Strict form of stratified set theories NF and NFU in the setting of sequent calculus

Andrzej Indrzejczak

Department of Logic, University of Lodz

ExtenDD Seminar, Łódź, October 9, 2024

Plan of the Talk

Plan of the Talk

1. NF and NFU.

Plan of the Talk

1. NF and NFU.
2. Language matters.

Plan of the Talk

1. NF and NFU.
2. Language matters.
3. Axiomatization.

Plan of the Talk

1. NF and NFU.
2. Language matters.
3. Axiomatization.
4. Sequent Calculus GSNF and its properties.

Plan of the Talk

1. NF and NFU.
2. Language matters.
3. Axiomatization.
4. Sequent Calculus GSNF and its properties.
5. Sequent Calculus GSNFU and its properties.

What is NF and NFU?

What is NF and NFU?

NF (New Foundations): the approach to set theory introduced by Quine in 1937 and based on the notion of stratification of formulae. (still problems with relative consistency proof)

What is NF and NFU?

NF (New Foundations): the approach to set theory introduced by Quine in 1937 and based on the notion of stratification of formulae. (still problems with relative consistency proof)

NFU (NF with urelements): subtheory of NF introduced by Jensen in 1969 and proved consistent.

What is NF and NFU?

NF (New Foundations): the approach to set theory introduced by Quine in 1937 and based on the notion of stratification of formulae. (still problems with relative consistency proof)

NFU (NF with urelements): subtheory of NF introduced by Jensen in 1969 and proved consistent.

Language of NF (NFU) may be with only $\in$ as primitive predicate ($=$ defined) or with both predicates primitive.

What is NF and NFU?

NF (New Foundations): the approach to set theory introduced by Quine in 1937 and based on the notion of stratification of formulae. (still problems with relative consistency proof)

NFU (NF with urelements): subtheory of NF introduced by Jensen in 1969 and proved consistent.

Language of NF (NFU) may be with only $\in$ as primitive predicate ($=$ defined) or with both predicates primitive.

It may be with set abstracts as primitive or as defined.

Preliminary issues concerning $=$; possible choices:

Preliminary issues concerning $=$; possible choices:

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

Preliminary issues concerning =; possible choices:

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

Preliminary issues concerning $=$; possible choices:

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

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

Preliminary issues concerning $=$; possible choices:

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

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

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

Preliminary issues concerning $=$; possible choices:

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

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

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

Preliminary issues concerning $=$; possible choices:

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

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

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

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

Preliminary issues concerning $=$; possible choices:

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

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

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

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

The Language:

The Language:

Our choice

The Language:

Our choice

Language with $=$ and $\in$ primitive.

The Language:

Our choice

Language with $=$ and $\in$ primitive.

Hence we assume that for $=$ it holds:
LL $t_1 = t_2 \wedge \varphi[x/t_1] \rightarrow \varphi[x/t_2]$

Our choice

Language with $=$ and $\in$ primitive.

Hence we assume that for $=$ it holds:
$LL \ t_1 = t_2 \wedge \varphi[x/t_1] \rightarrow \varphi[x/t_2]$

Moreover, we treat set abstracts $\{y : \varphi(y)\}$ as primitive notion.

The Language:

Our choice

Language with $=$ and $\in$ primitive.

Hence we assume that for $=$ it holds:
$LL\ t_1 = t_2 \wedge \varphi[x/t_1] \rightarrow \varphi[x/t_2]$

Moreover, we treat set abstracts $\{y : \varphi(y)\}$ as primitive notion.

Terms $t, t', t_1, \ldots$ are either variables x, y, z (bound), a, b, c (free) or set abstracts s.

Axiomatic form of NF (NFU)

Axiomatic form of NF (NFU)

Arbitrary adequate axiomatization of FOLI (first-order logic with identity), plus:

Axiomatic form of NF (NFU)

Arbitrary adequate axiomatization of FOLI (first-order logic with identity), plus:

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

COM: $\forall x(x \in \{y : \varphi(y)\} \leftrightarrow \varphi[y/x])$, where φ is stratified.

EXT': $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \{x : \varphi(x)\} = \{x : \psi(x)\}$

AV: $\{x : \varphi(x)\} = \{y : \varphi(y)\}$

Axiomatic form of NF (NFU)

Arbitrary adequate axiomatization of FOLI (first-order logic with identity), plus:

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

COM: $\forall x(x \in \{y : \varphi(y)\} \leftrightarrow \varphi[y/x])$, where φ is stratified.

EXT': $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \{x : \varphi(x)\} = \{x : \psi(x)\}$

AV: $\{x : \varphi(x)\} = \{y : \varphi(y)\}$

The condition of stratification may be defined roughly as follows: it is possible to define a mapping from variables of φ into integers in a way that for all atoms we have $i \in i + 1$ and $i = i$.

Axiomatic form of NF (NFU)

Arbitrary adequate axiomatization of FOLI (first-order logic with identity), plus:

EXT: $\forall xy(\forall z(z \in x \leftrightarrow z \in y) \to x = y)$

COM: $\forall x(x \in \{y : \varphi(y)\} \leftrightarrow \varphi[y/x])$, where φ is stratified.

EXT′: $\forall x(\varphi(x) \leftrightarrow \psi(x)) \to \{x : \varphi(x)\} = \{x : \psi(x)\}$

AV: $\{x : \varphi(x)\} = \{y : \varphi(y)\}$

The condition of stratification may be defined roughly as follows: it is possible to define a mapping from variables of φ into integers in a way that for all atoms we have $i \in i + 1$ and $i = i$.

NFU replaces *EXT* with:

EXTU: $\forall xy(\exists z(z \in x) \wedge \forall z(z \in x \leftrightarrow z \in y) \to x = y)$

Strict form of NF(NFU) called SNF (SNFU)

Strict form of NF(NFU) called SNF (SNFU)

NF formulated in restricted language in which only stratified φ is admitted in abstracts, i.e. we have a formation clause:

If φ is stratified, then $\{x : \varphi\}$ is a term.

Strict form of NF(NFU) called SNF (SNFU)

NF formulated in restricted language in which only stratified φ is admitted in abstracts, i.e. we have a formation clause:

If φ is stratified, then $\{x : \varphi\}$ is a term.

Differences:

Strict form of NF(NFU) called SNF (SNFU)

NF formulated in restricted language in which only stratified φ is admitted in abstracts, i.e. we have a formation clause:

If φ is stratified, then $\{x : \varphi\}$ is a term.

Differences:

$\vdash \{x : x \in x\} = \{x : x \in x\}$ is a thesis of NF (provable by EXT) but not of SNF, in fact it is not even well-formed formula since $\{x : x \in x\}$ is not a term in the language of SNF.

Strict form of NF(NFU) called SNF (SNFU)

NF formulated in restricted language in which only stratified φ is admitted in abstracts, i.e. we have a formation clause:

If φ is stratified, then $\{x : \varphi\}$ is a term.

Differences:

$\vdash \{x : x \in x\} = \{x : x \in x\}$ is a thesis of NF (provable by EXT) but not of SNF, in fact it is not even well-formed formula since $\{x : x \in x\}$ is not a term in the language of SNF.

EXT' $\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \{x : \varphi(x)\} = \{x : \psi(x)\}$ can be dropped from axiomatization since it is a thesis of SNF but restricted to stratified φ, ψ (is provable in SNF by EXT and COM, in contrast to NF).

Constructing Sequent Calculi for term-forming operators (set-abstracts in particular):

Constructing Sequent Calculi for term-forming operators (set-abstracts in particular):

The main problems to overcome:

Constructing Sequent Calculi for term-forming operators (set-abstracts in particular):

The main problems to overcome:

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

Constructing Sequent Calculi for term-forming operators (set-abstracts in particular):

The main problems to overcome:

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-formulae generated by means of rules for set abstracts?

The basic system GC for pure CFOL:

The basic system GC for pure CFOL:

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

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

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

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

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

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

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

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

The basic system GC for pure CFOL:

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

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

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

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

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

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

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

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

Note that set abstracts are not allowed as instances of variables in quantifier rules! (problem 1)

Variants:

Variants:

1. GI: at most one formula in the succedent, no $(\Rightarrow C)$ and alternative rules for $\vee, \leftrightarrow$:

Variants:

1. GI: at most one formula in the succedent, no $(\Rightarrow C)$ and alternative rules for $\vee, \leftrightarrow$:

$$(\Rightarrow\vee 1)\ \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \qquad (\Rightarrow\vee 2)\ \frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$$

$$(\leftrightarrow\Rightarrow 1)\ \frac{\Gamma\Rightarrow \Delta, \varphi \qquad \psi, \Gamma\Rightarrow \Delta}{\varphi \leftrightarrow \psi, \Gamma\Rightarrow \Delta} \qquad (\leftrightarrow\Rightarrow 2)\ \frac{\Gamma\Rightarrow \Delta, \psi \qquad \varphi, \Gamma\Rightarrow \Delta}{\varphi \leftrightarrow \psi, \Gamma\Rightarrow \Delta}$$

Variants:

1. GI: at most one formula in the succedent, no $(\Rightarrow C)$ and alternative rules for $\vee, \leftrightarrow$:

$$(\Rightarrow\vee 1)\ \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \qquad (\Rightarrow\vee 2)\ \frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$$

$$(\leftrightarrow\Rightarrow 1)\ \frac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \leftrightarrow \psi, \Gamma \Rightarrow \Delta} \qquad (\leftrightarrow\Rightarrow 2)\ \frac{\Gamma \Rightarrow \Delta, \psi \qquad \varphi, \Gamma \Rightarrow \Delta}{\varphi \leftrightarrow \psi, \Gamma \Rightarrow \Delta}$$

Note! all results obtained below for classical variant hold also for intuitionistic one.

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.

Rules for = (Rule-maker theorem Indrzejczak 2013)

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

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

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

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

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

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

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

Sequent calculus GSNF = G+:

Sequent calculus GSNF = G+:

$$(R)\ \frac{b = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad (N)\ \frac{a = s, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad (E)\ \frac{\Gamma \Rightarrow \Delta, t = b \qquad \Gamma \Rightarrow \Delta, t = c}{\Gamma \Rightarrow \Delta, b = c}$$

$$(\in 1)\ \frac{\Gamma \Rightarrow \Delta, t_1 = t_2 \qquad \Gamma \Rightarrow \Delta, t_1 \in t_3}{\Gamma \Rightarrow \Delta, t_2 \in t_3} \quad (\in 2)\ \frac{\Gamma \Rightarrow \Delta, t_1 = t_2 \qquad \Gamma \Rightarrow \Delta, t_3 \in t_1}{\Gamma \Rightarrow \Delta, t_3 \in t_2}$$

$$(:\Rightarrow 1)\ \frac{\Gamma \Rightarrow \Delta, t[b] \qquad \varphi[x/b], \Gamma \Rightarrow \Delta}{t \approx \{x : \varphi\}, \Gamma \Rightarrow \Delta} \quad (:\Rightarrow 2)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b] \qquad t[b], \Gamma \Rightarrow \Delta}{t \approx \{x : \varphi\}, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow:)\ \frac{t[a], \Gamma \Rightarrow \Delta, \varphi[x/a] \qquad \varphi[x/a], \Gamma \Rightarrow \Delta, t[a]}{\Gamma \Rightarrow \Delta, t \approx \{x : \varphi\}}$$

where: a is fresh in $(N), (\Rightarrow:)$, t is arbitrary term, s is arbitrary set abstract, $t[b]$ is either $b \in t$, if t is a parameter or $\varphi[x/b]$, if $t := \{x : \varphi\}$.

Sequent calculus GSNF = G+:

$$(R)\ \frac{b = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad (N)\ \frac{a = s, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad (E)\ \frac{\Gamma \Rightarrow \Delta, t = b \qquad \Gamma \Rightarrow \Delta, t = c}{\Gamma \Rightarrow \Delta, b = c}$$

$$(\in 1)\ \frac{\Gamma \Rightarrow \Delta, t_1 = t_2 \qquad \Gamma \Rightarrow \Delta, t_1 \in t_3}{\Gamma \Rightarrow \Delta, t_2 \in t_3} \quad (\in 2)\ \frac{\Gamma \Rightarrow \Delta, t_1 = t_2 \qquad \Gamma \Rightarrow \Delta, t_3 \in t_1}{\Gamma \Rightarrow \Delta, t_3 \in t_2}$$

$$(:\Rightarrow 1)\ \frac{\Gamma \Rightarrow \Delta, t[b] \qquad \varphi[x/b], \Gamma \Rightarrow \Delta}{t \approx \{x : \varphi\}, \Gamma \Rightarrow \Delta} \quad (:\Rightarrow 2)\ \frac{\Gamma \Rightarrow \Delta, \varphi[x/b] \qquad t[b], \Gamma \Rightarrow \Delta}{t \approx \{x : \varphi\}, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow:)\ \frac{t[a], \Gamma \Rightarrow \Delta, \varphi[x/a] \qquad \varphi[x/a], \Gamma \Rightarrow \Delta, t[a]}{\Gamma \Rightarrow \Delta, t \approx \{x : \varphi\}}$$

where: a is fresh in $(N), (\Rightarrow:)$, t is arbitrary term, s is arbitrary set abstract, $t[b]$ is either $b \in t$, if t is a parameter or $\varphi[x/b]$, if $t := \{x : \varphi\}$.

Since $\approx$ means that either $c = t$ or $t = c$ we have in fact six rules for abstract operator in two last lines (even 12 if we distinguish variants with t being parameter or set abstract)

Adequacy of GSNF

Adequacy of GSNF

Proving derivability of rules:

Proving derivability of rules:

(R) is obtained by cut with $\Rightarrow b = b$.

Adequacy of GSNF

Proving derivability of rules:

(R) is obtained by cut with $\Rightarrow b = b$.

(N):

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

Adequacy of GSNF

Proving derivability of rules:

(R) is obtained by cut with $\Rightarrow b = b$.

(N):

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

$(E), (\in 1), (\in 2)$ are all derivable by two cuts with suitable instances of $t = t', \varphi[x/t] \Rightarrow \varphi[x/t']$ and contractions.

Adequacy of GSNF

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Consider a variant with t a parameter c, so $t[b] := b \in c$.

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Consider a variant with t a parameter c, so $t[b] := b \in c$.

$$\dfrac{\dfrac{\dfrac{\dfrac{a=\{x:\varphi\}, b\in c \Rightarrow b\in\{x:\varphi\} \qquad b\in\{x:\varphi\}\Rightarrow\varphi[x/b]}{a=\{x:\varphi\}, b\in c\Rightarrow\varphi[x/b]}}{a=\{x:\varphi\}\Rightarrow b\in c\to\varphi[x/b]}(\Rightarrow\to)}{a=\{x:\varphi\}\Rightarrow\forall x(x\in c\to\varphi)}(\Rightarrow\forall) \qquad \dfrac{\dfrac{\Gamma\Rightarrow\Delta, b\in c \qquad \varphi[x/b],\Gamma\Rightarrow\Delta}{b\in c\to\varphi[x/b],\Gamma\Rightarrow\Delta}(\to\Rightarrow)}{\forall x(x\in c\to\varphi),\Gamma\Rightarrow\Delta}(\forall\Rightarrow)}{a=\{x:\varphi\},\Gamma\Rightarrow\Delta}(Cut)$$

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Consider a variant with t a parameter c, so $t[b] := b \in c$.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{a = \{x : \varphi\}, b \in c \Rightarrow b \in \{x : \varphi\} \qquad b \in \{x : \varphi\} \Rightarrow \varphi[x/b]}{a = \{x : \varphi\}, b \in c \Rightarrow \varphi[x/b]}
    }{a = \{x : \varphi\} \Rightarrow b \in c \to \varphi[x/b]}\,(\Rightarrow\to)
  }{a = \{x : \varphi\} \Rightarrow \forall x(x \in c \to \varphi)}\,(\Rightarrow\forall)
  \qquad
  \dfrac{
    \dfrac{\Gamma \Rightarrow \Delta, b \in c \qquad \varphi[x/b], \Gamma \Rightarrow \Delta}{b \in c \to \varphi[x/b], \Gamma \Rightarrow \Delta}\,(\to
  }{\forall x(x \in c \to \varphi), \Gamma \Rightarrow \Delta}\,(\forall\Rightarrow)
}{a = \{x : \varphi\}, \Gamma \Rightarrow \Delta}\,(Cut)
$$

Proof for $t := \{x : \psi\}$ similar but requiring additionally cut with $\psi[x/b] \Rightarrow b \in \{x : \psi\}$.

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Consider a variant with t a parameter c, so $t[b] := b \in c$.

$$\dfrac{\dfrac{\dfrac{\dfrac{a=\{x:\varphi\}, b\in c \Rightarrow b\in\{x:\varphi\} \qquad b\in\{x:\varphi\}\Rightarrow \varphi[x/b]}{a=\{x:\varphi\}, b\in c\Rightarrow \varphi[x/b]}}{a=\{x:\varphi\}\Rightarrow b\in c\to\varphi[x/b]}\,(\Rightarrow\to)}{a=\{x:\varphi\}\Rightarrow \forall x(x\in c\to\varphi)}\,(\Rightarrow\forall) \qquad \dfrac{\dfrac{\Gamma\Rightarrow\Delta, b\in c \qquad \varphi[x/b],\Gamma\Rightarrow\Delta}{b\in c\to\varphi[x/b],\Gamma\Rightarrow\Delta}\,(\to}{\forall x(x\in c\to\varphi),\Gamma\Rightarrow\Delta}\,(\forall\Rightarrow)}{a=\{x:\varphi\},\Gamma\Rightarrow\Delta}\,(Cut)$$

Proof for $t := \{x : \psi\}$ similar but requiring additionally cut with $\psi[x/b] \Rightarrow b \in \{x : \psi\}$.

For $\{x : \varphi\} = t$ by symmetric variant $t' = t, \varphi[x/t] \Rightarrow \varphi[x/t']$.

Adequacy of GSNF

Proving derivability of rules: $(:\Rightarrow 1)$

Consider a variant with t a parameter c, so $t[b] := b \in c$.

$$
\dfrac{
(\Rightarrow\to)\,\dfrac{\dfrac{a = \{x : \varphi\}, b \in c \Rightarrow b \in \{x : \varphi\} \qquad b \in \{x : \varphi\} \Rightarrow \varphi[x/b]}{a = \{x : \varphi\}, b \in c \Rightarrow \varphi[x/b]}}{(\Rightarrow\forall)\,\dfrac{a = \{x : \varphi\} \Rightarrow b \in c \to \varphi[x/b]}{a = \{x : \varphi\} \Rightarrow \forall x(x \in c \to \varphi)}}
\qquad
\dfrac{\dfrac{\Gamma \Rightarrow \Delta, b \in c \qquad \varphi[x/b], \Gamma \Rightarrow \Delta}{b \in c \to \varphi[x/b], \Gamma \Rightarrow \Delta}\,(\to}{\forall x(x \in c \to \varphi), \Gamma \Rightarrow \Delta}\,(\forall\Rightarrow)
}{a = \{x : \varphi\}, \Gamma \Rightarrow \Delta}\,(Cut)
$$

Proof for $t := \{x : \psi\}$ similar but requiring additionally cut with $\psi[x/b] \Rightarrow b \in \{x : \psi\}$.

For $\{x : \varphi\} = t$ by symmetric variant $t' = t, \varphi[x/t] \Rightarrow \varphi[x/t']$.

For $(:\Rightarrow 2)$ dual proofs.

Adequacy of GSNF

Adequacy of GSNF

Proving derivability of rules: ($\Rightarrow$:)

Adequacy of GSNF

Proving derivability of rules: ($\Rightarrow$:)

Consider a variant with t a parameter c, so $t[a] := a \in c$.

Adequacy of GSNF

Proving derivability of rules: $(\Rightarrow:)$

Consider a variant with t a parameter c, so $t[a] := a \in c$.

$$\cfrac{\cfrac{a \in c, \Gamma \Rightarrow \Delta, \varphi[x/a] \qquad \varphi[x/a] \Rightarrow a \in \{x : \varphi\}}{a \in c, \Gamma \Rightarrow \Delta, a \in \{x : \varphi\}} \qquad \cfrac{a \in \{x : \varphi\} \Rightarrow \varphi[x/a] \qquad \varphi[x/a], \Gamma \Rightarrow \Delta, a \in c}{a \in \{x : \varphi\}, \Gamma \Rightarrow \Delta, a \in c}}{\cfrac{\Gamma \Rightarrow \Delta, a \in c \leftrightarrow a \in \{x : \varphi\}}{\Gamma \Rightarrow \Delta, \forall x(x \in c \leftrightarrow x \in \{x : \varphi\})}(\Rightarrow \forall)}(\Rightarrow\leftrightarrow)$$

By cut with $\forall x(x \in c \leftrightarrow x \in \{x : \varphi\}) \Rightarrow c = \{x : \varphi\}$ we obtain $\Gamma \Rightarrow \Delta, c = \{x : \varphi\}$.

Adequacy of GSNF

Proving derivability of rules: $(\Rightarrow:)$

Consider a variant with t a parameter c, so $t[a] := a \in c$.

$$
\dfrac{\dfrac{a \in c, \Gamma \Rightarrow \Delta, \varphi[x/a] \qquad \varphi[x/a] \Rightarrow a \in \{x : \varphi\}}{a \in c, \Gamma \Rightarrow \Delta, a \in \{x : \varphi\}} \quad \dfrac{a \in \{x : \varphi\} \Rightarrow \varphi[x/a] \qquad \varphi[x/a], \Gamma \Rightarrow \Delta, a \in c}{a \in \{x : \varphi\}, \Gamma \Rightarrow \Delta, a \in c}}{\dfrac{\Gamma \Rightarrow \Delta, a \in c \leftrightarrow a \in \{x : \varphi\}}{\Gamma \Rightarrow \Delta, \forall x(x \in c \leftrightarrow x \in \{x : \varphi\})}\ (\Rightarrow \forall)}\ (\Rightarrow\leftrightarrow)
$$

By cut with $\forall x(x \in c \leftrightarrow x \in \{x : \varphi\}) \Rightarrow c = \{x : \varphi\}$ we obtain $\Gamma \Rightarrow \Delta, c = \{x : \varphi\}$.

The remaining variants of $(\Rightarrow:)$ proven similarly.

Adequacy of GSNF

Adequacy of GSNF

Proving LL:

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Proof by induction on the complexity of φ. In particular, one has to prove for the basis:

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Proof by induction on the complexity of φ. In particular, one has to prove for the basis:

1. $t_1 = t_2, t_1 = t_3 \Rightarrow t_2 = t_3$
2. $t_1 = t_2, t_1 \in t_3 \Rightarrow t_2 \in t_3$
3. $t_1 = t_2, t_3 \in t_1 \Rightarrow t_3 \in t_2$

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Proof by induction on the complexity of φ. In particular, one has to prove for the basis:

1. $t_1 = t_2, t_1 = t_3 \Rightarrow t_2 = t_3$
2. $t_1 = t_2, t_1 \in t_3 \Rightarrow t_2 \in t_3$
3. $t_1 = t_2, t_3 \in t_1 \Rightarrow t_3 \in t_2$

Proofs of cases 2 and 3, as well as $\Rightarrow s = s$ and $t_1 = t_2 \Rightarrow t_2 = t_1$ are immediate.

Adequacy of GSNF

Adequacy of GSNF

Proving LL:

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Case 1 $t_1 = t_2, t_1 = t_3 \Rightarrow t_2 = t_3$ requires 8 subcases to derive (t_i being either parameter or set abstract).

Adequacy of GSNF

Proving LL:

Lemma

$GNSF \vdash t_1 = t_2, \varphi[x/t_1] \Rightarrow \varphi[x/t_2]$

Case 1 $t_1 = t_2, t_1 = t_3 \Rightarrow t_2 = t_3$ requires 8 subcases to derive (t_i being either parameter or set abstract).

1. $a = b, a = c \Rightarrow b = c$
2. $\{x : \varphi\} = b, \{x : \varphi\} = c \Rightarrow b = c$
3. $a = \{x : \varphi\}, a = c \Rightarrow \{x : \varphi\} = c$
4. $a = b, a = \{x : \varphi\} \Rightarrow b = \{x : \varphi\}$
5. $\{x : \varphi\} = \{x : \psi\}, \{x : \varphi\} = c \Rightarrow \{x : \psi\} = c$
6. $a = \{x : \varphi\}, a = \{x : \psi\} \Rightarrow \{x : \varphi\} = \{x : \psi\}$
7. $\{x : \psi\} = b, \{x : \psi\} = \{x : \varphi\} \Rightarrow b = \{x : \varphi\}$
8. $\{x : \varphi\} = \{x : \psi\}, \{x : \varphi\} = \{x : \chi\} \Rightarrow \{x : \psi\} = \{x : \chi\}$

Adequacy of GSNF

Adequacy of GSNF

Proving LL, subcase 1.3:

$$\dfrac{D \qquad \dfrac{\varphi[x/b] \Rightarrow \varphi[x/b] \qquad \dfrac{a = c \Rightarrow a = c \qquad b \in a \Rightarrow b \in a}{b \in a, a = c \Rightarrow b \in c}\,(:\Rightarrow 1)}{\varphi[x/b], a = \{x : \varphi\}, a = c \Rightarrow b \in c}\,(:\Rightarrow 2)}{a = \{x : \varphi\}, a = c \Rightarrow \{x : \varphi\} = c}\,(\Rightarrow:)$$

where D is:

$$\dfrac{\dfrac{\dfrac{a = c \Rightarrow a = c \qquad \dfrac{a = a \Rightarrow a = a}{\Rightarrow a = a}\,(R)}{a = c \Rightarrow c = a}\,(E) \qquad b \in c \Rightarrow b \in c}{c = a, b \in c \Rightarrow b \in a}\,(\in 2) \qquad \varphi[x/b] \Rightarrow \varphi[x/b]}{b \in c, a = \{x : \varphi\}, a = c \Rightarrow \varphi[x/b]}\,(:\Rightarrow 1)$$

Adequacy of GSNF

Adequacy of GSNF

Proving COM:

Adequacy of GSNF

Proving COM:

$$
(\Rightarrow \forall)\ \dfrac{(\Rightarrow\leftrightarrow)\ \dfrac{(N)\ \dfrac{(\in 2)\ \dfrac{b = \{x : \varphi\} \Rightarrow \{x : \varphi\} = b \qquad a \in \{x : \varphi\} \Rightarrow a \in \{x : \varphi\}}{b = \{x : \varphi\}, a \in \{x : \varphi\} \Rightarrow a \in b}}{a \in \{x : \varphi\} \Rightarrow a \in b} \qquad D}{\Rightarrow a \in \{x : \varphi\} \leftrightarrow \varphi[y/a]}}{\Rightarrow \forall x (x \in \{y : \varphi(y)\} \leftrightarrow \varphi[y/x])}
$$

where the leftmost leaf is easily provable by $(\Rightarrow:), (:\Rightarrow 2)$ and D is:

$$
\dfrac{\dfrac{\varphi[x/a] \Rightarrow \varphi[x/a] \qquad \dfrac{b = \{x : \varphi\} \Rightarrow b = \{x : \varphi\} \qquad a \in b \Rightarrow a \in b}{a \in b, b = \{x : \varphi\} \Rightarrow a \in \{x : \varphi\}}\ (\in 2)}{b = \{x : \varphi\}, \varphi[x/a] \Rightarrow a \in \{x : \varphi\}}\ (:\Rightarrow 2)}{\varphi[x/a] \Rightarrow a \in \{x : \varphi\}}\ (N)
$$

Adequacy of GSNF

Adequacy of GSNF

Proving EXT:

Adequacy of GSNF

Proving EXT:

$$
(\Rightarrow\forall)\,\dfrac{(\Rightarrow\rightarrow)\,\dfrac{\dfrac{D \qquad \dfrac{c \in b \Rightarrow c \in b \qquad c \in b \Rightarrow c \in b}{\Rightarrow \{x : x \in b\} = b}\,(\Rightarrow:)}{\forall z(z \in a \leftrightarrow z \in b) \Rightarrow a = b}\,(E)}{\Rightarrow \forall z(z \in a \leftrightarrow z \in b) \rightarrow a = b}}{\Rightarrow \forall xy(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)}
$$

where D is:

$$
(\Rightarrow:)\,\dfrac{(\forall\Rightarrow)\,\dfrac{c \in a \leftrightarrow c \in b, c \in a \Rightarrow c \in b}{\forall z(z \in a \leftrightarrow z \in b), c \in a \Rightarrow c \in b} \qquad \dfrac{c \in a \leftrightarrow c \in b, c \in b \Rightarrow c \in a}{\forall z(z \in a \leftrightarrow z \in b), c \in b \Rightarrow c \in a}}{\forall z(z \in a \leftrightarrow z \in b) \Rightarrow \{x : x \in b\} = a}
$$

Summary of GSNF

1. It is an adequate formalisation of SNF.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.
4. All rules for connectives, quantifiers and set abstracts are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.
4. All rules for connectives, quantifiers and set abstracts are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. All rules, except cut satisfy the generalized subformula property closed for the addition of atomic formulae.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.
4. All rules for connectives, quantifiers and set abstracts are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. All rules, except cut satisfy the generalized subformula property closed for the addition of atomic formulae.
6. The remaining rules for $=$ and $\in$ are all one-sided (active formulae in the succedents only), hence reduction of cut-height holds.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.
4. All rules for connectives, quantifiers and set abstracts are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. All rules, except cut satisfy the generalized subformula property closed for the addition of atomic formulae.
6. The remaining rules for $=$ and $\in$ are all one-sided (active formulae in the succedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 2, 4 and 6.

Summary of GSNF

1. It is an adequate formalisation of SNF.
2. Substitution theorem: [if $\vdash_k \Gamma \Rightarrow \Delta$, then $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$] holds.
3. Rules for set abstracts $((:\Rightarrow 1), (:\Rightarrow 2), (\Rightarrow:))$ are explicit, separate symmetric and satisfy the subformula property which are usual requirements for well-behaved SC rules.
4. All rules for connectives, quantifiers and set abstracts are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. All rules, except cut satisfy the generalized subformula property closed for the addition of atomic formulae.
6. The remaining rules for $=$ and $\in$ are all one-sided (active formulae in the succedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 2, 4 and 6.
8. The system is analytic in the generalised sense due to 3 and 4.

GSNFU

GSNFU

$$(R)\ \frac{b=b,\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}\quad (N)\ \frac{a=s,\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}\quad (AV)\ \frac{\{x:\varphi(x)\}=\{y:\varphi(y)\},\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}$$

$$(E1)\ \frac{b=c,\Gamma\Rightarrow\Delta}{a=b,a=c,\Gamma\Rightarrow\Delta}\quad (L1)\ \frac{b\in c,\Gamma\Rightarrow\Delta}{a=b,a\in c,\Gamma\Rightarrow\Delta}\quad (L2)\ \frac{c\in b,\Gamma\Rightarrow\Delta}{a=b,c\in a,\Gamma\Rightarrow\Delta}$$

$$(E2)\ \frac{\Gamma\Rightarrow\Delta,s=a,\qquad \Pi\Rightarrow\Sigma,s=b\qquad a=b,\Theta\Rightarrow\Xi}{\Gamma,\Pi,\Theta\Rightarrow\Delta,\Sigma,\Xi}\quad (\in\Rightarrow 1)\ \frac{\varphi[x/b],\Gamma\Rightarrow\Delta}{b\in\{x:\varphi\},\Gamma\Rightarrow\Delta}$$

$$(E3)\ \frac{\Gamma\Rightarrow\Delta,t=t'\qquad \Pi\Rightarrow\Sigma,t=s\qquad t'\approx s,\Theta\Rightarrow\Xi}{\Gamma,\Pi,\Theta\Rightarrow\Delta,\Sigma,\Xi}\quad (\Rightarrow\in 1)\ \frac{\Gamma\Rightarrow\Delta,\varphi[x/b]}{\Gamma\Rightarrow\Delta,b\in\{x:\varphi\}}$$

$$(\in\Rightarrow 2)\ \frac{a=\{x:\varphi\},a\in t,\Gamma\Rightarrow\Delta}{\{x:\varphi\}\in t,\Gamma\Rightarrow\Delta}\quad (\Rightarrow\in 2)\ \frac{\Gamma\Rightarrow\Delta,b=\{x:\varphi\}\qquad \Gamma\Rightarrow\Delta,b\in t}{\Gamma\Rightarrow\Delta,\{x:\varphi\}\in t}$$

$$(:\Rightarrow 1)\ \frac{\Gamma\Rightarrow\Delta,t[b]\qquad \varphi[x/b],\Gamma\Rightarrow\Delta}{t\approx\{x:\varphi\},\Gamma\Rightarrow\Delta}\quad (:\Rightarrow 2)\ \frac{\Gamma\Rightarrow\Delta,\varphi[x/b]\qquad t[b],\Gamma\Rightarrow\Delta}{t\approx\{x:\varphi\},\Gamma\Rightarrow\Delta}$$

$$(\Rightarrow:)\ \frac{t[a],\Gamma\Rightarrow\Delta,\varphi[x/a]\qquad \varphi[x/a],\Gamma\Rightarrow\Delta,t[a]}{b\in t',\Gamma\Rightarrow\Delta,t\approx\{x:\varphi\}},$$

where: a is fresh in $(N), (\Rightarrow:), (\in\Rightarrow 2)$, $t[b]$ is either $b\in t$, if t is a parameter or $\psi[x/b]$, if $t := \{x:\psi\}$, t' in $(\Rightarrow:)$ is the rightmost argument of $t\approx\{x:\varphi\}$.

GSNFU - some results:

GSNFU - some results:

Theorem (Cut Elimination)

Every proof in GSNFU can be transformed into cut-free proof.

GSNFU - some results:

Theorem (Cut Elimination)

Every proof in GSNFU can be transformed into cut-free proof.

Lemma (Up-permutability of (S) and $(E3)$)

Let $\Gamma \Rightarrow \Delta, t = s$ be the left premiss of (S) or $(E3)$ and the conclusion of arbitrary rule r, then the application of the rules can be permuted.

GSNFU - some results:

Theorem (Cut Elimination)

Every proof in GSNFU can be transformed into cut-free proof.

Lemma (Up-permutability of (S) and $(E3)$)

Let $\Gamma \Rightarrow \Delta, t = s$ be the left premiss of (S) or $(E3)$ and the conclusion of arbitrary rule r, then the application of the rules can be permuted.

Lemma

Every proof can be transformed into a proof where $t = s$ in the left premiss of (S) or $(E3)$ is a principal formula.

GSNFU - some results:

Theorem (Cut Elimination)

Every proof in GSNFU can be transformed into cut-free proof.

Lemma (Up-permutability of (S) and $(E3)$)

Let $\Gamma \Rightarrow \Delta, t = s$ be the left premiss of (S) or $(E3)$ and the conclusion of arbitrary rule r, then the application of the rules can be permuted.

Lemma

Every proof can be transformed into a proof where $t = s$ in the left premiss of (S) or $(E3)$ is a principal formula.

Lemma

Every application of $(R), (AV), (N), (S)$ or $(E3)$ has a nonempty conclusion.

GSNFU - some results:

Theorem (Cut Elimination)

Every proof in GSNFU can be transformed into cut-free proof.

Lemma (Up-permutability of (S) and $(E3)$)

Let $\Gamma \Rightarrow \Delta, t = s$ be the left premiss of (S) or $(E3)$ and the conclusion of arbitrary rule r, then the application of the rules can be permuted.

Lemma

Every proof can be transformed into a proof where $t = s$ in the left premiss of (S) or $(E3)$ is a principal formula.

Lemma

Every application of $(R), (AV), (N), (S)$ or $(E3)$ has a nonempty conclusion.

Theorem (Consistency)

No proof in GSNFU ends with an empty sequent.

Open problems, further research:

Open problems, further research:

- Formalise in similar way NF and NFU (or show that they are equivalent to SNF, SNFU).

Open problems, further research:

- Formalise in similar way NF and NFU (or show that they are equivalent to SNF, SNFU).
- Find well-behaved sequent calculi for other approaches to set theory.

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.