \vdash Rat$ and $Rat \vdash \varphi(a)$ and Et, then $t = \tau x\varphi(x)$;
$\tau E1$ If $t = \tau x\varphi(x)$ and $\varphi(b)$ and Eb, then Rbt
$\tau E2$ If $t = \tau x\varphi(x)$, then Et
$\tau E3$ If $t = \tau x\varphi(x)$ and Rbt, then $\varphi(b)$

where a is an eigenvariable, and R is the specific relation involved in the characterisation of τ; e.g. $=$ for ι, $\in$ for set builder.

Note that Tennant's natural logicist's approach uses single-barreled characterisation of operators in contrast to double-barreled abstraction principles based on equivalences, preferred by neo-logicists and present also in the first approach.

The basic system GC for CFOL:

The basic system GC for CFOL:

$(Cut)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Pi \Rightarrow \Sigma}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}$ $\qquad (AX)\ \varphi, \Gamma \Rightarrow \Delta, \varphi$

$(\neg\Rightarrow)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi}{\neg\varphi, \Gamma \Rightarrow \Delta}$ $\qquad (\Rightarrow\neg)\ \dfrac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}$ $\qquad (W\Rightarrow)\ \dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}$

$(\Rightarrow\wedge)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}$ $\qquad (\wedge\Rightarrow)\ \dfrac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}$ $\qquad (\Rightarrow W)\ \dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}$

$(\vee\Rightarrow)\ \dfrac{\varphi, \Gamma \Rightarrow \Delta \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}$ $\qquad (\Rightarrow\vee)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$ $\qquad (C\Rightarrow)\ \dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}$

$(\rightarrow\Rightarrow)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \rightarrow \psi, \Gamma \Rightarrow \Delta}$ $\qquad (\Rightarrow\rightarrow)\ \dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi}$ $\qquad (\Rightarrow C)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}$

$(\leftrightarrow\Rightarrow)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi, \psi \qquad \varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \leftrightarrow \psi, \Gamma \Rightarrow \Delta}$ $\qquad (\forall\Rightarrow)\ \dfrac{\varphi[x/t], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta}$ $\qquad (\Rightarrow\exists)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$

$(\Rightarrow\leftrightarrow)\ \dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi \qquad \psi, \Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \leftrightarrow \psi}$ $\qquad (\Rightarrow\forall)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \forall x\varphi}$ $\qquad (\exists\Rightarrow)\ \dfrac{\varphi[x/a], \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta}$

where a is a fresh parameter (eigenvariable), not present in Γ, Δ and φ.

Variants:

Variants:

1. GPC: instead of $(\forall\Rightarrow)$ and $(\Rightarrow\exists)$ we have:

$$(\forall\Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

Variants:

1. GPC: instead of $(\forall\Rightarrow)$ and $(\Rightarrow\exists)$ we have:

$$(\forall\Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

2. GF: Change all quantifier rules into:

$$(\forall\Rightarrow)\ \frac{\varphi[x/t], \Gamma \Rightarrow \Delta}{Et, \forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\forall)\ \frac{Ea, \Gamma \Rightarrow \Delta, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \forall x\varphi}$$

$$(\exists\Rightarrow)\ \frac{Ea, \varphi[x/a], \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{Et, \Gamma \Rightarrow \Delta, \exists x\varphi}$$

Variants:

1. GPC: instead of $(\forall\Rightarrow)$ and $(\Rightarrow\exists)$ we have:

$$(\forall\Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

2. GF: Change all quantifier rules into:

$$(\forall\Rightarrow)\ \frac{\varphi[x/t], \Gamma \Rightarrow \Delta}{Et, \forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\forall)\ \frac{Ea, \Gamma \Rightarrow \Delta, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \forall x\varphi}$$

$$(\exists\Rightarrow)\ \frac{Ea, \varphi[x/a], \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{Et, \Gamma \Rightarrow \Delta, \exists x\varphi}$$

For pure version b instead of t.

Variants:

1. GPC: instead of (∀⇒) and (⇒∃) we have:

$$(\forall\Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

2. GF: Change all quantifier rules into:

$$(\forall\Rightarrow)\ \frac{\varphi[x/t], \Gamma \Rightarrow \Delta}{Et, \forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\forall)\ \frac{Ea, \Gamma \Rightarrow \Delta, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \forall x\varphi}$$

$$(\exists\Rightarrow)\ \frac{Ea, \varphi[x/a], \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{Et, \Gamma \Rightarrow \Delta, \exists x\varphi}$$

For pure version b instead of t.

3. For NFOL add:

$$(Str)\ \frac{Et, \Gamma \Rightarrow \Delta}{\varphi(t), \Gamma \Rightarrow \Delta}$$ where φ is atomic

Variants:

1. GPC: instead of $(\forall\Rightarrow)$ and $(\Rightarrow\exists)$ we have:

$$(\forall\Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

2. GF: Change all quantifier rules into:

$$(\forall\Rightarrow)\ \frac{\varphi[x/t], \Gamma \Rightarrow \Delta}{Et, \forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\forall)\ \frac{Ea, \Gamma \Rightarrow \Delta, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \forall x\varphi}$$

$$(\exists\Rightarrow)\ \frac{Ea, \varphi[x/a], \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{Et, \Gamma \Rightarrow \Delta, \exists x\varphi}$$

For pure version b instead of t.

3. For NFOL add:

$$(Str)\ \frac{Et, \Gamma \Rightarrow \Delta}{\varphi(t), \Gamma \Rightarrow \Delta} \qquad \text{where } \varphi \text{ is atomic}$$

Desiderata for proof-theoretic characterisation: cut-elimination, subformula-, subterm-property.

How to deal with identity?

In SC framework:

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

1. Addition of axiomatic sequents $\Rightarrow \varphi$ for each axiom φ.

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

1. Addition of axiomatic sequents $\Rightarrow \varphi$ for each axiom φ.
2. Addition of "mathematical basic sequents" which consists of atomic formulae.

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

1. Addition of axiomatic sequents $\Rightarrow \varphi$ for each axiom φ.
2. Addition of "mathematical basic sequents" which consists of atomic formulae.
3. Addition of all axioms as a context in the antecedents of all provable sequents.

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

1. Addition of axiomatic sequents $\Rightarrow \varphi$ for each axiom φ.
2. Addition of "mathematical basic sequents" which consists of atomic formulae.
3. Addition of all axioms as a context in the antecedents of all provable sequents.
4. Addition of new rules corresponding to axioms.

How to deal with identity?

In SC framework:

I Global approach (by substitution on the whole sequent).

II Local approach:

1. Addition of axiomatic sequents $\Rightarrow \varphi$ for each axiom φ.
2. Addition of "mathematical basic sequents" which consists of atomic formulae.
3. Addition of all axioms as a context in the antecedents of all provable sequents.
4. Addition of new rules corresponding to axioms.

In the first case:

Ref: $\Rightarrow t = t$

LL: $\Rightarrow t_1 = t_2 \rightarrow (\varphi[x/t_1] \rightarrow \varphi[x/t_2])$, where φ is atomic

Rules for = (Rule-maker theorem Indrzejczak 2013)

$(1=)\ \dfrac{t=t,\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}$ for Ref and the following for LL:

$(2=)\ \dfrac{\varphi[x/t_2],\Gamma\Rightarrow\Delta}{t_1=t_2,\varphi[x/t_1],\Gamma\Rightarrow\Delta}$ $(3=)\ \dfrac{\Gamma\Rightarrow\Delta,\varphi[x/t_1]}{t_1=t_2,\Gamma\Rightarrow\Delta,\varphi[x/t_2]}$

$$(4=)\ \dfrac{\Gamma\Rightarrow\Delta,t_1=t_2}{\varphi[x/t_1],\Gamma\Rightarrow\Delta,\varphi[x/t_2]}$$

$$(5=)\ \dfrac{\Gamma\Rightarrow\Delta,t_1=t_2 \quad \Gamma\Rightarrow\Delta,\varphi[x/t_1]}{\Gamma\Rightarrow\Delta,\varphi[x/t_2]}$$

$$(6=)\ \dfrac{\Gamma\Rightarrow\Delta,t_1=t_2 \quad \varphi[x/t_2],\Gamma\Rightarrow\Delta}{\varphi[x/t_1],\Gamma\Rightarrow\Delta}$$

$$(7=)\ \dfrac{\Gamma\Rightarrow\Delta,\varphi[x/t_1] \quad \varphi[x/t_2],\Gamma\Rightarrow\Delta}{t_1=t_2,\Gamma\Rightarrow\Delta}$$

$$(8=)\ \dfrac{\Gamma\Rightarrow\Delta,t_1=t_2 \quad \Gamma\Rightarrow\Delta,\varphi[x/t_1] \quad \varphi[x/t_2],\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}$$

The first approach to term-forming operators:

The first approach to term-forming operators:

The first formalisation GT1: to GC add:

$$(Ext)\ \frac{\varphi(a), \Gamma \Rightarrow \Delta, \psi(a) \qquad \psi(a), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x\varphi(x) = \tau x\psi(x)}$$

$$(AV)\ \frac{\tau x\varphi(x) = \tau y\varphi(y), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

The first approach to term-forming operators:

The first formalisation GT1: to GC add:

$$(Ext)\ \frac{\varphi(a), \Gamma \Rightarrow \Delta, \psi(a) \qquad \psi(a), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x \varphi(x) = \tau x \psi(x)}$$

$$(AV)\ \frac{\tau x \varphi(x) = \tau y \varphi(y), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

The second formalisation GT2: add only:

$$(ExtAV)\ \frac{a = b, \varphi(a), \Gamma \Rightarrow \Delta, \psi(b) \qquad a = b, \psi(b), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x \varphi(x) = \tau y \psi(y)}$$

The first approach to term-forming operators:

The first approach to term-forming operators:

The rules are adequate:

The first approach to term-forming operators:

The rules are adequate:
Provability of EXTAV (axiom):

$$
(ExtAV)\ \dfrac{(\forall\Rightarrow)\ \dfrac{(\to\Rightarrow)\ \dfrac{a=b\Rightarrow a=b \qquad \varphi(a)\leftrightarrow\psi(b),\varphi(a)\Rightarrow\psi(b)}{a=b\to(\varphi(a)\leftrightarrow\psi(b)),a=b,\varphi(a)\Rightarrow\psi(b)}}{\forall xy(x=y\to(\varphi(x)\leftrightarrow\psi(y))),a=b,\varphi(a)\Rightarrow\psi(b)} \qquad D}{(\Rightarrow\to)\ \dfrac{\forall xy(x=y\to(\varphi(x)\leftrightarrow\psi(y)))\Rightarrow\tau x\varphi(x)=\tau y\psi(y)}{\Rightarrow\forall xy(x=y\to(\varphi(x)\leftrightarrow\psi(y)))\to\tau x\varphi(x)=\tau y\psi(y)}}
$$

where D is a proof of
$\forall xy(x=y\to(\varphi(x)\leftrightarrow\psi(y))),a=b,\psi(b)\Rightarrow\varphi(b)$.

The first approach to term-forming operators:

The first approach to term-forming operators:

The rules are adequate:
Derivability of $(ExtAV)$:

$$
\dfrac{(\Rightarrow\leftrightarrow)\ \dfrac{a = b, \varphi(a), \Gamma \Rightarrow \Delta, \psi(b) \qquad a = b, \psi(b), \Gamma \Rightarrow \Delta, \varphi(a)}{(\Rightarrow\rightarrow)\ \dfrac{a = b, \Gamma \Rightarrow \Delta, \varphi(a) \leftrightarrow \psi(b)}{(\Rightarrow\forall)\ \dfrac{\Gamma \Rightarrow \Delta, a = b \rightarrow (\varphi(a) \leftrightarrow \psi(b))}{\Gamma \Rightarrow \Delta, \forall xy(x = y \rightarrow (\varphi(x) \leftrightarrow \psi(y)))}}} \qquad D}{\Gamma \Rightarrow \Delta, \tau x\varphi(x) = \tau y\psi(y)}\ (Cut)
$$

where D is a proof of
$\forall xy(x = y \rightarrow (\varphi(x) \leftrightarrow \psi(y))) \Rightarrow \tau x\varphi(x) = \tau y\psi(y)$ from the axiom $\Rightarrow EXTAV$.

The first approach to term-forming operators:

The first approach to term-forming operators:

Problems to overcome (in both systems):

The first approach to term-forming operators:

Problems to overcome (in both systems):

1) How to avoid the problem with the lost subformula-property for $(\Rightarrow \exists)$ and $(\forall \Rightarrow)$?

The first approach to term-forming operators:

Problems to overcome (in both systems):

1) How to avoid the problem with the lost subformula-property for ($\Rightarrow \exists$) and ($\forall \Rightarrow$)?

2) How to formulate the rules for LL to avoid clash on cut-formulas generated with (Ext) (($ExtAV$))?

The first approach to term-forming operators:

The first approach to term-forming operators:

ad 1. Restrict all quantifier rules to parameters (use GPC), and to avoid the loss of generality add to GT1 or GT2:

$$(a \Rightarrow)\ \frac{a = \tau x \varphi(x), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

The first approach to term-forming operators:

ad 1. Restrict all quantifier rules to parameters (use GPC), and to avoid the loss of generality add to GT1 or GT2:

$$(a \Rightarrow)\ \frac{a = \tau x\varphi(x), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

The resulting system is GPT1 (GPT2) [i.e. GC with $(a \Rightarrow)$ and $(Ext), (AV)$ or $(ExtAV)$] and it is equivalent to GT1 (GT2).

The first approach to term-forming operators:

The first approach to term-forming operators:

$(a \Rightarrow)$ is derivable in GT1 (GT2) with cut:

$$\dfrac{(\Rightarrow \exists)\,\dfrac{(Ext)\,\dfrac{\varphi(a) \Rightarrow \varphi(a) \qquad \varphi(a) \Rightarrow \varphi(a)}{\Rightarrow \tau x\varphi(x) = \tau x\varphi(x)}}{\Rightarrow \exists y(y = \tau x\varphi(x))} \qquad \dfrac{a = \tau x\varphi(x), \Gamma \Rightarrow \Delta}{\exists y(y = \tau x\varphi(x)), \Gamma \Rightarrow \Delta}\,(\exists \Rightarrow}{\Gamma \Rightarrow \Delta}\,(Cut)$$

The first approach to term-forming operators:

$(a \Rightarrow)$ is derivable in GT1 (GT2) with cut:

$$
(Cut)\ \dfrac{(\Rightarrow \exists)\ \dfrac{(Ext)\ \dfrac{\varphi(a) \Rightarrow \varphi(a) \qquad \varphi(a) \Rightarrow \varphi(a)}{\Rightarrow \tau x\varphi(x) = \tau x\varphi(x)}}{\Rightarrow \exists y(y = \tau x\varphi(x))} \qquad \dfrac{a = \tau x\varphi(x), \Gamma \Rightarrow \Delta}{\exists y(y = \tau x\varphi(x)), \Gamma \Rightarrow \Delta}\ (\exists \Rightarrow}{\Gamma \Rightarrow \Delta}
$$

[a proof in GT2 similar]

The first approach to term-forming operators:

The first approach to term-forming operators:

Unrestricted $(\forall \Rightarrow), (\Rightarrow \exists)$ are derivable in GPT1 (GPT2) with unrestricted LL and cut:

$$(Cut)\ \cfrac{\Gamma \Rightarrow \Delta, \varphi(\tau x \psi(x)) \qquad \varphi(\tau x \psi(x)), a = \tau x \psi(x) \Rightarrow \varphi(a)}{(a \Rightarrow)\ \cfrac{(\Rightarrow \exists)\ \cfrac{a = \tau x \psi(x), \Gamma \Rightarrow \Delta, \varphi(a)}{a = \tau x \psi(x), \Gamma \Rightarrow \Delta, \exists x \varphi}}{\Gamma \Rightarrow \Delta, \exists x \varphi}}$$

The first approach to term-forming operators:

Unrestricted $(\forall \Rightarrow), (\Rightarrow \exists)$ are derivable in GPT1 (GPT2) with unrestricted LL and cut:

$$(Cut)\ \dfrac{\Gamma \Rightarrow \Delta, \varphi(\tau x\psi(x)) \qquad \varphi(\tau x\psi(x)), a = \tau x\psi(x) \Rightarrow \varphi(a)}{(a \Rightarrow)\ \dfrac{(\Rightarrow \exists)\ \dfrac{a = \tau x\psi(x), \Gamma \Rightarrow \Delta, \varphi(a)}{a = \tau x\psi(x), \Gamma \Rightarrow \Delta, \exists x\varphi}}{\Gamma \Rightarrow \Delta, \exists x\varphi}}$$

and similarly for $(\forall \Rightarrow)$.

The first approach to term-forming operators:

The first approach to term-forming operators:

Which rule for LL should we use?

The first approach to term-forming operators:

Which rule for LL should we use?
All variants except $(5=)$ and $(8=)$ make a clash in the proof of cut elimination, e.g. $(2=)$:

$$\dfrac{(Ext)\ \dfrac{\varphi(a), \Gamma \Rightarrow \Delta, \psi(a) \qquad \psi(a), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x\varphi(x) = \tau x\psi(x)} \qquad \dfrac{\chi[z/\tau x\varphi(x)], \Pi \Rightarrow \Sigma}{\tau x\varphi(x) = \tau x\psi(x), \chi[z/\tau x\psi(x)], \Pi \Rightarrow \Sigma}\ (2=)}{\chi[z/\tau x\psi(x)], \Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)$$

The first approach to term-forming operators:

Which rule for LL should we use?
All variants except $(5 =)$ and $(8 =)$ make a clash in the proof of cut elimination, e.g. $(2 =)$:

$$(Cut)\ \dfrac{(Ext)\ \dfrac{\varphi(a), \Gamma \Rightarrow \Delta, \psi(a) \qquad \psi(a), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x\varphi(x) = \tau x\psi(x)} \qquad \dfrac{\chi[z/\tau x\varphi(x)], \Pi \Rightarrow \Sigma}{\tau x\varphi(x) = \tau x\psi(x), \chi[z/\tau x\psi(x)], \Pi \Rightarrow \Sigma}\ (2 =)}{\chi[z/\tau x\psi(x)], \Gamma, \Pi \Rightarrow \Delta, \Sigma}$$

But if we use $(5 =)$, i.e.

$$(LL)\ \dfrac{\Gamma \Rightarrow \Delta, t = t' \qquad \Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \varphi[x/t']}$$

We can avoid clash in the proof of cut elimination (all rules are right-sided).

The first approach to term-forming operators:

Which rule for LL should we use?
All variants except $(5=)$ and $(8=)$ make a clash in the proof of cut elimination, e.g. $(2=)$:

$$\dfrac{(Ext)\ \dfrac{\varphi(a), \Gamma \Rightarrow \Delta, \psi(a) \qquad \psi(a), \Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \tau x\varphi(x) = \tau x\psi(x)} \qquad \dfrac{\chi[z/\tau x\varphi(x)], \Pi \Rightarrow \Sigma}{\tau x\varphi(x) = \tau x\psi(x), \chi[z/\tau x\psi(x)], \Pi \Rightarrow \Sigma}\ (2=)}{\chi[z/\tau x\psi(x)], \Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)$$

But if we use $(5=)$, i.e.

$$(LL)\ \dfrac{\Gamma \Rightarrow \Delta, t = t' \qquad \Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \varphi[x/t']}$$

We can avoid clash in the proof of cut elimination (all rules are right-sided).

The cut elimination theorem and the subformula property (but not the subterm property) hold for both Systems GPT1 and GPT2.

The first approach to term-forming operators:

The first approach to term-forming operators:

Some remarks on the identity treatment:

Some remarks on the identity treatment:

1. Note that we can keep:

$$(R)\ \frac{b = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

[for $\tau x \varphi(x) = \tau x \varphi(x)$ it is derivable by (Ext)]

The first approach to term-forming operators:

Some remarks on the identity treatment:

1. Note that we can keep:

$$(R)\ \frac{b = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

[for $\tau x \varphi(x) = \tau x \varphi(x)$ it is derivable by (Ext)]

2. In fact we can keep also $(2=)$ for parameters (and even for mixed $b = t$ with the second premiss not of the form $t = t'$); the only troublesome cases of LL which make a clash in the proof of cut elimination are:

1. $b = t, t = t' \Rightarrow b = t'$
2. $t = t', \varphi(t) \Rightarrow \varphi(t')$
3. $t = t', t' = t'' \Rightarrow t = t''$

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)
- Tennant's basic logic of classes

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)
- Tennant's basic logic of classes
- paraconsistent set theory (naive)

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)
- Tennant's basic logic of classes
- paraconsistent set theory (naive)
- Cantorian set theory as developed by Oliver and Smiley

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)
- Tennant's basic logic of classes
- paraconsistent set theory (naive)
- Cantorian set theory as developed by Oliver and Smiley
- ZF

Application to set-builders

Several kinds of set theory can be taken into account, in particular:

- Quine's NF (its formalisation in Rosser, Hatcher)
- Quine's ML
- Quine's theory of virtual sets (its formalisation in Scott)
- Tennant's basic logic of classes
- paraconsistent set theory (naive)
- Cantorian set theory as developed by Oliver and Smiley
- ZF
- BG

Application to set-builders – preliminary issues concerning $=$; possible choices:

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: *ExtAx* $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: $ExtAx$ $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \to x = y)$ [the converse is provable by LL]

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: *ExtAx* $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
[the converse is provable by LL]
2. Start with CFOL (only $\in$ primitive) and defined $=$ either as:

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: *ExtAx* $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
[the converse is provable by LL]

2. Start with CFOL (only $\in$ primitive) and defined $=$ either as:

2.1. (Leibnizian): $t = t' := \forall z(t \in z \leftrightarrow t' \in z)$;
then obtain a standard characterisation of $=$ and add *ExtAx*

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: *ExtAx* $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \to x = y)$
[the converse is provable by LL]

2. Start with CFOL (only $\in$ primitive) and defined $=$ either as:

2.1. (Leibnizian): $t = t' := \forall z(t \in z \leftrightarrow t' \in z)$;
then obtain a standard characterisation of $=$ and add *ExtAx*
[the converse is provable by LL. In principle the same effect as in approach 1]

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: $ExtAx$ $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \to x = y)$ [the converse is provable by LL]

2. Start with CFOL (only $\in$ primitive) and defined $=$ either as:

2.1. (Leibnizian): $t = t' := \forall z(t \in z \leftrightarrow t' \in z)$;
then obtain a standard characterisation of $=$ and add $ExtAx$
[the converse is provable by LL. In principle the same effect as in approach 1]

2.2. $t = t' := \forall z(z \in t \leftrightarrow z \in t')$
but then we must add a form of LL as an extensionality axiom:
$ExtAx'$ $\forall xyz(x = y \to (x \in z \to y \in z))$

Application to set-builders – preliminary issues concerning $=$; possible choices:

1. Start with CFOLI ($=$ and $\in$ primitive) with some axioms/rules for $=$ and add: $ExtAx$ $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
[the converse is provable by LL]

2. Start with CFOL (only $\in$ primitive) and defined $=$ either as:

2.1. (Leibnizian): $t = t' := \forall z(t \in z \leftrightarrow t' \in z)$;
then obtain a standard characterisation of $=$ and add $ExtAx$
[the converse is provable by LL. In principle the same effect as in approach 1]

2.2. $t = t' := \forall z(z \in t \leftrightarrow z \in t')$
but then we must add a form of LL as an extensionality axiom:
$ExtAx'$ $\forall xyz(x = y \rightarrow (x \in z \rightarrow y \in z))$
[the form with $z \in x$ derivable from definition; cf. Quine, Rosser, Mendelson, Hatcher]

Application to set-builders – preliminary issues concerning =; possible choices:

1. Start with CFOLI (= and $\in$ primitive) with some axioms/rules for = and add: *ExtAx* $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \to x = y)$ [the converse is provable by LL]

2. Start with CFOL (only $\in$ primitive) and defined = either as:

2.1. (Leibnizian): $t = t' := \forall z(t \in z \leftrightarrow t' \in z)$;
then obtain a standard characterisation of = and add *ExtAx*
[the converse is provable by LL. In principle the same effect as in approach 1]

2.2. $t = t' := \forall z(z \in t \leftrightarrow z \in t')$
but then we must add a form of LL as an extensionality axiom:
ExtAx' $\forall xyz(x = y \to (x \in z \to y \in z))$
[the form with $z \in x$ derivable from definition; cf. Quine, Rosser, Mendelson, Hatcher]

It explains a difference between nomenclature in the use of the term extensionality axiom either for *ExtAx* or for LL (i.e. *ExtAx'*).

Application to set-builders

Quine's NF

Application to set-builders

Quine's NF

Language with $\in$ primitive.

Application to set-builders

Quine's NF

Language with $\in$ primitive.

$=$ defined:

$$t = t' := \forall z(z \in t \leftrightarrow z \in t')$$

Application to set-builders

Quine's NF

Language with $\in$ primitive.

$=$ defined:

$t = t' := \forall z(z \in t \leftrightarrow z \in t')$

[Note that the approach 2.2. is involved]

Application to set-builders

Quine's NF

Language with $\in$ primitive.

$=$ defined:

$t = t' := \forall z(z \in t \leftrightarrow z \in t')$

[Note that the approach 2.2. is involved]

Two axioms:

Abs $\forall x(x \in \{y : \varphi(y)\} \leftrightarrow \varphi(y/x))$, φ stratified.
Ext $\forall xy(x = y \rightarrow (\varphi(x) \leftrightarrow \varphi(y)))$

Application to set-builders – Quine's NF

Application to set-builders – Quine's NF

Take GCP and add:

Application to set-builders – Quine's NF

Take GCP and add:

$$(\Rightarrow=)\ \frac{a \in t, \Gamma \Rightarrow \Delta, a \in t' \qquad a \in t', \Gamma \Rightarrow \Delta, a \in t}{\Gamma \Rightarrow \Delta, t = t'}$$

$$(=\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, b \in t, b \in t' \qquad b \in t, b \in t', \Gamma \Rightarrow \Delta}{t = t', \Gamma \Rightarrow \Delta}$$

$$(Abs \Rightarrow)\ \frac{\varphi(t), \Gamma \Rightarrow \Delta}{t \in \{x : \varphi(x)\}, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow Abs)\ \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, t \in \{x : \varphi(x)\}}$$

Application to set-builders – Quine's NF

Take GCP and add:

$$(\Rightarrow=)\ \frac{a \in t, \Gamma \Rightarrow \Delta, a \in t' \qquad a \in t', \Gamma \Rightarrow \Delta, a \in t}{\Gamma \Rightarrow \Delta, t = t'}$$

$$(=\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, b \in t, b \in t' \qquad b \in t, b \in t', \Gamma \Rightarrow \Delta}{t = t', \Gamma \Rightarrow \Delta}$$

$$(Abs \Rightarrow)\ \frac{\varphi(t), \Gamma \Rightarrow \Delta}{t \in \{x : \varphi(x)\}, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow Abs)\ \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, t \in \{x : \varphi(x)\}}$$

these rules correspond to the definition of $=$ for sets and to axiom of abstraction with φ stratified (in fact a kind of β-reduction).

Application to set-builders – Quine's NF

Take GCP and add:

$$(\Rightarrow=)\ \frac{a \in t, \Gamma \Rightarrow \Delta, a \in t' \qquad a \in t', \Gamma \Rightarrow \Delta, a \in t}{\Gamma \Rightarrow \Delta, t = t'}$$

$$(=\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, b \in t, b \in t' \qquad b \in t, b \in t', \Gamma \Rightarrow \Delta}{t = t', \Gamma \Rightarrow \Delta}$$

$$(Abs \Rightarrow)\ \frac{\varphi(t), \Gamma \Rightarrow \Delta}{t \in \{x : \varphi(x)\}, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow Abs)\ \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, t \in \{x : \varphi(x)\}}$$

these rules correspond to the definition of $=$ for sets and to axiom of abstraction with φ stratified (in fact a kind of β-reduction).
All rules are reducible for cut elimination (providing we treat $\in$ as having smaller degree than $=$).

Application to set-builders – Quine's NF

Application to set-builders – Quine's NF

Note that (*Ext*) is derivable in such a system:

$$
(\Rightarrow=)\;\dfrac{(Abs\Rightarrow Abs)\;\dfrac{\varphi(a),\Gamma\Rightarrow\Delta,\psi(a)}{a\in\{x:\varphi(x)\},\Gamma\Rightarrow\Delta,a\in\{x:\psi(x)\}}\qquad\dfrac{\psi(a),\Gamma\Rightarrow\Delta,\varphi(a)}{a\in\{x:\psi(x)\},\Gamma\Rightarrow\Delta,a\in\{x:\varphi(x)\}}}{\Gamma\Rightarrow\Delta,\{x:\varphi(x)\}=\{x:\psi(x)\}}
$$

Application to set-builders – Quine's NF

Note that (Ext) is derivable in such a system:

$$(\Rightarrow=)\ \dfrac{(Abs\Rightarrow Abs)\ \dfrac{\varphi(a),\Gamma\Rightarrow\Delta,\psi(a)}{a\in\{x:\varphi(x)\},\Gamma\Rightarrow\Delta,a\in\{x:\psi(x)\}} \qquad \dfrac{\psi(a),\Gamma\Rightarrow\Delta,\varphi(a)}{a\in\{x:\psi(x)\},\Gamma\Rightarrow\Delta,a\in\{x:\varphi(x)\}}}{\Gamma\Rightarrow\Delta,\{x:\varphi(x)\}=\{x:\psi(x)\}}$$

[similar for the case of $(ExtAV)$]

Application to set-builders – Quine's NF

Application to set-builders – Quine's NF

But what with LL? There are following cases:

1. $t = t', t \in t'' \Rightarrow t' \in t''$
2. $t = t', t'' \in t \Rightarrow t'' \in t'$
3. $t = t', t' = t'' \Rightarrow t = t''$

Application to set-builders – Quine's NF

But what with LL? There are following cases:

1. $t = t', t \in t'' \Rightarrow t' \in t''$
2. $t = t', t'' \in t \Rightarrow t'' \in t'$
3. $t = t', t' = t'' \Rightarrow t = t''$

With 3 no problem; derivable by $(\Rightarrow =), (= \Rightarrow)$, as other properties of $=$, including ref and sym.

Application to set-builders – Quine's NF

But what with LL? There are following cases:

1. $t = t', t \in t'' \Rightarrow t' \in t''$
2. $t = t', t'' \in t \Rightarrow t'' \in t'$
3. $t = t', t' = t'' \Rightarrow t = t''$

With 3 no problem; derivable by $(\Rightarrow=), (=\Rightarrow)$, as other properties of $=$, including ref and sym.

2 is provable by $(=\Rightarrow)$ but on condition that instead of b we can use any term t''; so even this case is problematic (subformula property, cut reduction).

Application to set-builders – Quine's NF

But what with LL? There are following cases:

1. $t = t', t \in t'' \Rightarrow t' \in t''$
2. $t = t', t'' \in t \Rightarrow t'' \in t'$
3. $t = t', t' = t'' \Rightarrow t = t''$

With 3 no problem; derivable by $(\Rightarrow =), (= \Rightarrow)$, as other properties of $=$, including ref and sym.

2 is provable by $(= \Rightarrow)$ but on condition that instead of b we can use any term t''; so even this case is problematic (subformula property, cut reduction).

1 even worse. To avoid troubles we could follow the general solution sketched above (with LL as two-premiss right-sided rule $(5 =)$)

Application to set-builders – Quine's NF

But what with LL? There are following cases:

1. $t = t', t \in t'' \Rightarrow t' \in t''$
2. $t = t', t'' \in t \Rightarrow t'' \in t'$
3. $t = t', t' = t'' \Rightarrow t = t''$

With 3 no problem; derivable by $(\Rightarrow=), (=\Rightarrow)$, as other properties of $=$, including ref and sym.

2 is provable by $(=\Rightarrow)$ but on condition that instead of b we can use any term t''; so even this case is problematic (subformula property, cut reduction).

1 even worse. To avoid troubles we could follow the general solution sketched above (with LL as two-premiss right-sided rule $(5 =)$) but it does not work too. 1 is not reducible with $(Abs \Rightarrow)$:

$$\dfrac{(LL)\ \dfrac{\Gamma \Rightarrow \Delta, t = t' \qquad \Gamma \Rightarrow \Delta, t' \in \{x : \varphi\}}{\Gamma \Rightarrow \Delta, t \in \{x : \varphi\}} \qquad \dfrac{\varphi(t), \Pi \Rightarrow \Sigma}{t \in \{x : \varphi\}, \Pi \Rightarrow \Sigma}\ (Abs \Rightarrow)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)$$

Application to set-builders

Application to set-builders

Quine's NF

Application to set-builders

Quine's NF

In the presence of $(Abs \Rightarrow)$ and $(\Rightarrow Abs)$ only 3-premiss version of (LL):

$$\frac{\Gamma \Rightarrow \Delta, t = t' \qquad \Gamma \Rightarrow \Delta, \varphi(t) \qquad \varphi(t'), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \ (8 =)$$

works, but it is not fully satisfactory (no subformula, no term property).

Application to set-builders

Application to set-builders

Quine's NF

Application to set-builders

Quine's NF

Perhaps the application of the approach 1 or 2.1 to $=$ works better?

Application to set-builders

Quine's NF

Perhaps the application of the approach 1 or 2.1 to $=$ works better?

Take some set of rules characterising $=$ but still no set is reducible with either $(Abs \Rightarrow)$ or $(\Rightarrow Abs)$ except 3-premiss version of (LL).

Application to set-builders

Quine's NF

Perhaps the application of the approach 1 or 2.1 to $=$ works better?

Take some set of rules characterising $=$ but still no set is reducible with either ($Abs \Rightarrow$) or ($\Rightarrow Abs$) except 3-premiss version of (LL).

Moreover we need a rule-characterisation of ($ExtAx$); ($\Rightarrow=$) works (in particular with 3-premiss LL).

Application to set-builders

Quine's NF

Perhaps the application of the approach 1 or 2.1 to $=$ works better?

Take some set of rules characterising $=$ but still no set is reducible with either $(Abs \Rightarrow)$ or $(\Rightarrow Abs)$ except 3-premiss version of (LL).

Moreover we need a rule-characterisation of $(ExtAx)$; $(\Rightarrow=)$ works (in particular with 3-premiss LL).

No advantage over the approach 2.2 based on the original Quine's formulation.

Funded by the European Union (ERC, ExtenDD, project number: 101054714). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

European Research Council
Established by the European Commission