When Epsilon Meets Lambda: Extended Leśniewski's Ontology

Andrzej Indrzejczak

Department of Logic, University of Lodz

ExtenDD Seminar, Łódź, March 20, 2024

OUTLINE OF THE TALK

OUTLINE OF THE TALK

1. The Ontology of Leśniewski (LO).

1. The Ontology of Leśniewski (LO).
2. Sequent Calculus GO for LO.

OUTLINE OF THE TALK

1. The Ontology of Leśniewski (LO).
2. Sequent Calculus GO for LO.
3. Sequent Calculus GOP for LO with predicates.

OUTLINE OF THE TALK

1. The Ontology of Leśniewski (LO).
2. Sequent Calculus GO for LO.
3. Sequent Calculus GOP for LO with predicates.
4. Three variants of Extended LO (ELO).

OUTLINE OF THE TALK

1. The Ontology of Leśniewski (LO).
2. Sequent Calculus GO for LO.
3. Sequent Calculus GOP for LO with predicates.
4. Three variants of Extended LO (ELO).
5. Sequent Calculi GELO_i, for $i \in \{w, m, s\}$.

INTRODUCTION

FOL versus natural language:

FOL versus natural language:

Two features of natural languages badly represented in FOL:

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

- the calculi of names due to Sommers;

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

- the calculi of names due to Sommers;
- the variety of relational sylogistics of Moss and Pratt-Hartmann;

INTRODUCTION

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

- the calculi of names due to Sommers;
- the variety of relational sylogistics of Moss and Pratt-Hartmann;
- the logic QUARC of Ben-Yami;

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

- the calculi of names due to Sommers;
- the variety of relational sylogistics of Moss and Pratt-Hartmann;
- the logic QUARC of Ben-Yami;
- the plural logic of Oliver and Smiley.

FOL versus natural language:

Two features of natural languages badly represented in FOL:

1. the subject-predicate structure of atomic sentences, characteristic not only for traditional logic but also for modern linguistics with its NP+VP model of sentences applied in generative grammar;
2. the wide class of naming expressions which are used not only to refer to x, but also to convey information about x, and even if they refer to something it is not necessarily the singular reference.

Reaction – some alternatives to FOL:

- the calculi of names due to Sommers;
- the variety of relational sylogistics of Moss and Pratt-Hartmann;
- the logic QUARC of Ben-Yami;
- the plural logic of Oliver and Smiley.

The oldest approach of this kind: Leśniewski's ontology.

Leśniewski and his Systems:

Leśniewski and his Systems:

Stanisław Leśniewski (1886-1939) – Polish Philosopher and Logician.

INTRODUCTION

Leśniewski and his Systems:

Stanisław Leśniewski (1886-1939) – Polish Philosopher and Logician.

- Protothetics - a general form of propositional logic where, in addition to sentence variables and specific connectives, arbitrary sentence-forming variables, as well as quantifiers binding all these kinds of variables are considered.

Leśniewski and his Systems:

Stanisław Leśniewski (1886-1939) – Polish Philosopher and Logician.

- Protothetics - a general form of propositional logic where, in addition to sentence variables and specific connectives, arbitrary sentence-forming variables, as well as quantifiers binding all these kinds of variables are considered.
- Ontology - the most comprehensive calculus of names proposed as an alternative (to Fregean paradigm) formalization of elementary logic.

Leśniewski and his Systems:

Stanisław Leśniewski (1886-1939) – Polish Philosopher and Logician.

- Protothetics - a general form of propositional logic where, in addition to sentence variables and specific connectives, arbitrary sentence-forming variables, as well as quantifiers binding all these kinds of variables are considered.
- Ontology - the most comprehensive calculus of names proposed as an alternative (to Fregean paradigm) formalization of elementary logic.
- Mereology - a theory of parthood relation proposed as the alternative (to set theory) formalization of the theory of classes, providing a nominalistic approach to foundations of mathematics.

Leśniewski's Ontology:

Leśniewski's Ontology:

- the most comprehensive calculus of names proposed as an alternative formalization of logic;

Leśniewski's Ontology:

- the most comprehensive calculus of names proposed as an alternative formalization of logic;
- a theory of the binary predicate ε meant as the formalization of the Greek 'esti';

INTRODUCTION

Leśniewski's Ontology:

- the most comprehensive calculus of names proposed as an alternative formalization of logic;
- a theory of the binary predicate ε meant as the formalization of the Greek 'esti';
- originally based on the protothetics which is a more general form of propositional logic where functorial variables as well as quantifiers binding all kinds of variables are involved;

Leśniewski's Ontology:

- the most comprehensive calculus of names proposed as an alternative formalization of logic;
- a theory of the binary predicate ε meant as the formalization of the Greek 'esti';
- originally based on the protothetics which is a more general form of propositional logic where functorial variables as well as quantifiers binding all kinds of variables are involved;
- alternative approach – a kind of first-order theory of ε based on classical first-order logic (Słupecki SL 1955, Iwanuś SL 1973).

Leśniewski's Ontology:

Leśniewski's Ontology:

Convention: instead of $a\varepsilon b$ we write ab.

Leśniewski's Ontology:

Convention: instead of $a\varepsilon b$ we write ab.

In all languages we have only name variables (bound x, y, z and free $a, b, c, d, ...$ called parameters) which range over all names (individual, general and empty).

Leśniewski's Ontology:

Convention: instead of $a\varepsilon b$ we write ab.

In all languages we have only name variables (bound x, y, z and free $a, b, c, d, \ldots$ called parameters) which range over all names (individual, general and empty).

LA (Leśniewski's axiom):
$\forall xy(xy \leftrightarrow \exists z(zx) \wedge \forall z(zx \rightarrow zy) \wedge \forall zv(zx \wedge vx \rightarrow zv))$

Leśniewski's Ontology:

Convention: instead of $a\varepsilon b$ we write ab.

In all languages we have only name variables (bound x, y, z and free $a, b, c, d, \ldots$ called parameters) which range over all names (individual, general and empty).

LA (Leśniewski's axiom):
$\forall xy(xy \leftrightarrow \exists z(zx) \wedge \forall z(zx \rightarrow zy) \wedge \forall zv(zx \wedge vx \rightarrow zv))$

The following formulae are equivalent to LA:

1. $\forall xy(xy \leftrightarrow \exists z(zx \wedge zy) \wedge \forall zv(zx \wedge vx \rightarrow zv))$
2. $\forall xy(xy \leftrightarrow \exists z(zx \wedge zy \wedge \forall v(vx \rightarrow vz)))$
3. $\forall xy(xy \leftrightarrow \exists z(\forall v(vx \leftrightarrow vz) \wedge zy))$

Leśniewski's Ontology - proof theory:

Leśniewski's Ontology - proof theory:

Ontology was often developed as a kind of ND: Słupecki, Lejewski, Wojciechowski, indeed Leśniewski himself.

INTRODUCTION

Leśniewski's Ontology - proof theory:

Ontology was often developed as a kind of ND: Słupecki, Lejewski, Wojciechowski, indeed Leśniewski himself.

There is also a tableau system for a part of LO due to Kobayashi and Ishimoto SL 1982 (also Ishimoto SL 1977, Takano 1985).

Leśniewski's Ontology - proof theory:

Ontology was often developed as a kind of ND: Słupecki, Lejewski, Wojciechowski, indeed Leśniewski himself.

There is also a tableau system for a part of LO due to Kobayashi and Ishimoto SL 1982 (also Ishimoto SL 1977, Takano 1985).

Recently cut-free sequent calculus GO for LO and GOP for LO with predicates was proposed by Indrzejczak [IJCAR 2022].

INTRODUCTION

Leśniewski's Ontology - proof theory:

Ontology was often developed as a kind of ND: Słupecki, Lejewski, Wojciechowski, indeed Leśniewski himself.

There is also a tableau system for a part of LO due to Kobayashi and Ishimoto SL 1982 (also Ishimoto SL 1977, Takano 1985).

Recently cut-free sequent calculus GO for LO and GOP for LO with predicates was proposed by Indrzejczak [IJCAR 2022].

Moreover it was shown that LO (with predicates) satisfies Craig Interpolation Theorem, constructively, via Maehara's method in GO and GOP by Indrzejczak [AWPL 2024].

SEQUENT CALCULUS GO

SEQUENT CALCULUS GO

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

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

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

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

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

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

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

$$(R)\ \frac{bb, \Gamma\Rightarrow \Delta}{bc, \Gamma\Rightarrow \Delta} \qquad (T)\ \frac{bd, \Gamma\Rightarrow \Delta}{bc, cd, \Gamma\Rightarrow \Delta} \qquad (S)\ \frac{cb, \Gamma\Rightarrow \Delta}{bc, cc, \Gamma\Rightarrow \Delta}$$

$$(E)\ \frac{ab, \Gamma\Rightarrow \Delta, ac \qquad ac, \Gamma\Rightarrow \Delta, ab \qquad cd, \Gamma \Rightarrow \Delta}{bd, \Gamma \Rightarrow \Delta}$$ where a is a fresh parameter (eigenvariable)

ADEQUACY OF GO

ADEQUACY OF GO

$$
\dfrac{\dfrac{aa \Rightarrow aa}{ab \Rightarrow aa}\,(R)}{ab \Rightarrow \exists x(xa)}\,(\Rightarrow \exists)
\qquad
\dfrac{\dfrac{\dfrac{cb \Rightarrow cb}{ca, ab \Rightarrow cb}\,(T)}{ab \Rightarrow ca \rightarrow cb}\,(\Rightarrow\rightarrow)}{ab \Rightarrow \forall x(xa \rightarrow xb)}\,(\Rightarrow \forall)
$$

$$
\dfrac{ab \Rightarrow \exists x(xa) \qquad ab \Rightarrow \forall x(xa \rightarrow xb)}{ab \Rightarrow \exists x(xa) \wedge \forall x(xa \rightarrow xb)}\,(\Rightarrow \wedge)
$$

$(\Rightarrow \wedge)$ with:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{cd \Rightarrow cd}{ca, \underline{ad} \Rightarrow cd}\,(T)}{ca, {\color{red}da}, \underline{aa} \Rightarrow cd}\,(S)}{ca, da, {\color{red}ab} \Rightarrow cd}\,(R)}{ab, ca \wedge da \Rightarrow cd}\,(\wedge \Rightarrow)}{ab \Rightarrow ca \wedge da \rightarrow cd}\,(\Rightarrow\rightarrow)}{ab \Rightarrow \forall xy(xa \wedge ya \rightarrow xy)}\,(\Rightarrow \forall)
$$

yields $LA^{\rightarrow}$ after $(\Rightarrow\rightarrow)$.

ADEQUACY OF GO

ADEQUACY OF GO

$$
\dfrac{ca \Rightarrow ca \qquad \dfrac{\dfrac{\dfrac{\dfrac{da \Rightarrow da \qquad ca \Rightarrow ca}{da, ca \Rightarrow da \wedge ca}(\Rightarrow\wedge) \qquad dc \Rightarrow dc}{da, ca, da \wedge ca \to dc \Rightarrow dc}(\to\Rightarrow)}{\underline{da}, ca, \forall xy(xa \wedge ya \to xy) \Rightarrow \underline{dc}}(\forall\Rightarrow) \qquad \dfrac{da \Rightarrow da}{\underline{dc}, ca \Rightarrow \underline{da}}(T) \qquad \underline{ab} \Rightarrow ab}{{\color{red}cb}, ca, \forall xy(xa \wedge ya \to xy) \Rightarrow ab}(E)}{\dfrac{\dfrac{ca, ca \to cb, \forall xy(xa \wedge ya \to xy) \Rightarrow ab}{ca, \forall x(xa \to xb), \forall xy(xa \wedge ya \to xy) \Rightarrow ab}(\forall\Rightarrow)}{\exists x(xa), \forall x(xa \to xb), \forall xy(xa \wedge ya \to xy) \Rightarrow ab}(\exists\Rightarrow)}(\to\Rightarrow)
$$

yields $LA^{\leftarrow}$ after $(\wedge \Rightarrow), (\Rightarrow\to)$.

ADEQUACY OF GO

$$
\dfrac{ca \Rightarrow ca \qquad \dfrac{(\forall\Rightarrow)\,\dfrac{(\to\Rightarrow)\,\dfrac{(\Rightarrow\wedge)\,\dfrac{da \Rightarrow da \qquad ca \Rightarrow ca}{da, ca \Rightarrow da \wedge ca} \qquad dc \Rightarrow dc}{da, ca, da \wedge ca \to dc \Rightarrow dc}}{\underline{da}, ca, \forall xy(xa \wedge ya \to xy) \Rightarrow \underline{dc}} \qquad \dfrac{da \Rightarrow da}{\underline{dc}, ca \Rightarrow \underline{da}}\,(T) \qquad \underline{ab} \Rightarrow ab}{{\color{red}cb}, ca, \forall xy(xa \wedge ya \to xy) \Rightarrow ab}\,(E)}{\dfrac{\dfrac{ca, ca \to cb, \forall xy(xa \wedge ya \to xy) \Rightarrow ab}{ca, \forall x(xa \to xb), \forall xy(xa \wedge ya \to xy) \Rightarrow ab}\,(\forall\Rightarrow)}{\exists x(xa), \forall x(xa \to xb), \forall xy(xa \wedge ya \to xy) \Rightarrow ab}\,(\exists\Rightarrow)}\,(\to\Rightarrow)
$$

yields LA$^{\leftarrow}$ after $(\wedge \Rightarrow), (\Rightarrow\to)$.

On the other hand $(R), (S), (T), (E)$ are derivable in GO with additional axioms $\Rightarrow LA^{\to}, \Rightarrow LA^{\leftarrow}$.

EXTENSIONS – SYSTEM GOP

$Da := \exists x, xa \quad Ea := \neg\exists x, xa \quad Sa := \exists x, ax$
$Ga := \exists xy(xa \wedge ya \wedge \neg xy) \quad Ua := \forall xy(xa \wedge ya \to xy)$

EXTENSIONS – SYSTEM GOP

$Da := \exists x, xa \quad Ea := \neg\exists x, xa \quad Sa := \exists x, ax$
$Ga := \exists xy(xa \wedge ya \wedge \neg xy) \quad Ua := \forall xy(xa \wedge ya \rightarrow xy)$

$$(D \Rightarrow)\ \frac{ba, \Gamma \Rightarrow \Delta}{Da, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow D)\ \frac{\Gamma \Rightarrow \Delta, ca}{\Gamma \Rightarrow \Delta, Da} \qquad (S \Rightarrow)\ \frac{ab, \Gamma \Rightarrow \Delta}{Sa, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow S)\ \frac{\Gamma \Rightarrow \Delta, ac}{\Gamma \Rightarrow \Delta, Sa} \qquad (E \Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ca}{Ea, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow E)\ \frac{ba, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, Ea}$$

where b is new in all schemata.

$$(G \Rightarrow)\ \frac{ba, ca, \Gamma \Rightarrow \Delta, bc}{Ga, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow G)\ \frac{\Gamma \Rightarrow \Delta, da \quad \Pi \Rightarrow \Sigma, ea \quad de, \Theta \Rightarrow \Lambda}{\Gamma, \Pi, \Theta \Rightarrow \Delta, \Sigma, \Lambda, Ga}$$

$$(\Rightarrow U)\ \frac{ba, ca, \Gamma \Rightarrow \Delta, bc}{\Gamma \Rightarrow \Delta, Ua} \qquad (U \Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, da \quad \Pi \Rightarrow \Sigma, ea \quad de, \Theta \Rightarrow \Lambda}{Ua, \Gamma, \Pi, \Theta \Rightarrow \Delta, \Sigma, \Lambda}$$

where b, c are new, and d, e are arbitrary parameters.

EXTENSIONS – SYSTEM GOP

EXTENSIONS – SYSTEM GOP

Identity and coextensiveness:

$a = b := ab \wedge ba \quad a \equiv b := \forall x(xa \leftrightarrow xb) \quad a \approx b := a \equiv b \wedge Da$

Identity and coextensiveness:

$$a = b := ab \wedge ba \quad a \equiv b := \forall x(xa \leftrightarrow xb) \quad a \approx b := a \equiv b \wedge Da$$

$$(=\Rightarrow)\ \frac{ab, ba, \Gamma \Rightarrow \Delta}{a = b, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow=)\ \frac{\Gamma \Rightarrow \Delta, ab \quad \Pi \Rightarrow \Sigma, ba}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a = b}$$

$$(\equiv\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ca, cb \quad ca, cb, \Pi \Rightarrow \Sigma}{a \equiv b, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \quad (\Rightarrow\equiv)\ \frac{da, \Gamma \Rightarrow \Delta, db \quad db, \Pi \Rightarrow \Sigma, da}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a \equiv b}$$

$$(\approx\Rightarrow)\ \frac{da, \Gamma \Rightarrow \Delta, ca, cb \quad ca, cb, da, \Pi \Rightarrow \Sigma}{a \approx b, \Gamma, \Pi \Rightarrow \Delta, \Sigma}$$

$$(\Rightarrow\approx)\ \frac{da, \Gamma \Rightarrow \Delta, db \quad db, \Pi \Rightarrow \Sigma, da \quad \Theta \Rightarrow \Lambda, ca}{\Gamma, \Pi, \Theta \Rightarrow \Delta, \Sigma, \Lambda, a \approx b}$$

where d is new and c arbitrary.

EXTENSIONS – SYSTEM GOP

Identity and coextensiveness:

$$a = b := ab \wedge ba \quad a \equiv b := \forall x(xa \leftrightarrow xb) \quad a \approx b := a \equiv b \wedge Da$$

$$(=\Rightarrow)\ \frac{ab, ba, \Gamma \Rightarrow \Delta}{a = b, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow=)\ \frac{\Gamma \Rightarrow \Delta, ab \quad \Pi \Rightarrow \Sigma, ba}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a = b}$$

$$(\equiv\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ca, cb \quad ca, cb, \Pi \Rightarrow \Sigma}{a \equiv b, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \quad (\Rightarrow\equiv)\ \frac{da, \Gamma \Rightarrow \Delta, db \quad db, \Pi \Rightarrow \Sigma, da}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a \equiv b}$$

$$(\approx\Rightarrow)\ \frac{da, \Gamma \Rightarrow \Delta, ca, cb \quad ca, cb, da, \Pi \Rightarrow \Sigma}{a \approx b, \Gamma, \Pi \Rightarrow \Delta, \Sigma}$$

$$(\Rightarrow\approx)\ \frac{da, \Gamma \Rightarrow \Delta, db \quad db, \Pi \Rightarrow \Sigma, da \quad \Theta \Rightarrow \Lambda, ca}{\Gamma, \Pi, \Theta \Rightarrow \Delta, \Sigma, \Lambda, a \approx b}$$

where d is new and c arbitrary.

Attention: let us call GO with two rules for $\equiv$, GOI.

EXTENSIONS – SYSTEM GOP

Inclusion and noninclusion:

$a\bar{\varepsilon}b := aa \wedge \neg ab$
$a \subset b := \forall x(xa \rightarrow xb)$
$a \not\subseteq b := \forall x(xa \rightarrow \neg xb)$

Inclusion and noninclusion:

$a\bar{\varepsilon}b := aa \wedge \neg ab$
$a \subset b := \forall x(xa \to xb)$
$a \not\subset b := \forall x(xa \to \neg xb)$

$$(\bar{\varepsilon} \Rightarrow)\ \frac{aa, \Gamma \Rightarrow \Delta, ab}{a\bar{\varepsilon}b, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow \bar{\varepsilon})\ \frac{\Gamma \Rightarrow \Delta, aa \quad ab, \Pi \Rightarrow \Sigma}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a\bar{\varepsilon}b}$$

$$(\subset\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ca \quad cb, \Pi \Rightarrow \Sigma}{a \subset b, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \qquad (\Rightarrow\subset)\ \frac{da, \Gamma \Rightarrow \Delta, db}{\Gamma \Rightarrow \Delta, a \subset b}$$

$$(\not\subset\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ca \quad \Pi \Rightarrow \Sigma, cb}{a \not\subset b, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \qquad (\Rightarrow\not\subset)\ \frac{da, db, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, a \not\subset b}$$

where d is new and c arbitrary.

EXTENSIONS – SYSTEM GOP

Categorical sentences:

$$aAb := a \subset b \wedge Da \quad aEb := a \not\subset b \wedge Da$$
$$aIb := \exists x(xa \wedge xb) \quad aOb := \exists x(xa \wedge \neg xb)$$

EXTENSIONS – SYSTEM GOP

Categorical sentences:

$$aAb := a \subset b \wedge Da \quad aEb := a \not\subset b \wedge Da$$
$$aIb := \exists x(xa \wedge xb) \quad aOb := \exists x(xa \wedge \neg xb)$$

$$(A \Rightarrow)\ \frac{da, \Gamma \Rightarrow \Delta, ca \qquad cb, da, \Pi \Rightarrow \Sigma}{aAb, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \qquad (\Rightarrow A)\ \frac{da, \Gamma \Rightarrow \Delta, db \qquad \Pi \Rightarrow \Sigma, ca}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, aAb}$$

$$(E \Rightarrow)\ \frac{da, \Gamma \Rightarrow \Delta, ca \qquad da, \Pi \Rightarrow \Sigma, cb}{aEb, \Gamma, \Pi \Rightarrow \Delta, \Sigma} \qquad (\Rightarrow E)\ \frac{da, db, \Gamma \Rightarrow \Delta \qquad \Pi \Rightarrow \Sigma, ca}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, aEb}$$

$$(I \Rightarrow)\ \frac{da, db, \Gamma \Rightarrow \Delta}{aIb, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow I)\ \frac{\Gamma \Rightarrow \Delta, ca \qquad \Pi \Rightarrow \Sigma, cb}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, aIb}$$

$$(O \Rightarrow)\ \frac{da, \Gamma \Rightarrow \Delta, db}{aOb, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow O)\ \frac{\Gamma \Rightarrow \Delta, ca \qquad cb, \Pi \Rightarrow \Sigma}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, aOb}$$

where d is new and c arbitrary.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.
6. The only primitive rules for ε are all one-sided (active formulae in the antecedents only), hence reduction of cut-height holds.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.
6. The only primitive rules for ε are all one-sided (active formulae in the antecedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 4, 5 and 6 [Indrzejczak IJCAR, Hajfa 2022].

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.
6. The only primitive rules for ε are all one-sided (active formulae in the antecedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 4, 5 and 6 [Indrzejczak IJCAR, Hajfa 2022].
8. The interpolation theorem holds due to 3 and 7 [Indrzejczak AWPL, Sapporo 2024].

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.
6. The only primitive rules for ε are all one-sided (active formulae in the antecedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 4, 5 and 6 [Indrzejczak IJCAR, Hajfa 2022].
8. The interpolation theorem holds due to 3 and 7 [Indrzejczak AWPL, Sapporo 2024].
9. The system is analytic due to 3 and 7.

HARVEST

1. All rules for constants are explicit, separate and symmetric which are usual requirements for well-behaved SC rules.
2. Several quantifier-free fragments may be formalised due to 1.
3. All rules, except cut, satisfy the subformula property – side formulae are only atomic of degree 0.
4. All rules are pairwise reductive, modulo substitution of terms, hence reduction of cut-degree holds.
5. Substitution theorem holds for the system with any combination of the above rules.
6. The only primitive rules for ε are all one-sided (active formulae in the antecedents only), hence reduction of cut-height holds.
7. Cut elimination holds due to 4, 5 and 6 [Indrzejczak IJCAR, Hajfa 2022].
8. The interpolation theorem holds due to 3 and 7 [Indrzejczak AWPL, Sapporo 2024].
9. The system is analytic due to 3 and 7.
10. Semidecision procedures (and decision procedures for quantifier-free fragments) can be provided due to 9.

THE PROBLEM

THE PROBLEM

How to extend LO to cover complex terms?

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

The original approach of Leśniewski to the problem is not satisfactory. There are two problems:

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

The original approach of Leśniewski to the problem is not satisfactory. There are two problems:

1. Definitions of term-forming operations in LO are creative.

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

The original approach of Leśniewski to the problem is not satisfactory. There are two problems:

1. Definitions of term-forming operations in LO are creative. Iwanuś has shown that the problem can be overcome by enriching elementary ontology with two axioms corresponding to special versions of the comprehension axiom but this opens a problem of derivability of these axioms in GO (GOP) enriched with special rules.

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

The original approach of Leśniewski to the problem is not satisfactory. There are two problems:

1. Definitions of term-forming operations in LO are creative. Iwanuś has shown that the problem can be overcome by enriching elementary ontology with two axioms corresponding to special versions of the comprehension axiom but this opens a problem of derivability of these axioms in GO (GOP) enriched with special rules.
2. Even if we can provide reductive rules for Leśniewski's operations, we run into a problem with quantifier rules. If unrestricted instantiation of terms is admitted in $(\Rightarrow \exists), (\forall \Rightarrow)$ the subformula property is lost.

THE PROBLEM

How to extend LO to cover complex terms?

How to provide cut-free SC for LO with complex terms?

The original approach of Leśniewski to the problem is not satisfactory. There are two problems:

1. Definitions of term-forming operations in LO are creative. Iwanuś has shown that the problem can be overcome by enriching elementary ontology with two axioms corresponding to special versions of the comprehension axiom but this opens a problem of derivability of these axioms in GO (GOP) enriched with special rules.
2. Even if we can provide reductive rules for Leśniewski's operations, we run into a problem with quantifier rules. If unrestricted instantiation of terms is admitted in $(\Rightarrow \exists), (\forall \Rightarrow)$ the subformula property is lost.

Another approach proposed by Waragai 1990.

Leśniewski's solution:

Example term functors:

$a\overline{b} := aa \wedge \neg ab$
$a(b \cap c) := ab \wedge ac$
$a(b \cup c) := ab \vee ac$

Leśniewski's solution:

Example term functors:

$a\bar{b} := aa \wedge \neg ab$
$a(b \cap c) := ab \wedge ac$
$a(b \cup c) := ab \vee ac$

$(- \Rightarrow)\ \dfrac{\Gamma \Rightarrow \Delta, ab}{a\bar{b}, \Gamma \Rightarrow \Delta}$ $\quad$ $(\Rightarrow -)\ \dfrac{ab, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, a\bar{b}}$

$(\cap \Rightarrow)\ \dfrac{ab, ac, \Gamma \Rightarrow \Delta}{a(b \cap c), \Gamma \Rightarrow \Delta}$ $\quad$ $(\Rightarrow \cap)\ \dfrac{\Gamma \Rightarrow \Delta, ab \quad \Pi \Rightarrow \Sigma, ac}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a(b \cap c)}$

$(\cup \Rightarrow)\ \dfrac{ab, \Gamma \Rightarrow \Delta \quad ac, \Pi \Rightarrow \Sigma}{a(b \cup c), \Gamma, \Pi \Rightarrow \Delta, \Sigma}$ $\quad$ $(\Rightarrow \cup)\ \dfrac{\Gamma \Rightarrow \Delta, ab, ac}{\Gamma \Rightarrow \Delta, a(b \cup c)}$

Leśniewski's solution:

Example term functors:

$a\bar{b} := aa \wedge \neg ab$
$a(b \cap c) := ab \wedge ac$
$a(b \cup c) := ab \vee ac$

$$(- \Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, ab}{a\bar{b}, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow -)\ \frac{ab, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, a\bar{b}}$$

$$(\cap \Rightarrow)\ \frac{ab, ac, \Gamma \Rightarrow \Delta}{a(b \cap c), \Gamma \Rightarrow \Delta} \qquad (\Rightarrow \cap)\ \frac{\Gamma \Rightarrow \Delta, ab \quad \Pi \Rightarrow \Sigma, ac}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, a(b \cap c)}$$

$$(\cup \Rightarrow)\ \frac{ab, \Gamma \Rightarrow \Delta \quad ac, \Pi \Rightarrow \Sigma}{a(b \cup c), \Gamma, \Pi \Rightarrow \Delta, \Sigma} \qquad (\Rightarrow \cup)\ \frac{\Gamma \Rightarrow \Delta, ab, ac}{\Gamma \Rightarrow \Delta, a(b \cup c)}$$

The rules are reductive but the system with these rules fails to be cut-free if quantifier rules $(\Rightarrow \exists), (\forall \Rightarrow)$ are not modified.

LO with lambda operator:

LO with lambda operator:

The language of GO extended with:

EXTENDED LO

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of n-ary relational predicate variables $R^n, n > 1$.

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of n-ary relational predicate variables $R^n, n > 1$.

Complex terms are of the form $\lambda x \varphi$, where φ is a formula.

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of n-ary relational predicate variables $R^n, n > 1$.

Complex terms are of the form $\lambda x \varphi$, where φ is a formula.
There are three kinds of atoms:

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of *n*-ary relational predicate variables $R^n, n > 1$.

Complex terms are of the form $\lambda x \varphi$, where φ is a formula.
There are three kinds of atoms:

- relational atoms $Rt_1 ... t_n$, where all arguments are variables;

EXTENDED LO

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of n-ary relational predicate variables $R^n, n > 1$.

Complex terms are of the form $\lambda x \varphi$, where φ is a formula.
There are three kinds of atoms:

- relational atoms $Rt_1 ... t_n$, where all arguments are variables;
- identities $t_1 \equiv t_2$, where both arguments can be simple or complex;

EXTENDED LO

LO with lambda operator:

The language of GO extended with:

- additional binary predicate $\equiv$ (considered in GOP);
- lambda operator λ;
- a denumerable set of n-ary relational predicate variables $R^n, n > 1$.

Complex terms are of the form $\lambda x \varphi$, where φ is a formula.
There are three kinds of atoms:

- relational atoms $Rt_1 ... t_n$, where all arguments are variables;
- identities $t_1 \equiv t_2$, where both arguments can be simple or complex;
- ε-atoms $t_1 \varepsilon t_2$.

LO with lambda operator:

LO with lambda operator:

We consider the hierarchy of three languages: weak, medium and strong, depending on what kind of terms are admitted as arguments of ε-atoms $t_1 \varepsilon t_2$:

LO with lambda operator:

We consider the hierarchy of three languages: weak, medium and strong, depending on what kind of terms are admitted as arguments of ε-atoms $t_1 \varepsilon t_2$:

1. $\mathcal{L}_w$: t_1 simple, t_2 arbitrary;

LO with lambda operator:

We consider the hierarchy of three languages: weak, medium and strong, depending on what kind of terms are admitted as arguments of ε-atoms $t_1 \varepsilon t_2$:

1. $\mathcal{L}_w$: t_1 simple, t_2 arbitrary;
2. $\mathcal{L}_m$: additionally ε-atoms with both arguments complex;

LO with lambda operator:

We consider the hierarchy of three languages: weak, medium and strong, depending on what kind of terms are admitted as arguments of ε-atoms $t_1 \varepsilon t_2$:

1. $\mathcal{L}_w$: t_1 simple, t_2 arbitrary;
2. $\mathcal{L}_m$: additionally ε-atoms with both arguments complex;
3. $\mathcal{L}_s$: additionally ε-atoms with t_1 complex and t_2 simple.

LO with lambda operator:

We consider the hierarchy of three languages: weak, medium and strong, depending on what kind of terms are admitted as arguments of ε-atoms $t_1 \varepsilon t_2$:

1. $\mathcal{L}_w$: t_1 simple, t_2 arbitrary;
2. $\mathcal{L}_m$: additionally ε-atoms with both arguments complex;
3. $\mathcal{L}_s$: additionally ε-atoms with t_1 complex and t_2 simple.

So only $\mathcal{L}_s$ admits all possible combination of terms, as in identities.

EXTENDED LO

LO with lambda operator:

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LA_1 $ab \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow zb) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LA_1 $ab \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow zb) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LA_2 $a\lambda x\psi \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow z\lambda x\psi) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LA_1 $ab \leftrightarrow \exists z(za) \wedge \forall z(za \to zb) \wedge \forall zv(za \wedge va \to zv)$;

LA_2 $a\lambda x\psi \leftrightarrow \exists z(za) \wedge \forall z(za \to z\lambda x\psi) \wedge \forall zv(za \wedge va \to zv)$;

LA_3 $\lambda x\varphi\lambda x\psi \leftrightarrow \exists z(z\lambda x\varphi) \wedge \forall z(z\lambda x\varphi \to z\lambda x\psi) \wedge \forall zv(z\lambda x\varphi \wedge v\lambda x\varphi \to zv)$;

EXTENDED LO

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LA_1 $ab \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow zb) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LA_2 $a\lambda x\psi \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow z\lambda x\psi) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LA_3 $\lambda x\varphi\lambda x\psi \leftrightarrow \exists z(z\lambda x\varphi) \wedge \forall z(z\lambda x\varphi \rightarrow z\lambda x\psi) \wedge \forall zv(z\lambda x\varphi \wedge v\lambda x\varphi \rightarrow zv)$;

LA_4 $\lambda x\varphi b \leftrightarrow \exists z(z\lambda x\varphi) \wedge \forall z(z\lambda x\varphi \rightarrow zb) \wedge \forall zv(z\lambda x\varphi \wedge v\lambda x\varphi \rightarrow zv)$.

EXTENDED LO

LO with lambda operator:

Note that in the setting of ELO, the axiom *LA* covers in fact four schemata:

LA_1 $ab \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow zb) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LA_2 $a\lambda x\psi \leftrightarrow \exists z(za) \wedge \forall z(za \rightarrow z\lambda x\psi) \wedge \forall zv(za \wedge va \rightarrow zv)$;

LA_3 $\lambda x\varphi\lambda x\psi \leftrightarrow \exists z(z\lambda x\varphi) \wedge \forall z(z\lambda x\varphi \rightarrow z\lambda x\psi) \wedge \forall zv(z\lambda x\varphi \wedge v\lambda x\varphi \rightarrow zv)$;

LA_4 $\lambda x\varphi b \leftrightarrow \exists z(z\lambda x\varphi) \wedge \forall z(z\lambda x\varphi \rightarrow zb) \wedge \forall zv(z\lambda x\varphi \wedge v\lambda x\varphi \rightarrow zv)$.

They form a hierarchy of the commitment of complex terms in forming atoms of ELO, representing different strength of expression.

three variants of ELO formalised in respective languages:

three variants of ELO formalised in respective languages:

1. weak ELO_w in $\mathcal{L}_w$ satisfying LA_1, LA_2;

three variants of ELO formalised in respective languages:

1. weak ELO_w in $\mathcal{L}_w$ satisfying LA_1, LA_2;
2. medium ELO_m in $\mathcal{L}_m$ satisfying LA_1, LA_2, LA_3;

three variants of ELO formalised in respective languages:

1. weak ELO_w in $\mathcal{L}_w$ satisfying LA_1, LA_2;
2. medium ELO_m in $\mathcal{L}_m$ satisfying LA_1, LA_2, LA_3;
3. strong ELO_s in $\mathcal{L}_s$ satisfying LA_1, LA_2, LA_3, LA_4.

Lambda operator:

Lambda operator:

Even ELO_s is in a sense too weak for real applications to the analysis of reasoning in natural languages. For example, we are not able to demonstrate the validity of such simple argument as:

EXTENDED LO

Lambda operator:

Even ELO_s is in a sense too weak for real applications to the analysis of reasoning in natural languages. For example, we are not able to demonstrate the validity of such simple argument as:

'Ann is the oldest daughter of Betty. Therefore, she is Betty's daughter.'

Lambda operator:

Even ELO_s is in a sense too weak for real applications to the analysis of reasoning in natural languages. For example, we are not able to demonstrate the validity of such simple argument as:

‘Ann is the oldest daughter of Betty. Therefore, she is Betty’s daughter.’

To resolve this problem we need a kind of β-conversion (BC) of the form:

Lambda operator:

Even ELO_s is in a sense too weak for real applications to the analysis of reasoning in natural languages. For example, we are not able to demonstrate the validity of such simple argument as:

'Ann is the oldest daughter of Betty. Therefore, she is Betty's daughter.'

To resolve this problem we need a kind of β-conversion (BC) of the form:

$$a\lambda x\varphi \leftrightarrow aa \wedge \varphi[x/a]$$

where aa is added to restrict a to individual names.

Lambda operator:

Even ELO_s is in a sense too weak for real applications to the analysis of reasoning in natural languages. For example, we are not able to demonstrate the validity of such simple argument as:

'Ann is the oldest daughter of Betty. Therefore, she is Betty's daughter.'

To resolve this problem we need a kind of β-conversion (BC) of the form:

$$a\lambda x\varphi \leftrightarrow aa \wedge \varphi[x/a]$$

where aa is added to restrict a to individual names.

Similar principle was considered by Waragai in his system combining FOL with LO.

How to obtain well-behaved SC for ELO?

EXTENDED LO

How to obtain well-behaved SC for ELO?

One may think about the generalisation of the rules of GO to arbitrary terms but we loose the subformula property.

How to obtain well-behaved SC for ELO?

One may think about the generalisation of the rules of GO to arbitrary terms but we loose the subformula property.

The better option is to introduce new rules for ε-atoms with complex terms to obtain the (three systems of) GELO.

How to obtain well-behaved SC for ELO?

One may think about the generalisation of the rules of GO to arbitrary terms but we loose the subformula property.

The better option is to introduce new rules for ε-atoms with complex terms to obtain the (three systems of) GELO.

The starting point is the system GOI, i.e. GO with two rules for $\equiv$:

$$(\equiv\Rightarrow) \quad \frac{\Gamma \Rightarrow \Delta, db, dc \qquad db, dc, \Pi \Rightarrow \Sigma}{b \equiv c, \Gamma, \Pi \Rightarrow \Delta, \Sigma}$$

$$(\Rightarrow\equiv) \quad \frac{ab, \Gamma \Rightarrow \Delta, ac \qquad ac, \Pi \Rightarrow \Sigma, ab}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, b \equiv c}$$

where a is new.

$\text{GELO}_w := \text{GOI in } \mathcal{L}_w + \text{the following rules:}$

EXTENDED LO

GELO$_w$:= GOI in $\mathcal{L}_w$ + the following rules:

$$(\beta \Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{b\lambda x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow \beta)\ \frac{\Gamma \Rightarrow \Delta, bb \quad \Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, b\lambda x\varphi}$$

$$(\equiv\Rightarrow E)\ \frac{a \equiv t, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad (\Rightarrow\equiv E)\ \frac{\Gamma \Rightarrow \Delta, b \equiv c \quad \Gamma \Rightarrow \Delta, \varphi[x/c]}{\Gamma \Rightarrow \Delta, \varphi[x/b]}$$

where a is a fresh parameter (eigenvariable), b, c are arbitrary parameters,
$t \in term(\Gamma \cup \Delta)$ [the set of complex terms of $\Gamma \cup \Delta$] in $(\equiv\Rightarrow E)$,
φ in $(\Rightarrow\equiv E)$ is a relational atom.

$GELO_w$:= GOI in $\mathcal{L}_w$ + the following rules:

$$(\beta \Rightarrow)\ \frac{\varphi[x/b], \Gamma \Rightarrow \Delta}{b\lambda x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow \beta)\ \frac{\Gamma \Rightarrow \Delta, bb \quad \Gamma \Rightarrow \Delta, \varphi[x/b]}{\Gamma \Rightarrow \Delta, b\lambda x\varphi}$$

$$(\equiv\Rightarrow E)\ \frac{a \equiv t, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad (\Rightarrow\equiv E)\ \frac{\Gamma \Rightarrow \Delta, b \equiv c \quad \Gamma \Rightarrow \Delta, \varphi[x/c]}{\Gamma \Rightarrow \Delta, \varphi[x/b]}$$

where a is a fresh parameter (eigenvariable), b, c are arbitrary parameters,
$t \in term(\Gamma \cup \Delta)$ [the set of complex terms of $\Gamma \cup \Delta$] in $(\equiv\Rightarrow E)$,
φ in $(\Rightarrow\equiv E)$ is a relational atom.

Note that there is no need to generalise the rules $(R), (T), (S), (E)$ to cover complex terms!

THE IMPORTANCE OF ($\equiv\Rightarrow E$):

THE IMPORTANCE OF $(\equiv\Rightarrow E)$:

$(\forall\Rightarrow), (\Rightarrow\exists)$ are derivable by $(\equiv\Rightarrow E)$:

$$
(Cut)\ \dfrac{(\equiv\Rightarrow E)\ \dfrac{(\forall\Rightarrow)\ \dfrac{a \equiv t, \varphi[x/a] \Rightarrow \varphi[x/t]}{a \equiv t, \forall x\varphi \Rightarrow \varphi[x/t]}}{\forall x\varphi \Rightarrow \varphi[x/t]} \qquad \varphi[x/t], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta}
$$

where the left top sequent is a provable instance of Leibniz Law *LL*. In a similar way we prove derivability of unrestricted $(\Rightarrow\exists)$.

THE IMPORTANCE OF $(\equiv\Rightarrow E)$:

$(\forall\Rightarrow), (\Rightarrow\exists)$ are derivable by $(\equiv\Rightarrow E)$:

$$
(Cut)\ \dfrac{(\equiv\Rightarrow E)\ \dfrac{(\forall\Rightarrow)\ \dfrac{a \equiv t, \varphi[x/a] \Rightarrow \varphi[x/t]}{a \equiv t, \forall x\varphi \Rightarrow \varphi[x/t]}}{\forall x\varphi \Rightarrow \varphi[x/t]} \qquad \varphi[x/t], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta}
$$

where the left top sequent is a provable instance of Leibniz Law *LL*.
In a similar way we prove derivability of unrestricted $(\Rightarrow\exists)$.
$(\equiv\Rightarrow E)$ is derivable in the calculus with unrestricted $(\Rightarrow\exists)$:

$$
(Cut)\ \dfrac{(\Rightarrow\exists)\ \dfrac{(\Rightarrow\equiv)\ \dfrac{at \Rightarrow at \qquad at \Rightarrow at}{\Rightarrow t \equiv t}}{\Rightarrow \exists x(x \equiv t)} \qquad \dfrac{a \equiv t, \Gamma \Rightarrow \Delta}{\exists x(x \equiv t), \Gamma \Rightarrow \Delta}\ (\exists\Rightarrow)}{\Gamma \Rightarrow \Delta}
$$

ADEQUACY OF GELO$_w$

ADEQUACY OF GELO$_w$

$$
\dfrac{a\lambda x\varphi \Rightarrow ab, a\lambda x\varphi \qquad \dfrac{\dfrac{aa \Rightarrow aa}{aa \Rightarrow \exists x(xa)}\,(\Rightarrow \exists)}{ab, a\lambda x\varphi \Rightarrow \exists x(xa)}\,(R)}{\dfrac{b \equiv \lambda x\varphi, a\lambda x\varphi \Rightarrow \exists x(xa)}{a\lambda x\varphi \Rightarrow \exists x(xa)}\,(\equiv\Rightarrow E)}\,(\equiv\Rightarrow)
$$

$$
\dfrac{\dfrac{a\lambda x\varphi \Rightarrow ac, a\lambda x\varphi \qquad \dfrac{bc \Rightarrow bc}{ac, a\lambda x\varphi, ba \Rightarrow bc}\,(T)}{c \equiv \lambda x\varphi, a\lambda x\varphi, ba \Rightarrow bc, b\lambda x\varphi}\,(\equiv\Rightarrow) \qquad bc, b\lambda x\varphi \Rightarrow b\lambda x\varphi}{\dfrac{\dfrac{\dfrac{c \equiv \lambda x\varphi, a\lambda x\varphi, ba \Rightarrow b\lambda x\varphi}{a\lambda x\varphi, ba \Rightarrow b\lambda x\varphi}\,(\equiv\Rightarrow E)}{a\lambda x\varphi \Rightarrow ba \rightarrow b\lambda x\varphi}\,(\Rightarrow\rightarrow)}{a\lambda x\varphi \Rightarrow \forall x(xa \rightarrow x\lambda x\varphi)}\,(\Rightarrow \forall)}\,(\equiv\Rightarrow)
$$

$$
\dfrac{a\lambda x\varphi \Rightarrow ab, a\lambda x\varphi \qquad \dfrac{\dfrac{\dfrac{cd \Rightarrow cd}{ca, \underline{ad} \Rightarrow cd}\,(T)}{\underline{aa}, ca, da \Rightarrow cd}\,(S)}{ab, a\lambda x\varphi, ca, da \Rightarrow cd}\,(R)}{\dfrac{\dfrac{\dfrac{\dfrac{b \equiv \lambda x\varphi, a\lambda x\varphi, ca, da \Rightarrow cd}{b \equiv \lambda x\varphi, a\lambda x\varphi, ca \wedge da \Rightarrow cd}\,(\wedge \Rightarrow)}{b \equiv \lambda x\varphi, a\lambda x\varphi \Rightarrow ca \wedge da \rightarrow cd}\,(\Rightarrow\rightarrow)}{b \equiv \lambda x\varphi, a\lambda x\varphi \Rightarrow \forall xy(xa \wedge ya \rightarrow xy)}\,(\Rightarrow \forall)}{a\lambda x\varphi \Rightarrow \forall xy(xa \wedge ya \rightarrow xy)}\,(\equiv\Rightarrow E)}\,(\equiv\Rightarrow)
$$

yield together by $(\Rightarrow \wedge)$ and $(\Rightarrow\rightarrow)$ the left-right implication of LA_2.

ADEQUACY OF GELO$_w$

$$\dfrac{ba \Rightarrow ba \qquad \dfrac{\dfrac{b\lambda x\varphi \Rightarrow bc, b\lambda x\varphi \qquad D}{c \equiv \lambda x\varphi, ba, b\lambda x\varphi, \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}(\equiv\Rightarrow)}{ba, b\lambda x\varphi, \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}(\equiv\Rightarrow E)}{\dfrac{ba, ba \to b\lambda x\varphi, \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}{\dfrac{ba, \forall x(xa \to x\lambda x\varphi), \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}{\exists x(xa), \forall x(xa \to x\lambda x\varphi), \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}(\exists\Rightarrow)}(\forall\Rightarrow)}(\to\Rightarrow)$$

where D is:

$$\dfrac{D_1 \qquad \dfrac{da \Rightarrow da}{ba, \underline{db} \Rightarrow \underline{da}}(T) \qquad \dfrac{ac \Rightarrow ac, a\lambda x\varphi \qquad ac, a\lambda x\varphi \Rightarrow a\lambda x\varphi}{\underline{ac}, c \equiv \lambda x\varphi \Rightarrow a\lambda x\varphi}(\equiv\Rightarrow)}{{\color{red}bc}, b\lambda x\varphi, c \equiv \lambda x\varphi, ba, \forall xy(xa \wedge ya \to xy) \Rightarrow a\lambda x\varphi}(E)$$

and D_1 is:

$$(\forall\Rightarrow)\dfrac{(\to\Rightarrow)\dfrac{(\Rightarrow\wedge)\dfrac{ba \Rightarrow ba \qquad da \Rightarrow da}{ba, da \Rightarrow da \wedge ba} \qquad db \Rightarrow db}{ba, da, da \wedge ba \to db \Rightarrow db}}{ba, \forall xy(xa \wedge ya \to xy), \underline{da} \Rightarrow \underline{db}}$$

□

yields $\mathrm{LA}_2^{\leftarrow}$

$\text{GELO}_m := \text{GELO}_w$ in $\mathcal{L}_m$ + the following rules:

GELO$_m$:= GELO$_w$ in $\mathcal{L}_m$ + the following rules:

$$(\lambda \Rightarrow 1)\quad \frac{a\lambda x\varphi, at, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\lambda \Rightarrow 2)\quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, d\lambda x\varphi \qquad cd, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow \lambda)\quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, ct \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi t}$$

where a, b are new parameters (eigenvariable), c, d are arbitrary, t is complex.

$\text{GELO}_m := \text{GELO}_w$ in $\mathcal{L}_m$ + the following rules:

$$(\lambda \Rightarrow 1)\quad \frac{a\lambda x\varphi, at, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\lambda \Rightarrow 2)\quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, d\lambda x\varphi \qquad cd, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow \lambda)\quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, ct \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi t}$$

where a, b are new parameters (eigenvariable), c, d are arbitrary, t is complex.

$\text{GELO}_s := \text{GELO}_m$ in $\mathcal{L}_s$:

EXTENDED LO

$GELO_m$:= $GELO_w$ in $\mathcal{L}_m$ + the following rules:

$$(\lambda \Rightarrow 1) \quad \frac{a\lambda x\varphi, at, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\lambda \Rightarrow 2) \quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, d\lambda x\varphi \qquad cd, \Gamma \Rightarrow \Delta}{\lambda x\varphi t, \Gamma \Rightarrow \Delta}$$

$$(\Rightarrow \lambda) \quad \frac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, ct \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi t}$$

where a, b are new parameters (eigenvariable), c, d are arbitrary, t is complex.

$GELO_s$:= $GELO_m$ in $\mathcal{L}_s$:

Note – no new rules! Just the relaxation of formulation: in $GELO_s$ t may be an arbitrary term in $(\lambda \Rightarrow 1)$, $(\lambda \Rightarrow 2)$ and $(\Rightarrow \lambda)$.

ADEQUACY OF GELO$_m$

$$\dfrac{\dfrac{a\lambda x\varphi, at \Rightarrow a\lambda x\varphi}{a\lambda x\varphi, at \Rightarrow \exists x(x\lambda x\varphi)}\;(\Rightarrow \exists)}{\lambda x\varphi t \Rightarrow \exists x(x\lambda x\varphi)}\;(\lambda \Rightarrow 1)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{a\lambda x\varphi \Rightarrow \underline{a\lambda x\varphi} \qquad b\lambda x\varphi \Rightarrow \underline{b\lambda x\varphi} \qquad \underline{ab}, bt \Rightarrow at}{a\lambda x\varphi, \underline{b\lambda x\varphi}, \underline{bt}, \color{red}{\lambda x\varphi t} \Rightarrow at}\;(\lambda \Rightarrow 2)}{a\lambda x\varphi, \color{red}{\lambda x\varphi t} \Rightarrow at}\;(\lambda \Rightarrow 1)}{\lambda x\varphi t \Rightarrow a\lambda x\varphi \rightarrow at}\;(\Rightarrow \rightarrow)}{\lambda x\varphi t \Rightarrow \forall x(x\lambda x\varphi \rightarrow xt)}\;(\Rightarrow \forall)$$

where the rightmost sequent is provable.

$$\dfrac{\dfrac{\dfrac{\dfrac{a\lambda x\varphi \Rightarrow \underline{a\lambda x\varphi} \qquad b\lambda x\varphi \Rightarrow \underline{b\lambda x\varphi} \qquad \underline{ab} \Rightarrow ab}{\color{red}{\lambda x\varphi t}, a\lambda x\varphi, b\lambda x\varphi \Rightarrow ab}\;(\lambda \Rightarrow 2)}{\lambda x\varphi t, a\lambda x\varphi \wedge b\lambda x\varphi \Rightarrow ab}\;(\wedge \Rightarrow)}{\lambda x\varphi t \Rightarrow a\lambda x\varphi \wedge b\lambda x\varphi \rightarrow ab}\;(\Rightarrow \rightarrow)}{\lambda x\varphi t \Rightarrow \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \rightarrow xy)}\;(\Rightarrow \forall)$$

the above proofs yield the left-right part of LA_3 after application of $(\Rightarrow \wedge)$ and $(\Rightarrow\rightarrow)$.

ADEQUACY OF GELO$_m$

ADEQUACY OF GELO$_m$

$$\dfrac{\dfrac{\dfrac{a\lambda x\varphi \Rightarrow a\lambda x\varphi \qquad at, a\lambda x\varphi, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}{a\lambda x\varphi, a\lambda x\varphi \to at, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}\,(\to\Rightarrow)}{a\lambda x\varphi t, \forall x(x\lambda x\varphi \to xt), \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}\,(\forall\Rightarrow)}{\exists x(x\lambda x\varphi), \forall x(x\lambda x\varphi \to xt), \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}\,(\exists\Rightarrow)$$

where the rightmost sequent is proved as follows:

$$\dfrac{a\lambda x\varphi \Rightarrow \underline{a\lambda x\varphi} \qquad at \Rightarrow \underline{at} \qquad \dfrac{\dfrac{(\Rightarrow\wedge)\,\dfrac{b\lambda x\varphi \Rightarrow b\lambda x\varphi \qquad c\lambda x\varphi \Rightarrow c\lambda x\varphi}{b\lambda x\varphi, c\lambda x\varphi \Rightarrow b\lambda x\varphi \wedge c\lambda x\varphi} \qquad bc \Rightarrow bc}{b\lambda x\varphi, c\lambda x\varphi, b\lambda x\varphi \wedge c\lambda x\varphi \to bc \Rightarrow bc}\,(\to\Rightarrow)}{\underline{b\lambda x\varphi}, \underline{c\lambda x\varphi}, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \underline{bc}}\,(\forall\Rightarrow)}{at, a\lambda x\varphi, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}\,(\Rightarrow\lambda)$$

ADEQUACY OF GELO$_m$

$$\dfrac{\dfrac{\dfrac{a\lambda x\varphi \Rightarrow a\lambda x\varphi \qquad at, a\lambda x\varphi, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}{a\lambda x\varphi, a\lambda x\varphi \to at, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}(\to\Rightarrow)}{a\lambda x\varphi t, \forall x(x\lambda x\varphi \to xt), \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}(\forall\Rightarrow)}{\exists x(x\lambda x\varphi), \forall x(x\lambda x\varphi \to xt), \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}(\exists\Rightarrow)$$

where the rightmost sequent is proved as follows:

$$\dfrac{a\lambda x\varphi \Rightarrow \underline{a\lambda x\varphi} \qquad at \Rightarrow \underline{at} \qquad \dfrac{\dfrac{(\Rightarrow\wedge)\dfrac{b\lambda x\varphi \Rightarrow b\lambda x\varphi \qquad c\lambda x\varphi \Rightarrow c\lambda x\varphi}{b\lambda x\varphi, c\lambda x\varphi \Rightarrow b\lambda x\varphi \wedge c\lambda x\varphi} \qquad bc \Rightarrow bc}{b\lambda x\varphi, c\lambda x\varphi, b\lambda x\varphi \wedge c\lambda x\varphi \to bc \Rightarrow bc}(\to\Rightarrow)}{\underline{b\lambda x\varphi}, \underline{c\lambda x\varphi}, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \underline{bc}}(\forall\Rightarrow)}{at, a\lambda x\varphi, \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \to xy) \Rightarrow \lambda x\varphi t}(\Rightarrow\lambda)$$

For GELO$_s$ the proof is similar.

EXTENDED LO – SUMMARY OF RESULTS

EXTENDED LO – SUMMARY OF RESULTS

Lemma

$GELO_i \vdash s \equiv t, \varphi[x/s] \Rightarrow \varphi[x/t]$, *for* $i \in \{w, m, s\}$.

EXTENDED LO – SUMMARY OF RESULTS

Lemma

$GELO_i \vdash s \equiv t, \varphi[x/s] \Rightarrow \varphi[x/t]$, *for* $i \in \{w, m, s\}$.

Lemma

$a\lambda x\psi \leftrightarrow \exists x(xa) \land \forall x(xa \rightarrow x\lambda x\psi) \land \forall xy(xa \land ya \rightarrow xy)$ *is provable in* $GELO_w$.

EXTENDED LO – SUMMARY OF RESULTS

Lemma

$GELO_i \vdash s \equiv t, \varphi[x/s] \Rightarrow \varphi[x/t]$, for $i \in \{w, m, s\}$.

Lemma

$a\lambda x\psi \leftrightarrow \exists x(xa) \wedge \forall x(xa \rightarrow x\lambda x\psi) \wedge \forall xy(xa \wedge ya \rightarrow xy)$ is provable in $GELO_w$.

Lemma

$\lambda x\varphi t \leftrightarrow \exists x(x\lambda x\varphi) \wedge \forall x(x\lambda x\varphi \rightarrow xt) \wedge \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \rightarrow xy)$ is provable in $GELO_m$ with t complex, and in $GELO_s$ with t arbitrary.

EXTENDED LO – SUMMARY OF RESULTS

Lemma

$GELO_i \vdash s \equiv t, \varphi[x/s] \Rightarrow \varphi[x/t]$, for $i \in \{w, m, s\}$.

Lemma

$a\lambda x\psi \leftrightarrow \exists x(xa) \wedge \forall x(xa \rightarrow x\lambda x\psi) \wedge \forall xy(xa \wedge ya \rightarrow xy)$ is provable in $GELO_w$.

Lemma

$\lambda x\varphi t \leftrightarrow \exists x(x\lambda x\varphi) \wedge \forall x(x\lambda x\varphi \rightarrow xt) \wedge \forall xy(x\lambda x\varphi \wedge y\lambda x\varphi \rightarrow xy)$ is provable in $GELO_m$ with t complex, and in $GELO_s$ with t arbitrary.

Lemma

The rules of GELOi are derivable in $GOI+LA_i$ used as an additional axiomatic sequent, for $i \in \{w, m, s\}$. .

CUT ELIMINATION IN ELO

Note that:

Note that:

1. if st is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;

Note that:

1. if st is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;
2. if it is of the form $b\lambda x\varphi$, it can be principal in both premisses of cut but only via $(\Rightarrow \beta)$ and $(\beta \Rightarrow)$;

Note that:

1. if *st* is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;
2. if it is of the form $b\lambda x\varphi$, it can be principal in both premisses of cut but only via $(\Rightarrow \beta)$ and $(\beta \Rightarrow)$;
3. if it is of the form $\lambda x\varphi t$, it can be principal in both premisses of cut but only via $(\Rightarrow \lambda)$ and $(\lambda \Rightarrow 1)$ or $(\lambda \Rightarrow 2)$;

Note that:

1. if st is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;
2. if it is of the form $b\lambda x\varphi$, it can be principal in both premisses of cut but only via $(\Rightarrow \beta)$ and $(\beta \Rightarrow)$;
3. if it is of the form $\lambda x\varphi t$, it can be principal in both premisses of cut but only via $(\Rightarrow \lambda)$ and $(\lambda \Rightarrow 1)$ or $(\lambda \Rightarrow 2)$;
4. identity is principal in both premisses of cut only via $(\Rightarrow\equiv)$ and $(\equiv\Rightarrow)$;

Note that:

1. if *st* is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;
2. if it is of the form $b\lambda x\varphi$, it can be principal in both premisses of cut but only via $(\Rightarrow \beta)$ and $(\beta \Rightarrow)$;
3. if it is of the form $\lambda x\varphi t$, it can be principal in both premisses of cut but only via $(\Rightarrow \lambda)$ and $(\lambda \Rightarrow 1)$ or $(\lambda \Rightarrow 2)$;
4. identity is principal in both premisses of cut only via $(\Rightarrow\equiv)$ and $(\equiv\Rightarrow)$;
5. relational atom is principal only in the succedent of the left premiss via $(\Rightarrow\equiv E)$.

CUT ELIMINATION IN ELO

Note that:

1. if *st* is strictly atomic, i.e. containing parameters only, it can be principal only in the antecedent of the right premiss of cut, due to $(R), (S), (T), (E)$;
2. if it is of the form $b\lambda x\varphi$, it can be principal in both premisses of cut but only via $(\Rightarrow \beta)$ and $(\beta \Rightarrow)$;
3. if it is of the form $\lambda x\varphi t$, it can be principal in both premisses of cut but only via $(\Rightarrow \lambda)$ and $(\lambda \Rightarrow 1)$ or $(\lambda \Rightarrow 2)$;
4. identity is principal in both premisses of cut only via $(\Rightarrow\equiv)$ and $(\equiv\Rightarrow)$;
5. relational atom is principal only in the succedent of the left premiss via $(\Rightarrow\equiv E)$.

In cases 1, 5 we proceed by induction on the height, in cases 2, 3, 4 by induction on the grade.

CUT ELIMINATION IN ELO

Lemma (Substitution)

If $\vdash_k \Gamma \Rightarrow \Delta$, *then* $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$.

Lemma (Substitution)

If $\vdash_k \Gamma \Rightarrow \Delta$, *then* $\vdash_k \Gamma[a/b] \Rightarrow \Delta[a/b]$.

Lemma

1. *The rules* $(\Rightarrow \beta)$ *with* $(\beta \Rightarrow)$ *are reductive in general;*
2. $(\Rightarrow \lambda)$ *with* $(\lambda \Rightarrow 1)$, *and* $(\Rightarrow \lambda)$ *with* $(\lambda \Rightarrow 2)$ *are pairwise reductive in* $GELO_m$.

CUT ELIMINATION IN ELO - reductivity of λ-rules:

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$
\dfrac{(\Rightarrow\lambda)\ \dfrac{\Gamma\Rightarrow\Delta, c\lambda x\varphi \qquad \Gamma\Rightarrow\Delta, c\lambda y\psi \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma\Rightarrow\Delta, ab}{\Gamma\Rightarrow\Delta, \lambda x\varphi\lambda y\psi} \qquad \dfrac{d\lambda x\varphi, d\lambda y\psi, \Pi\Rightarrow\Sigma}{\lambda x\varphi\lambda y\psi, \Pi\Rightarrow\Sigma}\ (\lambda\Rightarrow 1)}{\Gamma, \Pi\Rightarrow\Delta, \Sigma}\ (Cut)
$$

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$\dfrac{(\Rightarrow\lambda)\ \dfrac{\Gamma\Rightarrow\Delta, c\lambda x\varphi \qquad \Gamma\Rightarrow\Delta, c\lambda y\psi \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma\Rightarrow\Delta, ab}{\Gamma\Rightarrow\Delta, \lambda x\varphi\lambda y\psi} \qquad \dfrac{d\lambda x\varphi, d\lambda y\psi, \Pi\Rightarrow\Sigma}{\lambda x\varphi\lambda y\psi, \Pi\Rightarrow\Sigma}\ (\lambda\Rightarrow 1)}{\Gamma, \Pi\Rightarrow\Delta, \Sigma}\ (Cut)$$

we apply substitution lemma to premiss of $(\lambda\Rightarrow 1)$ to replace the occurrences of fresh d with c, then we continue:

$$\dfrac{\dfrac{\Gamma\Rightarrow\Delta, c\lambda y\psi \qquad \dfrac{\Gamma\Rightarrow\Delta, c\lambda x\varphi \qquad c\lambda x\varphi, c\lambda y\psi, \Pi\Rightarrow\Sigma}{c\lambda y\psi, \Gamma, \Pi\Rightarrow\Delta, \Sigma}\ (Cut)}{\Gamma, \Gamma, \Pi\Rightarrow\Delta, \Delta, \Sigma}\ (Cut)}{\Gamma, \Pi\Rightarrow\Delta, \Sigma}\ (C\Rightarrow), (\Rightarrow C)$$

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$\cfrac{(\Rightarrow \lambda)\ \cfrac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad \Gamma \Rightarrow \Delta, c\lambda y\psi \qquad a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi\lambda y\psi} \qquad \cfrac{d\lambda x\varphi, d\lambda y\psi, \Pi \Rightarrow \Sigma}{\lambda x\varphi\lambda y\psi, \Pi \Rightarrow \Sigma}\ (\lambda \Rightarrow 1)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)$$

we apply substitution lemma to premiss of $(\lambda \Rightarrow 1)$ to replace the occurrences of fresh d with c, then we continue:

$$\cfrac{\cfrac{\Gamma \Rightarrow \Delta, c\lambda y\psi \qquad \cfrac{\Gamma \Rightarrow \Delta, c\lambda x\varphi \qquad c\lambda x\varphi, c\lambda y\psi, \Pi \Rightarrow \Sigma}{c\lambda y\psi, \Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)}{\Gamma, \Gamma, \Pi \Rightarrow \Delta, \Delta, \Sigma}\ (Cut)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (C \Rightarrow), (\Rightarrow C)$$

Both cuts are of lower degree, hence both rules are reductive.

CUT ELIMINATION IN ELO - reductivity of λ-rules:

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$\dfrac{(\Rightarrow \lambda)\ \dfrac{a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi\lambda y\psi} \qquad \dfrac{\Pi \Rightarrow \Sigma, c\lambda x\varphi \qquad \Pi \Rightarrow \Sigma, d\lambda x\varphi \qquad cd, \Pi \Rightarrow \Sigma}{\lambda x\varphi\lambda y\psi, \Pi \Rightarrow \Sigma}\ (\lambda \Rightarrow 2)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)$$

where on the left side we display only one (relevant) premiss.

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$
\dfrac{(\Rightarrow \lambda)\ \dfrac{a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi\lambda y\psi} \qquad \dfrac{\Pi \Rightarrow \Sigma, c\lambda x\varphi \qquad \Pi \Rightarrow \Sigma, d\lambda x\varphi \qquad cd, \Pi \Rightarrow \Sigma}{\lambda x\varphi\lambda y\psi, \Pi \Rightarrow \Sigma}\ (\lambda \Rightarrow 2)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (Cut)
$$

where on the left side we display only one (relevant) premiss.

We apply substitution lemma (twice) to the rightmost premiss of the application of $(\Rightarrow \lambda)$ instead, to replace the occurrences of fresh a, b with c, d respectively, then we continue:

$$
\dfrac{\dfrac{\dfrac{\Pi \Rightarrow \Sigma, d\lambda x\varphi \qquad \dfrac{\Pi \Rightarrow \Sigma, c\lambda x\varphi \qquad c\lambda x\varphi, d\lambda x\varphi, \Gamma \Rightarrow \Delta, cd}{d\lambda x\varphi, \Gamma, \Pi \Rightarrow \Delta, \Sigma, cd}\ (Cut)}{\Gamma, \Pi, \Pi \Rightarrow \Delta, \Sigma, \Sigma, cd}\ (Cut) \qquad cd, \Pi \Rightarrow \Sigma}{\Gamma, \Pi, \Pi, \Pi \Rightarrow \Delta, \Sigma, \Sigma, \Sigma}\ (Cut)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\ (C \Rightarrow), (\Rightarrow C)
$$

All cuts are of lower degree, hence both rules are reductive.

CUT ELIMINATION IN ELO - reductivity of λ-rules:

$$\dfrac{\dfrac{a\lambda x\varphi, b\lambda x\varphi, \Gamma \Rightarrow \Delta, ab}{\Gamma \Rightarrow \Delta, \lambda x\varphi\lambda y\psi}\,(\Rightarrow \lambda) \qquad \dfrac{\Pi \Rightarrow \Sigma, c\lambda x\varphi \qquad \Pi \Rightarrow \Sigma, d\lambda x\varphi \qquad cd, \Pi \Rightarrow \Sigma}{\lambda x\varphi\lambda y\psi, \Pi \Rightarrow \Sigma}\,(\lambda \Rightarrow 2)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\,(Cut)$$

where on the left side we display only one (relevant) premiss.

We apply substitution lemma (twice) to the rightmost premiss of the application of $(\Rightarrow \lambda)$ instead, to replace the occurrences of fresh a, b with c, d respectively, then we continue:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi \Rightarrow \Sigma, d\lambda x\varphi \qquad \dfrac{\Pi \Rightarrow \Sigma, c\lambda x\varphi \qquad c\lambda x\varphi, d\lambda x\varphi, \Gamma \Rightarrow \Delta, cd}{d\lambda x\varphi, \Gamma, \Pi \Rightarrow \Delta, \Sigma, cd}\,(Cut)}{\Gamma, \Pi, \Pi \Rightarrow \Delta, \Sigma, \Sigma, cd}\,(Cut) \qquad cd, \Pi \Rightarrow \Sigma}{\Gamma, \Pi, \Pi, \Pi \Rightarrow \Delta, \Sigma, \Sigma, \Sigma}\,(Cut)}{\Gamma, \Pi \Rightarrow \Delta, \Sigma}\,(C \Rightarrow), (\Rightarrow C)}{}$$

All cuts are of lower degree, hence both rules are reductive.

But it does not work for GELO_s!

CUT ELIMINATION IN ELO

Theorem

Every proof in $GELO_w$ and $GELO_m$ can be transformed into a cut-free proof.

CUT ELIMINATION IN ELO

Theorem

Every proof in $GELO_w$ and $GELO_m$ can be transformed into a cut-free proof.

Corollary

If $\vdash \Gamma \Rightarrow \Delta$ in $GELO_w$ or $GELO_m$, then it is provable in a proof which is closed under subformulae of $\Gamma \cup \Delta$ and atomic formulae with possibly new parameters.

ELO - CONCLUDING REMARKS

ELO - CONCLUDING REMARKS

Open problems and further developments:

ELO - CONCLUDING REMARKS

Open problems and further developments:

1. Better solution for $GELO_s$ - satisfying cut elimination.

ELO - CONCLUDING REMARKS

Open problems and further developments:

1. Better solution for $GELO_s$ - satisfying cut elimination.
2. Proving Interpolation for ELO.

Open problems and further developments:

1. Better solution for $GELO_s$ - satisfying cut elimination.
2. Proving Interpolation for ELO.
3. Changing the additional linguistic component of ELO (e.g. DL or relational syllogistics) and its grammatical status (e.g. instead of fusion with the language of LO, introduce the second component only inside lambda terms).

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.