Cut-Simulation in Impredicative Logics

Christoph E. Benzmüller

and

Chad E. Brown and Michael Kohlhase

Saarland University and International University Bremen

IJCAR 2006, Seattle, US

UNIVERSITÄT DES SAARLANDES

Many Connected Stories in the Paper

- Study challenges for automating impredicate logics/HOL
  - Leibniz-equations
  - axioms of:
    comprehension, extensionality, induction,
    choice/description, etc.

Many Connected Stories in the Paper

- Study challenges for automating impredicate logics/HOL
  - Leibniz-equations
  - axioms of:
    comprehension, extensionality, induction,
    choice/description, etc.
- Cut-simulation
- Is ‘cut-freeness’ a meaningful criterion in our context?

Many Connected Stories in the Paper

- Study challenges for automating impredicate logics/HOL
  - Leibniz-equations
  - axioms of:
    comprehension, extensionality, induction,
    choice/description, etc.
- Cut-simulation
- Is ‘cut-freeness’ a meaningful criterion in our context?
- What is the connection between ‘cut’ and ‘saturation’?
- Abstract cut-elimination result for HOL

Many Connected Stories in the Paper

- Study challenges for automating impredicate logics/HOL
  - Leibniz-equations
  - axioms of:
    comprehension, extensionality, induction, choice/description, etc.
- Cut-simulation
- Is ‘cut-freeness’ a meaningful criterion in our context?
- What is the connection between ‘cut’ and ‘saturation’?
- Abstract cut-elimination result for HOL
- . . .

Difficulty for AR: Impredicativity

Notion: (Impredicativity)

- quantification over sets and predicates
- support impredicative definitions and reflection

Difficulty for AR: Impredicativity

Notion: (Impredicativity)

- quantification over sets and predicates
- support impredicative definitions and reflection

Ex.: Automation already problematic for very simple quantifications over sets: $\exists P\,(P\ 1)$

Difficulty for AR: Impredicativity

Notion: (Impredicativity)

- quantification over sets and predicates
- support impredicative definitions and reflection

Ex.: Automation already problematic for very simple quantifications over sets: $\exists \mathsf{P}\,(\mathsf{P}\ 1)$

- $\mathsf{P} \longleftarrow \{\mathsf{x} | \mathsf{true}\}$ $\qquad (\lambda \mathsf{X}\ \mathbf{T}_{\mathsf{o}})$
- $\mathsf{P} \longleftarrow \{\mathsf{x} | \mathsf{x} = 1\}$ $\qquad (\lambda \mathsf{X}\ \mathsf{X} = 1)$
- $\mathsf{P} \longleftarrow \{\mathsf{x} | \mathsf{x} = 1 \vee \mathsf{x} = 2\}$ $\qquad (\lambda \mathsf{X}\ \mathsf{X} = 1 \vee \mathsf{X} = 2)$
- $\mathsf{P} \longleftarrow \{\mathsf{x} | \mathsf{x} > 0\}$ $\qquad (\lambda \mathsf{X}\ \mathsf{X} > 0)$
- etc.

Difficulty for AR: Impredicativity

Notion: (Impredicativity)

- quantification over sets and predicates
- support impredicative definitions and reflection

Ex.: Automation already problematic for very simple quantifications over sets: $\exists P\,(P\ 1)$

- $P \longleftarrow \{x|\mathrm{true}\}$ $\qquad (\lambda X\ \mathbf{T}_o)$
- $P \longleftarrow \{x|x = 1\}$ $\qquad (\lambda X\ X = 1)$
- $P \longleftarrow \{x|x = 1 \vee x = 2\}$ $\qquad (\lambda X\ X = 1 \vee X = 2)$
- $P \longleftarrow \{x|x > 0\}$ $\qquad (\lambda X\ X > 0)$
- etc.

- unification not powerful enough $\Longrightarrow$ guessing is state of the art
- problem not limited to HOL

HOL: Simple Types

Simple Types $\mathcal{T}$:

o	(truth values)
ι	(individuals)
$(\alpha \longrightarrow \beta)$	(functions from α to β)

HOL: Simply Typed λ-Terms

Typed Terms:

X_α	Variables ($\mathcal{V}$)
a_α	Parameters ($\mathcal{P}$)
c_α	Constants (Σ)
$(\mathbf{F}_{\alpha\to\beta}\,\mathbf{B}_\alpha)_\beta$	Application
$(\lambda \mathsf{Y}_\alpha\,\mathbf{A}_\beta)_{\alpha\to\beta}$	λ-abstraction

HOL: Simply Typed λ-Terms

Typed Terms:

X_α	Variables ($\mathcal{V}$)
a_α	Parameters ($\mathcal{P}$)
c_α	Constants (Σ)
$(\mathbf{F}_{\alpha\rightarrow\beta}\,\mathbf{B}_\alpha)_\beta$	Application
$(\lambda \mathsf{Y}_\alpha\,\mathbf{A}_\beta)_{\alpha\rightarrow\beta}$	λ-abstraction

Equality of terms:

α-conversion	Changing bound variables	
β-reduction	$((\lambda \mathsf{Y}_\beta\,\mathbf{A}_\alpha)\,\mathbf{B}) \xrightarrow{\beta} [\mathbf{B}/\mathsf{Y}]\mathbf{A}$	
η-reduction	$(\lambda \mathsf{Y}_\alpha\,(\mathbf{F}_{\alpha\rightarrow\beta}\,\mathsf{Y})) \xrightarrow{\eta} \mathbf{F}$	$(\mathsf{Y}_\beta \notin \mathbf{Free}(\mathbf{F}))$

HOL: Adding Logical Constants to Σ

- $\top_o$ – true
- $\bot_o$ – false
- $\neg_{o\to o}$ – negation
- $\vee_{o\to o\to o}$ – disjunction
- $\wedge_{o\to o\to o}$ – conjunction
- $\supset_{o\to o\to o}$ – implication
- $\Leftrightarrow_{o\to o\to o}$ – equivalence
- $\Pi^{\alpha}_{(\alpha\to o)\to o}$ – universal quantification over type α ($\forall$ types α)
- $\Sigma^{\alpha}_{(\alpha\to o)\to o}$ – existential quantification over type α ($\forall$ types α)
- $=^{\alpha}_{\alpha\to\alpha\to o}$ – equality at type α ($\forall$ types α)

HOL: Adding Logical Constants to Σ

Our choice for signature Σ in this paper:

- $\neg_{o \to o}$ – negation
- $\vee_{o \to o \to o}$ – disjunction
- $\Pi^{\alpha}_{o(o\alpha)}$ – universal quantification over type α ($\forall$ types α)

HOL: Adding Logical Constants to Σ

Our choice for signature Σ in this paper:

- $\neg_{o \to o}$ – negation
- $\vee_{o \to o \to o}$ – disjunction
- $\Pi^{\alpha}_{o(o\alpha)}$ – universal quantification over type α ($\forall$ types α)

Use abbreviations for other logical operators

$\mathbf{A} \vee \mathbf{B}$	means	$(\vee\, \mathbf{A}\, \mathbf{B})$
$\mathbf{A} \wedge \mathbf{B}$	means	$\neg(\neg\mathbf{A} \vee \neg\mathbf{B})$
$\mathbf{A} \supset \mathbf{B}$	means	$\neg\mathbf{A} \vee \mathbf{B}$
$\mathbf{A} \Leftrightarrow \mathbf{B}$	means	$(\mathbf{A} \supset \mathbf{B}) \wedge (\mathbf{B} \supset \mathbf{A})$
$\forall X_{\alpha}\, \mathbf{A}$	means	$\Pi^{\alpha}(\lambda X_{\alpha}\, \mathbf{A})$
$\exists X_{\alpha}\, \mathbf{A}$	means	$\neg(\forall X_{\alpha}\, \neg\mathbf{A})$

HOL: Adding Logical Constants to Σ

Our choice for signature Σ in this paper:

- $\neg_{o \to o}$ – negation
- $\vee_{o \to o \to o}$ – disjunction
- $\Pi^{\alpha}_{o(o\alpha)}$ – universal quantification over type α (∀ types α)

Use Leibniz-equality to encode equality

$$\mathbf{A}_\alpha \doteq^\alpha \mathbf{B}_\alpha \quad \text{means} \quad \forall P_{\alpha \to o} (P\, \mathbf{A} \supset P\, \mathbf{B})$$

$$\text{resp.} \quad \Pi^\alpha (\lambda P_{\alpha \to o} \neg P \mathbf{A} \vee P \mathbf{B})$$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{}{\Gamma \Longrightarrow \Delta, \mathbf{A} \vee \mathbf{B}}\ \mathcal{G}(\vee_{Intro})$$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \Delta, \mathbf{A}, \mathbf{B}}{\Gamma \Longrightarrow \Delta, \mathbf{A} \lor \mathbf{B}} \mathcal{G}(\lor_{Intro})$$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A},\mathbf{B}}{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\ \mathcal{G}(\vee_{Intro})$$

- corresponding one-sided rules:

$$\frac{}{\neg(\Gamma) \cup \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\ \mathcal{G}(\vee_{+})$$

$\mathbf{\Delta},\mathbf{A}$ stands for $\mathbf{\Delta} \cup \{\mathbf{A}\}$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A},\mathbf{B}}{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{Intro})$$

- corresponding one-sided rules:

$$\frac{\neg(\Gamma) \cup \mathbf{\Delta},\mathbf{A},\mathbf{B}}{\neg(\Gamma) \cup \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{+})$$

$\mathbf{\Delta},\mathbf{A}$ stands for $\mathbf{\Delta} \cup \{\mathbf{A}\}$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A},\mathbf{B}}{\Gamma \Longrightarrow \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{Intro}) \qquad \frac{}{\Gamma,\mathbf{A} \vee \mathbf{B} \Longrightarrow \Delta}\,\mathcal{G}(\vee_{Elim})$$

- corresponding one-sided rules:

$$\frac{\neg(\Gamma) \cup \mathbf{\Delta},\mathbf{A},\mathbf{B}}{\neg(\Gamma) \cup \mathbf{\Delta},\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{+})$$

$\Delta,\mathbf{A}$ stands for $\Delta \cup \{\mathbf{A}\}$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \Delta,\mathbf{A},\mathbf{B}}{\Gamma \Longrightarrow \Delta,\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{Intro}) \qquad \frac{\Gamma,\mathbf{A} \Longrightarrow \Delta \quad \Gamma,\mathbf{B} \Longrightarrow \Delta}{\Gamma,\mathbf{A} \vee \mathbf{B} \Longrightarrow \Delta}\,\mathcal{G}(\vee_{Elim})$$

- corresponding one-sided rules:

$$\frac{\neg(\Gamma) \cup \Delta,\mathbf{A},\mathbf{B}}{\neg(\Gamma) \cup \Delta,\mathbf{A} \vee \mathbf{B}}\,\mathcal{G}(\vee_{+})$$

$\Delta,\mathbf{A}$ stands for $\Delta \cup \{\mathbf{A}\}$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \Delta, \mathbf{A}, \mathbf{B}}{\Gamma \Longrightarrow \Delta, \mathbf{A} \lor \mathbf{B}}\,\mathcal{G}(\lor_{Intro}) \qquad \frac{\Gamma, \mathbf{A} \Longrightarrow \Delta \quad \Gamma, \mathbf{B} \Longrightarrow \Delta}{\Gamma, \mathbf{A} \lor \mathbf{B} \Longrightarrow \Delta}\,\mathcal{G}(\lor_{Elim})$$

- corresponding one-sided rules:

$$\frac{\neg(\Gamma) \cup \Delta, \mathbf{A}, \mathbf{B}}{\neg(\Gamma) \cup \Delta, \mathbf{A} \lor \mathbf{B}}\,\mathcal{G}(\lor_{+}) \qquad \frac{}{\neg(\Gamma) \cup \Delta, \neg(\mathbf{A} \lor \mathbf{B})}\,\mathcal{G}(\lor_{-})$$

$\Delta, \mathbf{A}$ stands for $\Delta \cup \{\mathbf{A}\}$

A Sequent Calculus for HOL

We work with a one-sided sequent calculus:

- examples for two-sided rules:

$$\frac{\Gamma \Longrightarrow \Delta, \mathbf{A}, \mathbf{B}}{\Gamma \Longrightarrow \Delta, \mathbf{A} \lor \mathbf{B}} \mathcal{G}(\lor_{Intro}) \qquad \frac{\Gamma, \mathbf{A} \Longrightarrow \Delta \quad \Gamma, \mathbf{B} \Longrightarrow \Delta}{\Gamma, \mathbf{A} \lor \mathbf{B} \Longrightarrow \Delta} \mathcal{G}(\lor_{Elim})$$

- corresponding one-sided rules:

$$\frac{\neg(\Gamma) \cup \Delta, \mathbf{A}, \mathbf{B}}{\neg(\Gamma) \cup \Delta, \mathbf{A} \lor \mathbf{B}} \mathcal{G}(\lor_{+}) \qquad \frac{\neg(\Gamma) \cup \Delta, \neg\mathbf{A} \quad \neg(\Gamma) \cup \Delta, \neg\mathbf{B}}{\neg(\Gamma) \cup \Delta, \neg(\mathbf{A} \lor \mathbf{B})} \mathcal{G}(\lor_{-})$$

$\Delta, \mathbf{A}$ stands for $\Delta \cup \{\mathbf{A}\}$

Sequent Calculus Rules

Def.: (Basic Rules)

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{}{\mathbf{\Delta}, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\,\mathcal{G}(init) \qquad\qquad \frac{}{\Delta, \neg\neg\mathbf{A}}\,\mathcal{G}(\neg)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\,\mathcal{G}(init) \qquad\qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\,\mathcal{G}(\neg)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init) \qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\ \mathcal{G}(\neg)$$

$$\frac{\Delta, \neg\mathbf{A} \quad \Delta, \neg\mathbf{B}}{\Delta, \neg(\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_-) \qquad \frac{\Delta, \mathbf{A}, \mathbf{B}}{\Delta, (\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_+)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init) \qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\ \mathcal{G}(\neg)$$

$$\frac{\Delta, \neg\mathbf{A} \quad \Delta, \neg\mathbf{B}}{\Delta, \neg(\mathbf{A} \lor \mathbf{B})}\ \mathcal{G}(\lor_-) \qquad \frac{\Delta, \mathbf{A}, \mathbf{B}}{\Delta, (\mathbf{A} \lor \mathbf{B})}\ \mathcal{G}(\lor_+)$$

$$\frac{}{\Delta, \neg\forall X_\alpha \mathbf{A} \ (:= \neg\Pi^\alpha \mathbf{A})}\ \mathcal{G}(\Pi_-^{\mathbf{C}})$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\,\mathcal{G}(init) \qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\,\mathcal{G}(\neg)$$

$$\frac{\Delta, \neg\mathbf{A} \quad \Delta, \neg\mathbf{B}}{\Delta, \neg(\mathbf{A} \vee \mathbf{B})}\,\mathcal{G}(\vee_-) \qquad \frac{\Delta, \mathbf{A}, \mathbf{B}}{\Delta, (\mathbf{A} \vee \mathbf{B})}\,\mathcal{G}(\vee_+)$$

$$\frac{\Delta, \neg(\mathbf{A}\mathbf{C}){\downarrow_\beta} \quad \mathbf{C} \in \mathit{cwff}_\alpha(\Sigma)}{\Delta, \neg\forall \mathrm{X}_\alpha \mathbf{A} \ (:= \neg\Pi^\alpha \mathbf{A})}\,\mathcal{G}(\Pi^{\mathbf{C}}_-)$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init) \qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\ \mathcal{G}(\neg)$$

$$\frac{\Delta, \neg\mathbf{A} \quad \Delta, \neg\mathbf{B}}{\Delta, \neg(\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_{-}) \qquad \frac{\Delta, \mathbf{A}, \mathbf{B}}{\Delta, (\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_{+})$$

$$\frac{\Delta, \neg(\mathbf{A}\mathbf{C}){\downarrow_\beta} \quad \mathbf{C} \in \mathit{cwff}_\alpha(\Sigma)}{\Delta, \neg\forall X_\alpha \mathbf{A} \ (:= \neg\Pi^\alpha \mathbf{A})}\ \mathcal{G}(\Pi_{-}^{\mathbf{C}}) \qquad \frac{}{\Delta, \forall X_\alpha \mathbf{A} \ (:= \Pi^\alpha \mathbf{A})}\ \mathcal{G}(\Pi_{+}^{c})$$

Sequent Calculus Rules

Def.: (Basic Rules)

$$\frac{\mathbf{A} \text{ atomic (and } \beta\text{-normal)}}{\Delta, \neg\mathbf{A}, \mathbf{A}}\ \mathcal{G}(init) \qquad \frac{\Delta, \mathbf{A}}{\Delta, \neg\neg\mathbf{A}}\ \mathcal{G}(\neg)$$

$$\frac{\Delta, \neg\mathbf{A} \quad \Delta, \neg\mathbf{B}}{\Delta, \neg(\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_-) \qquad \frac{\Delta, \mathbf{A}, \mathbf{B}}{\Delta, (\mathbf{A} \vee \mathbf{B})}\ \mathcal{G}(\vee_+)$$

$$\frac{\Delta, \neg(\mathbf{A}\mathbf{C}){\downarrow_\beta} \quad \mathbf{C} \in cwff_\alpha(\Sigma)}{\Delta, \neg\forall X_\alpha \mathbf{A}\ (:= \neg\Pi^\alpha \mathbf{A})}\ \mathcal{G}(\Pi_-^{\mathbf{C}}) \qquad \frac{\Delta, (\mathbf{A}c){\downarrow_\beta} \quad c_\alpha \in \Sigma \text{ new}}{\Delta, \forall X_\alpha \mathbf{A}\ (:= \Pi^\alpha \mathbf{A})}\ \mathcal{G}(\Pi_+^c)$$

The Sequent Calculus $\mathcal{G}_\beta$

Def.: The sequent calculus $\mathcal{G}_\beta$ is defined by the rules

$$\mathcal{G}(init), \mathcal{G}(\neg), \mathcal{G}(\vee_-), \mathcal{G}(\vee_+), \mathcal{G}(\Pi^{\mathbf{C}}_-), \mathcal{G}(\Pi^c_+)$$

The Sequent Calculus $\mathcal{G}_\beta$

Def.: The sequent calculus $\mathcal{G}_\beta$ is defined by the rules

$$\mathcal{G}(init), \mathcal{G}(\neg), \mathcal{G}(\vee_-), \mathcal{G}(\vee_+), \mathcal{G}(\Pi_-^{\mathbf{C}}), \mathcal{G}(\Pi_+^c)$$

- Analysis of admissibility of cut:

$$\frac{\Delta, \mathbf{C} \quad \Delta, \neg\mathbf{C}}{\Delta}\ \mathcal{G}(cut)$$

The Sequent Calculus $\mathcal{G}_\beta$

Def.: The sequent calculus $\mathcal{G}_\beta$ is defined by the rules

$$\mathcal{G}(init), \mathcal{G}(\neg), \mathcal{G}(\vee_-), \mathcal{G}(\vee_+), \mathcal{G}(\Pi^{\mathbf{C}}_-), \mathcal{G}(\Pi^c_+)$$

- Analysis of admissibility of cut:

$$\frac{\Delta, \mathbf{C} \quad \Delta, \neg\mathbf{C}}{\Delta}\ \mathcal{G}(cut)$$

- Analysis of Soundness and Completeness:
  - we need appropriate notions of semantics for HOL
  - standard semantics not appropriate

Semantics: HOL-CUBE

$\mathfrak{M}_{\beta}(\Sigma)$ elementary type theory (Andrews)

$\mathfrak{b}$: **Boolean extensionality** $|\mathcal{D}_o| = 2$

$\mathfrak{f}(= \eta + \xi)$: **functional extensionality**

see [Journal of Symbolic Logic (2004) 69(4)]

$\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma)$ extensional type theory (Henkin semantics)

full function domains

$\mathfrak{S}\mathfrak{T}(\Sigma)$ **standard semantics**

Semantics: HOL-CUBE

$$
\begin{tikzcd}
\mathfrak{M}_{\beta}(\Sigma) \arrow[dr, "\mathfrak{b}"] \arrow[dd, dotted, "{\mathfrak{b}, \mathfrak{f}(= \eta + \xi)}"'] & \\
& \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[dl, "\mathfrak{f}"] \\
\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, "\text{full function domains}"] & \\
\mathfrak{S}\mathfrak{T}(\Sigma) &
\end{tikzcd}
$$

elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE

$$
\begin{tikzcd}
\mathfrak{M}_{\beta}(\Sigma) \arrow[dr, "\mathfrak{b}"] \arrow[dd, dotted, "{\mathfrak{b}, \mathfrak{f}(= \eta + \xi)}"'] & \\
& \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[dl, "\mathfrak{f}"] \\
\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, dotted, "\text{full function domains}"] & \\
\mathfrak{S}\mathfrak{T}(\Sigma) &
\end{tikzcd}
$$

elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE

$$
\begin{tikzcd}
& \mathfrak{M}_{\beta}(\Sigma) \arrow[dl, "\mathfrak{f}"'] \arrow[dr, "\mathfrak{b}"] \arrow[dd, dotted, "{\mathfrak{b}, \mathfrak{f}(= \eta + \xi)}" description] & \\
\mathfrak{M}_{\beta\mathfrak{f}}(\Sigma) \arrow[dr, "\mathfrak{b}"'] & & \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[dl, "\mathfrak{f}"] \\
& \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, dotted, "\text{full function domains}"] & \\
& \mathfrak{S}\mathfrak{T}(\Sigma) &
\end{tikzcd}
$$

elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE

$$
\begin{tikzcd}
& \mathfrak{M}_{\beta}(\Sigma) \arrow[dl, "\mathfrak{f}"'] \arrow[dr, "\mathfrak{b}"] \arrow[dd, dotted, "{\mathfrak{b},\, \mathfrak{f}(=\eta+\xi)}" description] & \\
\mathfrak{M}_{\beta\mathfrak{f}}(\Sigma) \arrow[dr, "\mathfrak{b}"'] & & \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[dl, "\mathfrak{f}"] \\
& \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, dotted, "\text{full function domains}" description] & \\
& \mathfrak{S}\mathfrak{T}(\Sigma) &
\end{tikzcd}
$$

elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE

$$
\begin{tikzcd}
 & \mathfrak{M}_{\beta}(\Sigma) \arrow[dl, "\xi"'] \arrow[dr, "\mathfrak{b}"] \arrow[ddl, dotted, "\mathfrak{f}"] & \\
\mathfrak{M}_{\beta\xi}(\Sigma) \arrow[d, "\eta"] & & \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[ddl, "\mathfrak{f}"] \\
\mathfrak{M}_{\beta\mathfrak{f}}(\Sigma) \arrow[dr, "\mathfrak{b}"] & & \\
 & \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, dotted, "\text{full function domains}"] & \\
 & \mathfrak{S}\mathfrak{T}(\Sigma) &
\end{tikzcd}
$$

elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE

$$
\begin{tikzcd}
 & & \mathfrak{M}_{\beta}(\Sigma) \arrow[dll, "\xi"'] \arrow[ddll, dotted, "\mathfrak{f}"] \arrow[drr, "\mathfrak{b}"] & & \\
\mathfrak{M}_{\beta\xi}(\Sigma) \arrow[d, "\eta"] & & & & \mathfrak{M}_{\beta\mathfrak{b}}(\Sigma) \arrow[dll, "\xi"'] \arrow[ddll, dotted, "\mathfrak{f}"] \\
\mathfrak{M}_{\beta\mathfrak{f}}(\Sigma) \arrow[drr, "\mathfrak{b}"] & & \mathfrak{M}_{\beta\xi\mathfrak{b}}(\Sigma) \arrow[d, "\eta"] & & \\
 & & \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma) \simeq \mathfrak{H}(\Sigma) \arrow[d, dotted, "\text{full function domains}"] & & \\
 & & \mathfrak{S}\mathfrak{T}(\Sigma) & &
\end{tikzcd}
$$

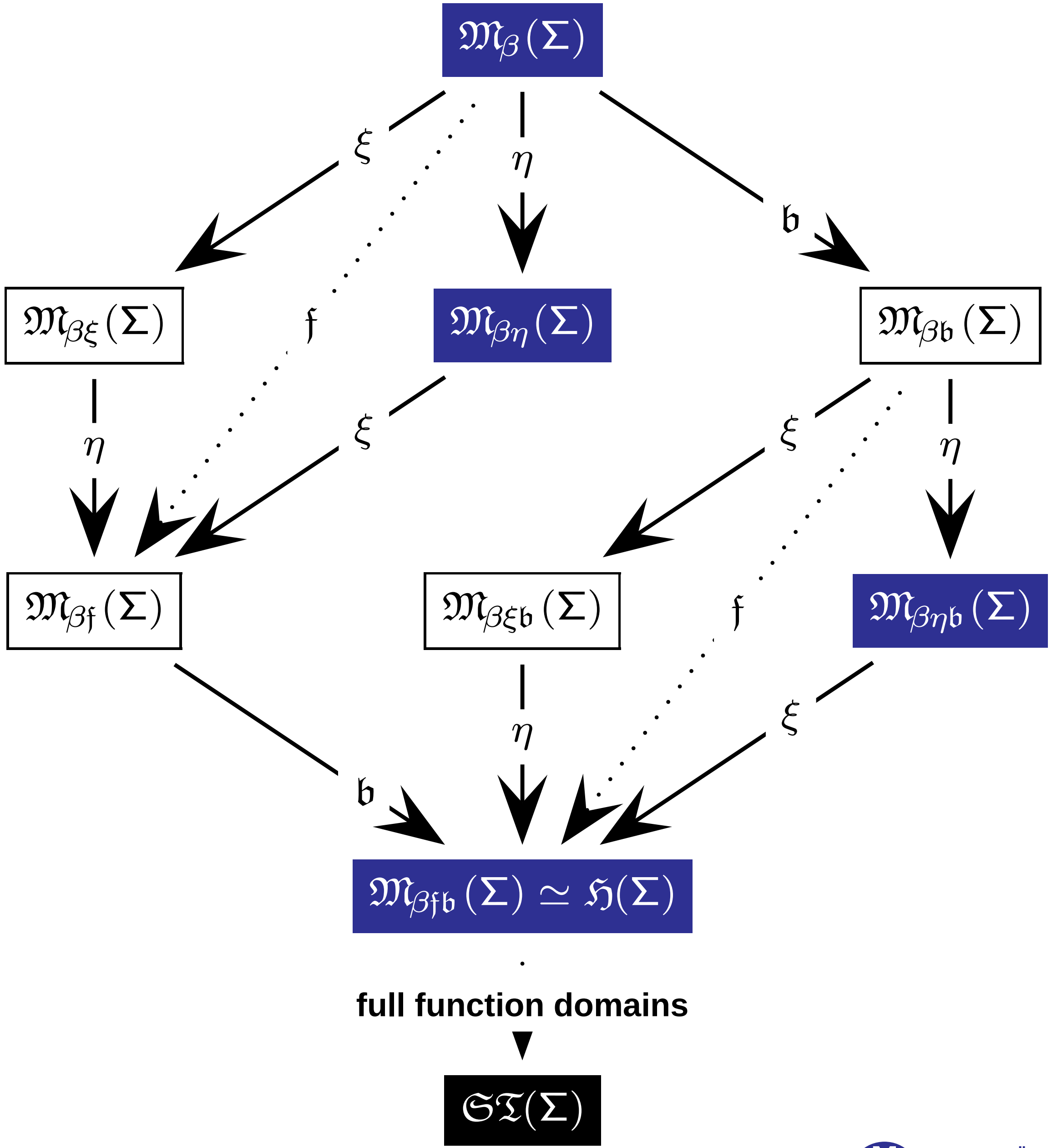
elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE



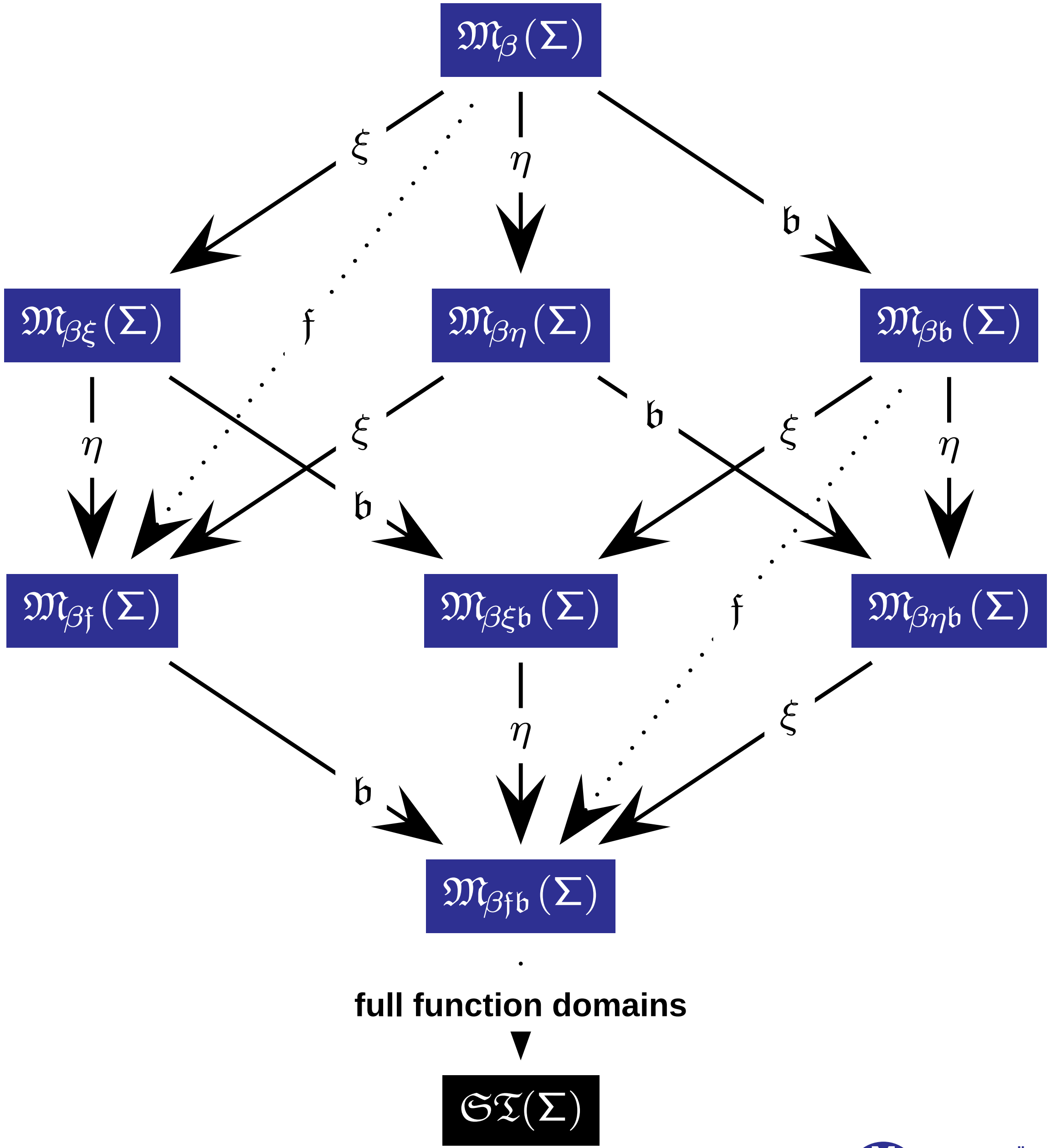
elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

extensional type theory (Henkin semantics)

standard semantics

Semantics: HOL-CUBE



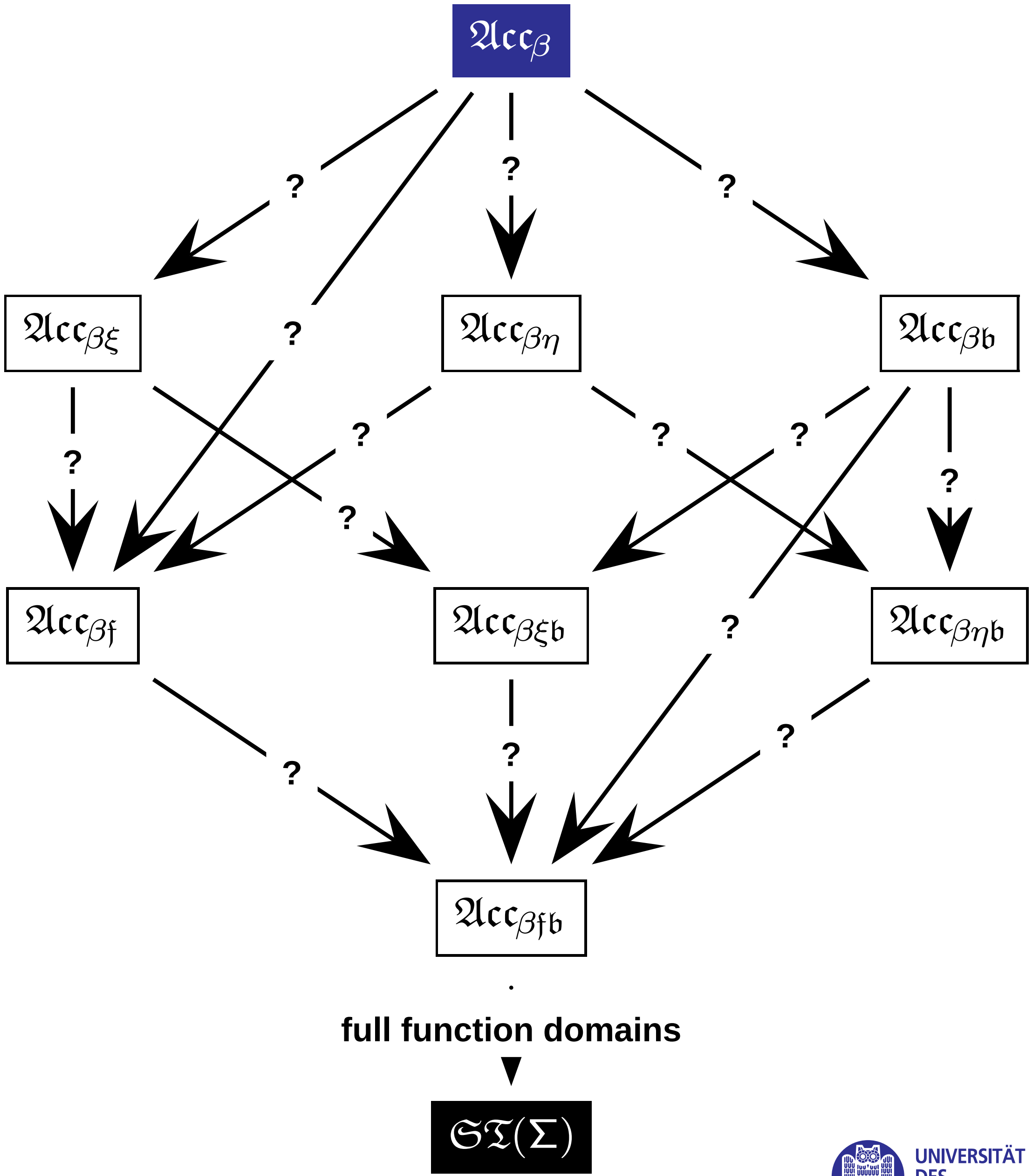
elementary type theory (Andrews)

see [Journal of Symbolic Logic (2004) 69(4)]

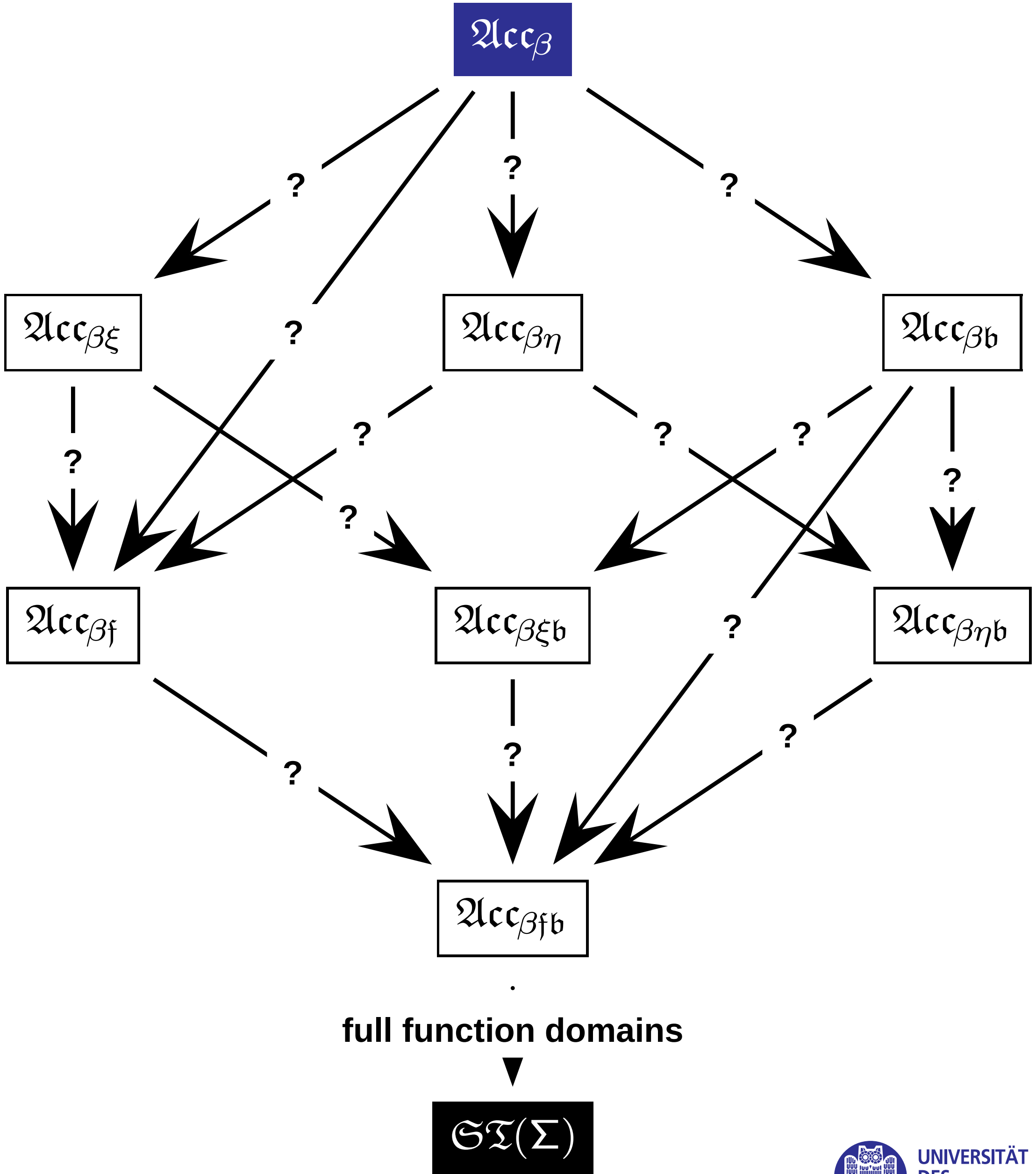
extensional type theory (Henkin semantics)

standard semantics

Abstract Consistency Proof Method



Abstract Consistency Proof Method



Properties for $\mathfrak{Acc}_\beta$: (Γ_Σ is class of sets of formulas; $\Phi \in \Gamma_\Sigma$)

∇_c If $\mathbf{A}$ is atomic, then $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$.

$\nabla_\neg$ If $\neg\neg\mathbf{A} \in \Phi$, then $\Phi,\mathbf{A} \in \Gamma_\Sigma$.

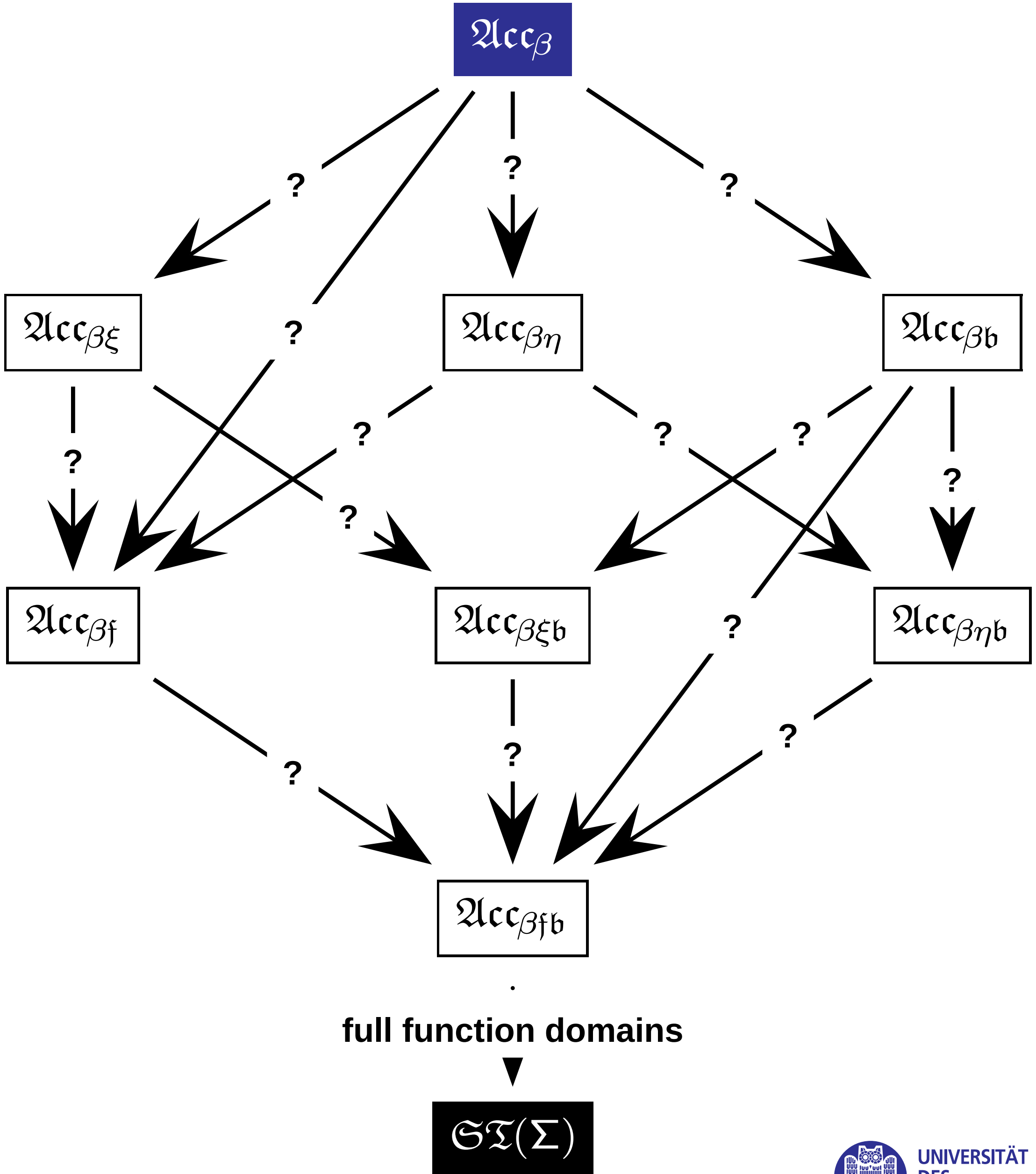
$\nabla_\vee$ If $\mathbf{A} \vee \mathbf{B} \in \Phi$, then $\Phi,\mathbf{A} \in \Gamma_\Sigma$ or $\Phi,\mathbf{B} \in \Gamma_\Sigma$.

$\nabla_\wedge$ If $\neg(\mathbf{A} \vee \mathbf{B}) \in \Phi$, then $\Phi,\neg\mathbf{A},\neg\mathbf{B} \in \Gamma_\Sigma$.

$\nabla_\forall$ If $\Pi^\alpha\mathbf{F} \in \Phi$, then $\Phi,\mathbf{F}\mathbf{W} \in \Gamma_\Sigma$ for each $\mathbf{W} \in cwff_\alpha(\Sigma)$.

$\nabla_\exists$ If $\neg\Pi^\alpha\mathbf{F} \in \Phi$, then $\Phi,\neg(\mathbf{F}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.

Abstract Consistency Proof Method

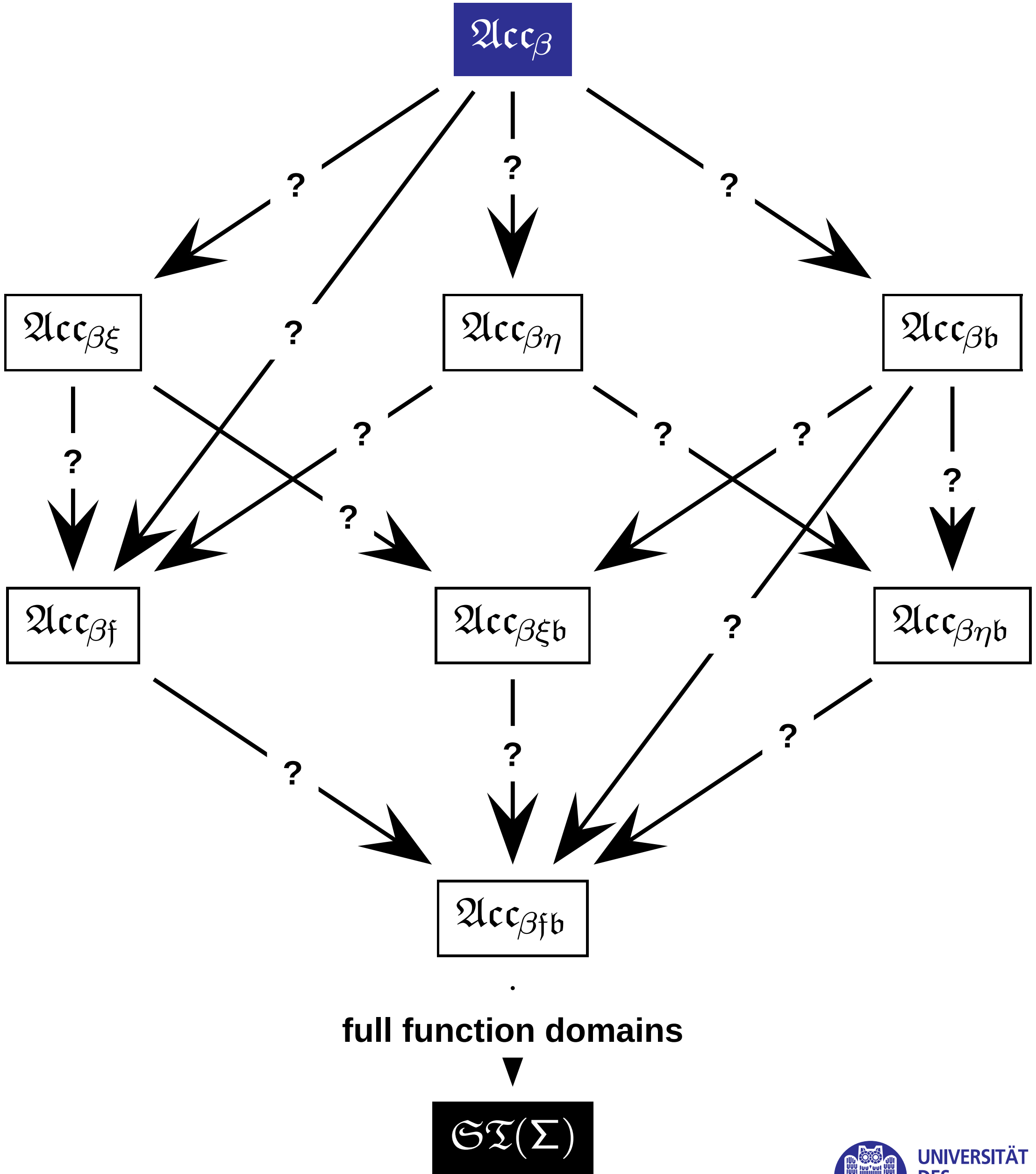


Properties for $\mathfrak{Acc}_\beta$: (Γ_Σ is class of sets of formulas; $\Phi \in \Gamma_\Sigma$)

- ∇_c If $\mathbf{A}$ is atomic, then $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$.
- $\nabla_\neg$ If $\neg\neg\mathbf{A} \in \Phi$, then $\Phi,\mathbf{A} \in \Gamma_\Sigma$.
- ∇_β If $\mathbf{A} =_\beta \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi,\mathbf{B} \in \Gamma_\Sigma$.
- $\nabla_\vee$ If $\mathbf{A} \vee \mathbf{B} \in \Phi$, then $\Phi,\mathbf{A} \in \Gamma_\Sigma$ or $\Phi,\mathbf{B} \in \Gamma_\Sigma$.
- $\nabla_\wedge$ If $\neg(\mathbf{A} \vee \mathbf{B}) \in \Phi$, then $\Phi,\neg\mathbf{A},\neg\mathbf{B} \in \Gamma_\Sigma$.
- $\nabla_\forall$ If $\Pi^\alpha\mathbf{F} \in \Phi$, then $\Phi,\mathbf{F}\mathbf{W} \in \Gamma_\Sigma$ for each $\mathbf{W} \in cwff_\alpha(\Sigma)$.
- $\nabla_\exists$ If $\neg\Pi^\alpha\mathbf{F} \in \Phi$, then $\Phi,\neg(\mathbf{F}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.

UNIVERSITÄT DES SAARLANDES

Abstract Consistency

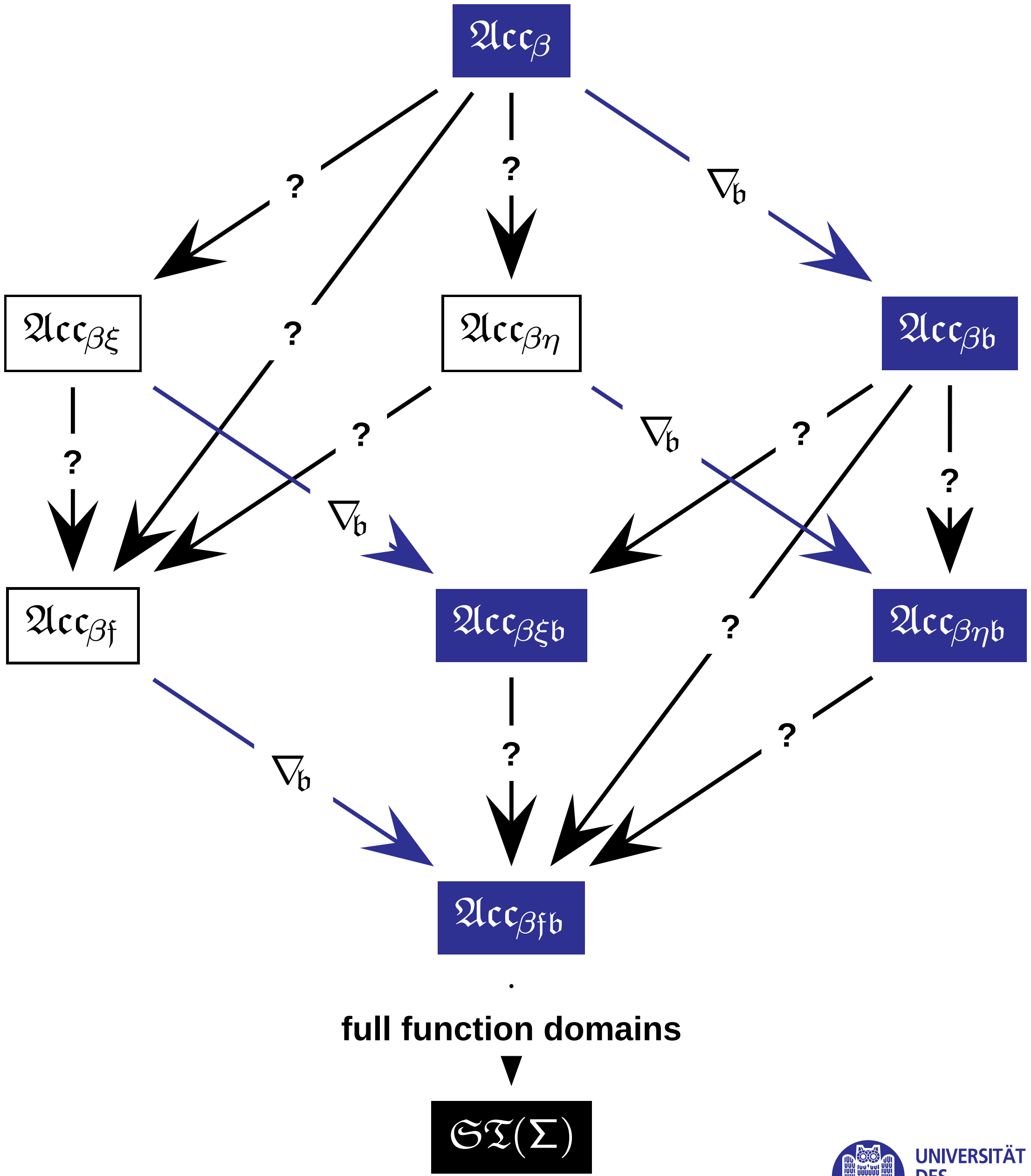


Properties for $\mathfrak{Acc}_\beta$

∇_c	...	$\nabla_\vee$	...
$\nabla_\neg$	...	$\nabla_\wedge$	...
∇_β	...	$\nabla_\forall$	...
		$\nabla_\exists$	...

Properties for Extensionality

Abstract Consistency



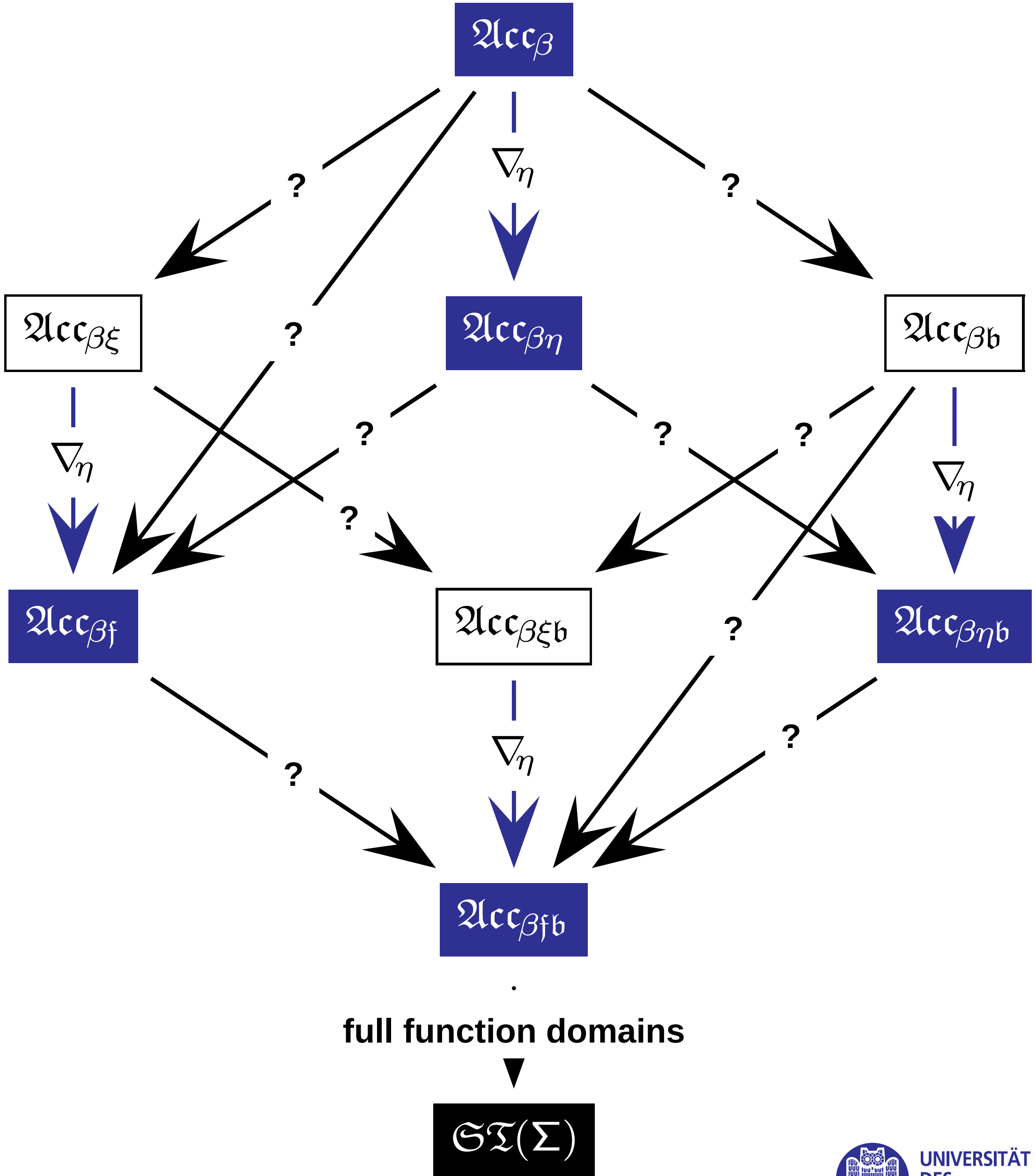
Properties for $\mathfrak{Acc}_{\beta}$

∇_c	…	$\nabla_\vee$	…
$\nabla_\neg$	…	$\nabla_\wedge$	…
∇_β	…	$\nabla_\forall$	…
		$\nabla_\exists$	…

Properties for Extensionality

$\nabla_{\mathfrak{b}}$ If $\neg(\mathbf{A} \doteq^{o} \mathbf{B}) \in \Phi$, then $\Phi,\mathbf{A},\neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi,\neg\mathbf{A},\mathbf{B} \in \Gamma_\Sigma$.

Abstract Consistency



Properties for $\mathfrak{Acc}_\beta$

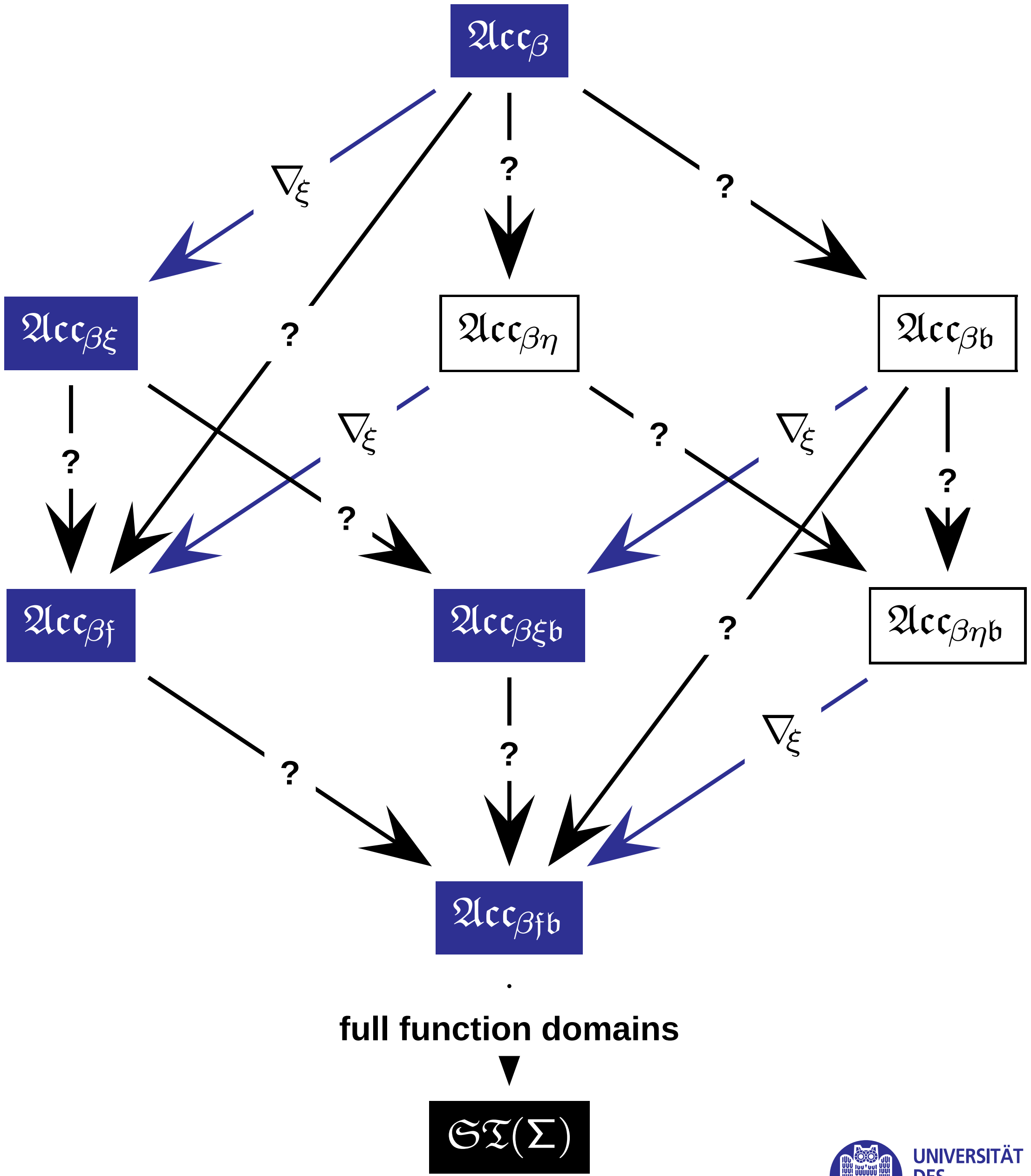
∇_c	…	$\nabla_\vee$	…
$\nabla_\neg$	…	$\nabla_\wedge$	…
∇_β	…	$\nabla_\forall$	…
		$\nabla_\exists$	…

Properties for Extensionality

$\nabla_\mathfrak{b}$ If $\neg(\mathbf{A} \doteq^{o} \mathbf{B}) \in \Phi$, then $\Phi, \mathbf{A}, \neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi, \neg\mathbf{A}, \mathbf{B} \in \Gamma_\Sigma$.

∇_η If $\mathbf{A} \stackrel{\beta\eta}{=} \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi, \mathbf{B} \in \Gamma_\Sigma$.

Abstract Consistency



Properties for $\mathfrak{Acc}_\beta$

∇_c	…	$\nabla_\vee$	…
$\nabla_\neg$	…	$\nabla_\wedge$	…
∇_β	…	$\nabla_\forall$	…
		$\nabla_\exists$	…

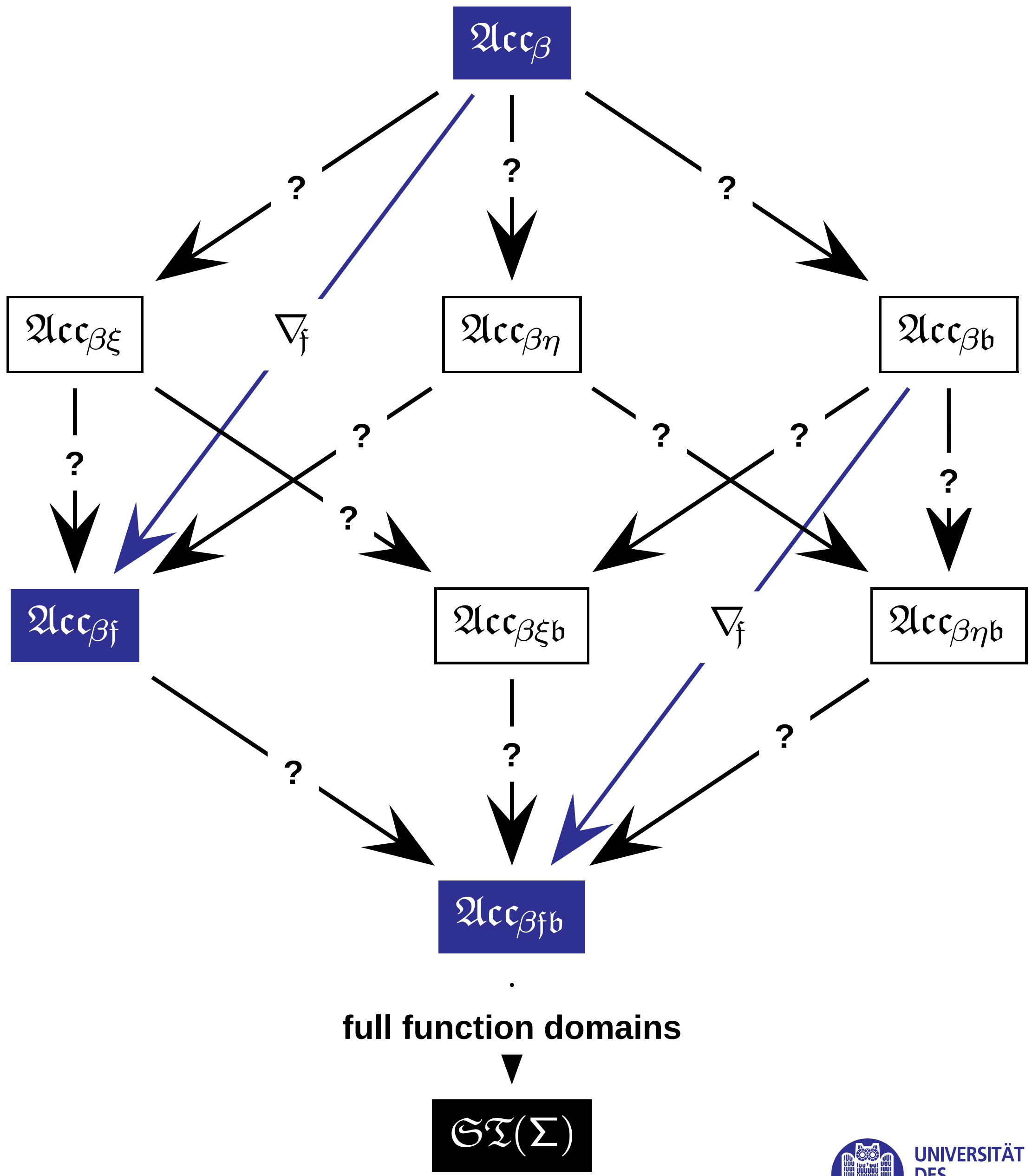
Properties for Extensionality

$\nabla_\mathfrak{b}$ If $\neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Phi$, then $\Phi, \mathbf{A}, \neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi, \neg\mathbf{A}, \mathbf{B} \in \Gamma_\Sigma$.

∇_η If $\mathbf{A} \overset{\beta\eta}{\equiv} \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi, \mathbf{B} \in \Gamma_\Sigma$.

∇_ξ If $\neg(\lambda X_\alpha.\mathbf{M} \doteq^{\alpha\to\beta} \lambda X_\alpha.\mathbf{N}) \in \Phi$, then $\Phi, \neg([w/X]\mathbf{M} \doteq^\beta [w/X]\mathbf{N}) \in \Gamma_\Sigma$ for any new $w_\alpha \in \Sigma_\alpha$.

Abstract Consistency



Properties for $\mathfrak{Acc}_\beta$

∇_c	...	$\nabla_\vee$	...
$\nabla_\neg$	...	$\nabla_\wedge$	...
∇_β	...	$\nabla_\forall$	...
		$\nabla_\exists$	...

Properties for Extensionality

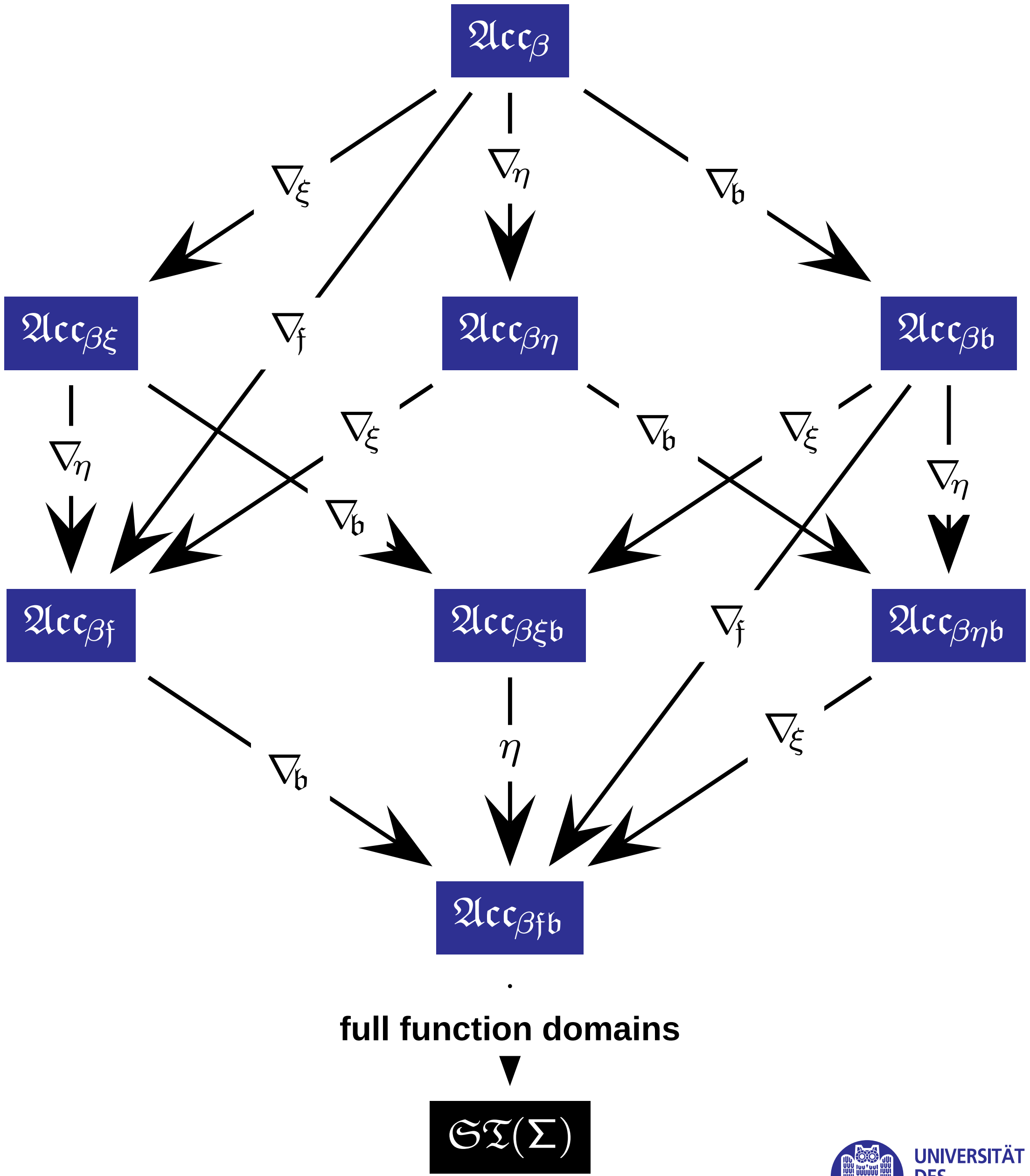
$\nabla_\mathfrak{b}$ If $\neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Phi$, then $\Phi, \mathbf{A}, \neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi, \neg\mathbf{A}, \mathbf{B} \in \Gamma_\Sigma$.

∇_η If $\mathbf{A} \stackrel{\beta\eta}{=} \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi, \mathbf{B} \in \Gamma_\Sigma$.

∇_ξ If $\neg(\lambda X_\alpha.\mathbf{M} \doteq^{\alpha\to\beta} \lambda X_\alpha.\mathbf{N}) \in \Phi$, then $\Phi, \neg([w/X]\mathbf{M} \doteq^\beta [w/X]\mathbf{N}) \in \Gamma_\Sigma$ for any new $w_\alpha \in \Sigma_\alpha$.

$\nabla_\mathfrak{f}$ If $\neg(\mathbf{G} \doteq^{\alpha\to\beta} \mathbf{H}) \in \Phi$, then $\Phi, \neg(\mathbf{G}w \doteq^\beta \mathbf{H}w) \in \Gamma_\Sigma$ for any new $w_\alpha \in \Sigma_\alpha$.

Abstract Consistency



Properties for $\mathfrak{Acc}_\beta$

∇_c	...	$\nabla_\vee$	...
$\nabla_\neg$	...	$\nabla_\wedge$	...
∇_β	...	$\nabla_\forall$	...
		$\nabla_\exists$	...

Properties for Extensionality

$\nabla_\mathfrak{b}$ If $\neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Phi$, then $\Phi, \mathbf{A}, \neg\mathbf{B} \in \Gamma_\Sigma$ or $\Phi, \neg\mathbf{A}, \mathbf{B} \in \Gamma_\Sigma$.

∇_η If $\mathbf{A} \overset{\beta\eta}{\equiv} \mathbf{B}$ and $\mathbf{A} \in \Phi$, then $\Phi, \mathbf{B} \in \Gamma_\Sigma$.

∇_ξ If $\neg(\lambda X_\alpha.\mathbf{M} \doteq^{\alpha\to\beta} \lambda X_\alpha.\mathbf{N}) \in \Phi$, then $\Phi, \neg([w/X]\mathbf{M} \doteq^\beta [w/X]\mathbf{N}) \in \Gamma_\Sigma$ for any new $w_\alpha \in \Sigma_\alpha$.

$\nabla_\mathfrak{f}$ If $\neg(\mathbf{G} \doteq^{\alpha\to\beta} \mathbf{H}) \in \Phi$, then $\Phi, \neg(\mathbf{G}w \doteq^\beta \mathbf{H}w) \in \Gamma_\Sigma$ for any new $w_\alpha \in \Sigma_\alpha$.

Model Existence Theorem

Thm.: (Model Existence Theorem(s) for $\mathfrak{Acc}_*$)
If a class of sets of formulas Γ_Σ
is a saturated abstract consistency class in $\mathfrak{Acc}_*$
then each $\Phi \in \Gamma_\Sigma$ is satisfied by a model $\mathcal{M}$ in $\mathfrak{M}_*$.

Model Existence Theorem

Thm.: (Model Existence Theorem(s) for $\mathfrak{Acc}_*$)
If a class of sets of formulas Γ_Σ
is a saturated abstract consistency class in $\mathfrak{Acc}_*$
then each $\Phi \in \Gamma_\Sigma$ is satisfied by a model $\mathcal{M}$ in $\mathfrak{M}_*$.

Appl.: (Completeness proofs by pure syntactical means)
Sufficient syntactical criteria for completeness of a calculus G
wrt. model class $\mathfrak{M}_*$:

Model Existence Theorem

Thm.: (Model Existence Theorem(s) for $\mathfrak{Acc}_*$)
If a class of sets of formulas Γ_Σ
is a saturated abstract consistency class in $\mathfrak{Acc}_*$
then each $\Phi \in \Gamma_\Sigma$ is satisfied by a model $\mathcal{M}$ in $\mathfrak{M}_*$.

Appl.: (Completeness proofs by pure syntactical means)
Sufficient syntactical criteria for completeness of a calculus G
wrt. model class $\mathfrak{M}_*$:

$$\Gamma_\Sigma^{\mathsf{G}} := \{\Phi | \Phi \text{ is } \mathsf{G}\text{-consistent}\} \text{ is a saturated } \mathfrak{Acc}_*$$

Model Existence Theorem

Thm.: (Model Existence Theorem(s) for $\mathfrak{Acc}_*$)
If a class of sets of formulas Γ_Σ
is a saturated abstract consistency class in $\mathfrak{Acc}_*$
then each $\Phi \in \Gamma_\Sigma$ is satisfied by a model $\mathcal{M}$ in $\mathfrak{M}_*$.

Appl.: (Completeness proofs by pure syntactical means)
Sufficient syntactical criteria for completeness of a calculus G
wrt. model class $\mathfrak{M}_*$:

$$\Gamma_\Sigma^G := \{\Phi | \Phi \text{ is G-consistent}\} \text{ is a saturated } \mathfrak{Acc}_*$$

Def.: (Saturated)
Γ_Σ is saturated if for all $\Phi \in \Gamma_\Sigma$ and formulas $\mathbf{A}$:

Model Existence Theorem

Thm.: (Model Existence Theorem(s) for $\mathfrak{Acc}_*$)
If a class of sets of formulas Γ_Σ
is a saturated abstract consistency class in $\mathfrak{Acc}_*$
then each $\Phi \in \Gamma_\Sigma$ is satisfied by a model $\mathcal{M}$ in $\mathfrak{M}_*$.

Appl.: (Completeness proofs by pure syntactical means)
Sufficient syntactical criteria for completeness of a calculus G
wrt. model class $\mathfrak{M}_*$:

$$\Gamma_\Sigma^{G} := \{\Phi | \Phi \text{ is G-consistent}\} \text{ is a saturated } \mathfrak{Acc}_*$$

Def.: (Saturated)
Γ_Σ is saturated if for all $\Phi \in \Gamma_\Sigma$ and formulas $\mathbf{A}$:

∇_{sat} Either $\Phi, \mathbf{A} \in \Gamma_\Sigma$ or $\Phi, \neg\mathbf{A} \in \Gamma_\Sigma$

The Sequent Calculus $\mathcal{G}_\beta$

Thm.: (Soundness of $\mathcal{G}_\beta$ for $\mathfrak{M}_*$)
$\mathcal{G}_\beta$ is sound for the eight model classes $\mathfrak{M}_*$.

The Sequent Calculus $\mathcal{G}_\beta$

Thm.: (Soundness of $\mathcal{G}_\beta$ for $\mathfrak{M}_*$)
$\mathcal{G}_\beta$ is sound for the eight model classes $\mathfrak{M}_*$.

Thm.: (Completeness of $\mathcal{G}_\beta$ for $\mathfrak{M}_\beta(\Sigma)$)
The calculus $\mathcal{G}_\beta$ is complete for the model class $\mathfrak{M}_\beta(\Sigma)$.

The Sequent Calculus $\mathcal{G}_\beta$

Thm.: (Soundness of $\mathcal{G}_\beta$ for $\mathfrak{M}_*$)
$\mathcal{G}_\beta$ is sound for the eight model classes $\mathfrak{M}_*$.

Thm.: (Completeness of $\mathcal{G}_\beta$ for $\mathfrak{M}_\beta(\Sigma)$)
The calculus $\mathcal{G}_\beta$ is complete for the model class $\mathfrak{M}_\beta(\Sigma)$.

Proof: Show $\Gamma_\Sigma^{\mathcal{G}_\beta} := \{\Phi | \Phi$ finite and $\mathcal{G}_\beta$-consistent (i.e. $\nVdash_{\mathcal{G}_\beta} \neg\Phi\big\downarrow_\beta$)$\}$
is a saturated abstract consistency class in $\mathfrak{Acc}_\beta$.

The Sequent Calculus $\mathcal{G}_\beta$

Thm.: (Soundness of $\mathcal{G}_\beta$ for $\mathfrak{M}_*$)
$\mathcal{G}_\beta$ is sound for the eight model classes $\mathfrak{M}_*$.

Thm.: (Completeness of $\mathcal{G}_\beta$ for $\mathfrak{M}_\beta(\Sigma)$)
The calculus $\mathcal{G}_\beta$ is complete for the model class $\mathfrak{M}_\beta(\Sigma)$.

Proof: Show $\Gamma_\Sigma^{\mathcal{G}_\beta} := \{\Phi | \Phi$ finite and $\mathcal{G}_\beta$-consistent (i.e. $\not\Vdash_{\mathcal{G}_\beta} \neg\Phi\downarrow_\beta$)$\}$
is a saturated abstract consistency class in $\mathfrak{Acc}_\beta$.

Thm.: (Cut and Saturation)
$\Gamma_\Sigma^{\mathcal{G}_\beta}$ is saturated if and only if $\mathcal{G}(cut)$ is admissible in $\mathcal{G}_\beta$.

The Sequent Calculus $\mathcal{G}_\beta$

Thm.: (Soundness of $\mathcal{G}_\beta$ for $\mathfrak{M}_*$)
$\mathcal{G}_\beta$ is sound for the eight model classes $\mathfrak{M}_*$.

Thm.: (Completeness of $\mathcal{G}_\beta$ for $\mathfrak{M}_\beta(\Sigma)$)
The calculus $\mathcal{G}_\beta$ is complete for the model class $\mathfrak{M}_\beta(\Sigma)$.

Proof: Show $\Gamma_\Sigma^{\mathcal{G}_\beta} := \{\Phi | \Phi$ finite and $\mathcal{G}_\beta$-consistent (i.e. $\not\Vdash_{\mathcal{G}_\beta} \neg\Phi{\downarrow_\beta}$)$\}$
is a saturated abstract consistency class in $\mathfrak{Acc}_\beta$.

Thm.: (Cut and Saturation)
$\Gamma_\Sigma^{\mathcal{G}_\beta}$ is saturated if and only if $\mathcal{G}(cut)$ is admissible in $\mathcal{G}_\beta$.

Cor.: (Cut-freeness of $\mathcal{G}_\beta$)
The cut rule $\mathcal{G}(cut)$ is admissible in $\mathcal{G}_\beta$.

k-Admissibility

Def.: We say a sequent calculus rule

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\, r \text{ is admissible in } \mathcal{G}$$

if $\Vdash_{\mathcal{G}} \Delta$ holds whenever $\Vdash_{\mathcal{G}} \Delta_i \qquad (\forall\, 1 \leq i \leq n)$.

k-Admissibility

Def.: We say a sequent calculus rule

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\, r \text{ is admissible in } \mathcal{G}$$

if $\Vdash_{\mathcal{G}} \Delta$ holds whenever $\Vdash_{\mathcal{G}} \Delta_i$ $\quad (\forall\, 1 \leq i \leq n)$.

Def.: For any $k \geq 0$, we call an admissible rule r

k-admissible

if elimination of any instance of r causes $\leq k$ additional steps.

k-Admissibility

Def.: We say a sequent calculus rule

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\, r \text{ is admissible in } \mathcal{G}$$

if $\Vdash_{\mathcal{G}} \Delta$ holds whenever $\Vdash_{\mathcal{G}} \Delta_i$ $(\forall\, 1 \leq i \leq n)$.

Def.: For any $k \geq 0$, we call an admissible rule r

k-admissible

if elimination of any instance of r causes $\leq k$ additional steps.

Idea: k-admissible (or k-derivable) rules are effectively simulated by the calculus

Is $\mathcal{G}(cut)$ k-Admissible in $\mathcal{G}_\beta$?

Claim: $\mathcal{G}(cut)$ is not k-admissible (i.e. not effectively simulated) in $\mathcal{G}_\beta$ for any $\mathsf{k} \in \mathbb{N}$.

Is $\mathcal{G}(cut)$ k-Admissible in $\mathcal{G}_\beta$?

Claim: $\mathcal{G}(cut)$ is not k-admissible (i.e. not effectively simulated) in $\mathcal{G}_\beta$ for any $k \in \mathbb{N}$.

Why?

- $\mathcal{G}_\beta$ cannot use higher-order to prove a first-order sequent Δ
- hyper-exponential speed-up results for first-order logic

Is $\mathcal{G}(cut)$ k-Admissible in $\mathcal{G}_\beta$?

Claim: $\mathcal{G}(cut)$ is not k-admissible (i.e. not effectively simulated) in $\mathcal{G}_\beta$ for any $k \in \mathbb{N}$.

Why?

- $\mathcal{G}_\beta$ cannot use higher-order to prove a first-order sequent Δ
- hyper-exponential speed-up results for first-order logic

Oops!: $\mathcal{G}(cut)$ becomes k-admissible (i.e. can be effectively simulated) in $\mathcal{G}_\beta$ if certain formulas are available the sequent Δ we wish to prove.

Is $\mathcal{G}(cut)$ k-Admissible in $\mathcal{G}_\beta$?

Claim: $\mathcal{G}(cut)$ is not k-admissible (i.e. not effectively simulated) in $\mathcal{G}_\beta$ for any $\mathsf{k} \in \mathbb{N}$.

Why?

- $\mathcal{G}_\beta$ cannot use higher-order to prove a first-order sequent Δ
- hyper-exponential speed-up results for first-order logic

Oops!: $\mathcal{G}(cut)$ becomes k-admissible (i.e. can be effectively simulated) in $\mathcal{G}_\beta$ if certain formulas are available the sequent Δ we wish to prove.

Hence: Certain formulas allow for effective cut-simulation.

Thread to the suitability of a calculus for proof automation!

Cut-Simulation

Leibniz-equations support Cut-Simulation

Ex.: Available Leibniz-equations $\mathbf{M} \doteq^{\alpha} \mathbf{N}$ $(:= \forall \mathrm{P}_{o\alpha}. \neg \mathrm{P}\mathbf{M} \vee \mathrm{P}\mathbf{N}))$
support cut-simulation in $\mathcal{G}_{\beta}$ in only 3 steps.
($\mathcal{G}(cut)$ is 3-derivable, hence 3-admissible, in $\mathcal{G}_{\beta}$)

Leibniz-equations support Cut-Simulation

Ex.: Available Leibniz-equations $\mathbf{M} \doteq^{\alpha} \mathbf{N}$ $(:= \forall \mathrm{P}_{o\alpha} \centerdot \neg \mathrm{P}\mathbf{M} \vee \mathrm{P}\mathbf{N}))$
support cut-simulation in $\mathcal{G}_{\beta}$ in only 3 steps.
($\mathcal{G}(cut)$ is 3-derivable, hence 3-admissible, in $\mathcal{G}_{\beta}$)

Proof:

$$\dfrac{\dfrac{\dfrac{\Delta, \mathbf{C}}{\Delta, \neg\neg\mathbf{C}}\ \mathcal{G}(\neg) \qquad \Delta, \neg\mathbf{C}}{\Delta, \neg(\neg\mathbf{C} \vee \mathbf{C})}\ \mathcal{G}(\vee_{-})}{\Delta' := \Delta, \neg\forall \mathrm{P}_{o\alpha} \centerdot \neg \mathrm{P}\mathbf{M} \vee \mathrm{P}\mathbf{N}}\ \mathcal{G}(\Pi_{-}^{\lambda X_{\alpha} \centerdot \mathbf{C}})$$

Extensionality Axioms

- The extensionality axioms are usually added to HOL systems (e.g. [Huet73]) to reach Henkin completeness.

Extensionality Axioms

- The extensionality axioms are usually added to HOL systems (e.g. [Huet73]) to reach Henkin completeness.

Def.: The Boolean extensionality axiom $\mathcal{B}_o$ is:

$$\forall A_o \centerdot \forall B_o \centerdot (A \Leftrightarrow B) \Rightarrow A \doteq^o B$$

Extensionality Axioms

- The extensionality axioms are usually added to HOL systems (e.g. [Huet73]) to reach Henkin completeness.

Def.: The Boolean extensionality axiom $\mathcal{B}_o$ is:

$$\forall A_o. \forall B_o. (A \Leftrightarrow B) \Rightarrow A \doteq^o B$$

Def.: The infinitely many functional extensionality axioms $\mathcal{F}_{\alpha\beta}$ are:

$$\forall F_{\alpha\to\beta}. \forall G_{\alpha\to\beta}. (\forall X_\alpha. FX \doteq^\beta GX) \Rightarrow F \doteq^{\alpha\to\beta} G$$

Extensionality Axioms

- The extensionality axioms are usually added to HOL systems (e.g. [Huet73]) to reach Henkin completeness.

Def.: The Boolean extensionality axiom $\mathcal{B}_o$ is:

$$\forall A_o. \forall B_o. (A \Leftrightarrow B) \Rightarrow A \doteq^o B$$

Def.: The infinitely many functional extensionality axioms $\mathcal{F}_{\alpha\beta}$ are:

$$\forall F_{\alpha\to\beta}. \forall G_{\alpha\to\beta}. (\forall X_\alpha. FX \doteq^\beta GX) \Rightarrow F \doteq^{\alpha\to\beta} G$$

- Is adding these axioms a suitable option for proof automation?

Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$

Def.: (Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$)

We define $\mathcal{G}_\beta^{\mathsf{E}} := \mathcal{G}_\beta \cup \{\mathcal{G}(\mathcal{F}_{\alpha\beta}), \mathcal{G}(\mathcal{B})\}$

Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$

Def.: (Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$)

We define $\mathcal{G}_\beta^{\mathsf{E}} := \mathcal{G}_\beta \cup \{\mathcal{G}(\mathcal{F}_{\alpha\beta}), \mathcal{G}(\mathcal{B})\}$

$$\frac{\Delta, \neg\mathcal{F}_{\alpha\beta} \quad \alpha\beta \in \mathcal{T}}{\Delta}\ \mathcal{G}(\mathcal{F}_{\alpha\beta}) \qquad \frac{\Delta, \neg\mathcal{B}_o}{\Delta}\ \mathcal{G}(\mathcal{B})$$

Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$

Def.: (Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$)

We define $\mathcal{G}_\beta^{\mathsf{E}} := \mathcal{G}_\beta \cup \{\mathcal{G}(\mathcal{F}_{\alpha\beta}), \mathcal{G}(\mathcal{B})\}$

$$\frac{\Delta, \neg\mathcal{F}_{\alpha\beta} \quad \alpha\beta \in \mathcal{T}}{\Delta}\,\mathcal{G}(\mathcal{F}_{\alpha\beta}) \qquad\qquad \frac{\Delta, \neg\mathcal{B}_{\circ}}{\Delta}\,\mathcal{G}(\mathcal{B})$$

Thm.: (Soundness and Completeness of $\mathcal{G}_\beta^{\mathsf{E}}$)

$\mathcal{G}_\beta^{\mathsf{E}}$ is sound and complete for Henkin semantics.

Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$

Def.: (Sequent Calculus $\mathcal{G}_\beta^{\mathsf{E}}$)

We define $\mathcal{G}_\beta^{\mathsf{E}} := \mathcal{G}_\beta \cup \{\mathcal{G}(\mathcal{F}_{\alpha\beta}), \mathcal{G}(\mathcal{B})\}$

$$\frac{\Delta, \neg\mathcal{F}_{\alpha\beta} \quad \alpha\beta \in \mathcal{T}}{\Delta}\ \mathcal{G}(\mathcal{F}_{\alpha\beta}) \qquad\qquad \frac{\Delta, \neg\mathcal{B}_o}{\Delta}\ \mathcal{G}(\mathcal{B})$$

Thm.: (Soundness and Completeness of $\mathcal{G}_\beta^{\mathsf{E}}$)

$\mathcal{G}_\beta^{\mathsf{E}}$ is sound and complete for Henkin semantics.

But: $\mathcal{G}_\beta^{\mathsf{E}}$ supports effective cut-simulation.

Ext. Axioms support Cut-Simulation

Ex.: The functional extensionality axioms support effective cut-simulation in $\mathcal{G}_\beta^{\mathsf{E}}$ in 12-steps.
($\mathcal{G}(cut)$ is 12-derivable, hence 12-admissible, in $\mathcal{G}_\beta^{\mathsf{E}}$)

Ext. Axioms support Cut-Simulation

Ex.: The functional extensionality axioms support effective cut-simulation in $\mathcal{G}_\beta^{\mathsf{E}}$ in 12-steps.
($\mathcal{G}(cut)$ is 12-derivable, hence 12-admissible, in $\mathcal{G}_\beta^{\mathsf{E}}$)

Proof:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\begin{array}{c}\text{3 steps; easy}\\ \vdots \\ \Delta, \mathsf{fa} \doteq^{\beta} \mathsf{fa}\end{array}}{\Delta, (\forall \mathsf{X}_\alpha \centerdot \mathsf{fX} \doteq^{\beta} \mathsf{fX})}\ \mathcal{G}(\Pi_+^{a_\alpha})
}{\Delta, \neg\neg\forall \mathsf{X}_\alpha \centerdot \mathsf{fX} \doteq^{\beta} \mathsf{fX}}\ \mathcal{G}(\neg)
\qquad
\begin{array}{c}\Delta, \mathbf{C} \quad \Delta, \neg\mathbf{C}\\ \vdots\ \text{3 steps; see before}\\ \Delta, \neg(\mathsf{f} \doteq^{\alpha\to\beta} \mathsf{f})\end{array}
}{\Delta, \neg(\neg(\forall \mathsf{X}_\alpha \centerdot \mathsf{fX} \doteq^{\beta} \mathsf{fX}) \vee \mathsf{f} \doteq^{\alpha\to\beta} \mathsf{f})}\ \mathcal{G}(\vee_-)
}{\Delta, \neg\mathcal{F}_{\alpha\beta}}\ 2 \times \mathcal{G}(\Pi_-^{f})
}{\Delta}\ \mathcal{G}(\mathcal{F}_{\alpha\beta})
$$

Ext. Axioms support Cut-Simulation

Ex.: It also works with Boolean extensionality axiom – in 15 steps.

Ext. Axioms support Cut-Simulation

Ex.: It also works with Boolean extensionality axiom – in 15 steps.

Proof:

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \begin{array}{c}\text{7 steps; easy}\\ \vdots\\ \Delta, a \Leftrightarrow a\end{array}
      }{
        \Delta, \neg\neg(a \Leftrightarrow a)
      }\;\mathcal{G}(\neg)
      \qquad
      \begin{array}{c}\Delta, \mathbf{C} \quad \Delta, \neg\mathbf{C}\\ \vdots \;\text{3 steps; see before}\\ \Delta, \neg(a \doteq^{o} a)\end{array}
    }{
      \Delta, \neg(\neg(a \Leftrightarrow a) \lor a \doteq^{o} a)
    }\;\mathcal{G}(\lor_{-})
  }{
    \Delta, \neg\mathcal{B}_{o}
  }\;2 \times \mathcal{G}(\Pi^{a}_{-})
}{
  \Delta
}\;\mathcal{G}(\mathcal{B})
$$

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) 4 steps

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) 4 steps
- Instances of Comprehension axioms 16 steps

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) — 4 steps
- Instances of Comprehension axioms — 16 steps
- Axiom of Induction — 18 steps

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) 4 steps
- Instances of Comprehension axioms 16 steps
- Axiom of Induction 18 steps
- Axiom of Choice 7 steps

Cut-Simulation with other Formulas

Reflexivity definition of equality (Andrews)	4 steps
Instances of Comprehension axioms	16 steps
Axiom of Induction	18 steps
Axiom of Choice	7 steps
Axiom of Description	25 steps

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) 4 steps
- Instances of Comprehension axioms 16 steps
- Axiom of Induction 18 steps
- Axiom of Choice 7 steps
- Axiom of Description 25 steps
- Axiom of Excluded Middle 3 steps

Cut-Simulation with other Formulas

- Reflexivity definition of equality (Andrews) 4 steps
- Instances of Comprehension axioms 16 steps
- Axiom of Induction 18 steps
- Axiom of Choice 7 steps
- Axiom of Description 25 steps
- Axiom of Excluded Middle 3 steps
- ???

How to Avoid Cut-Simulation?

Key: Avoid cut-strong axioms (here extensionality)

How to Avoid Cut-Simulation?

Key: Avoid cut-strong axioms (here extensionality)

Def.: We define calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}} := \mathcal{G}_{\beta} \cup \{\mathcal{G}(\mathfrak{f}), \mathcal{G}(\mathfrak{b}), \mathcal{G}(Init^{\dot{=}}), \mathcal{G}(d)\}$

$$\frac{\Delta, (\forall \mathsf{X}_\alpha . \mathbf{F}\mathsf{X} \doteq^{\beta} \mathbf{G}\mathsf{X}){\downarrow_\beta}}{\Delta, (\mathbf{F} \doteq^{\alpha\to\beta} \mathbf{G})}\ \mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta, \neg\mathbf{A}, \mathbf{B} \quad \Delta, \neg\mathbf{B}, \mathbf{A}}{\Delta, (\mathbf{A} \doteq^{o} \mathbf{B})}\ \mathcal{G}(\mathfrak{b})$$

$$\frac{\Delta, \mathbf{A} \doteq^{o} \mathbf{B} \quad \dagger}{\Delta, \neg\mathbf{A}, \mathbf{B}}\ \mathcal{G}(Init^{\dot{=}}) \qquad \frac{\Delta, \mathbf{A}^1 \doteq^{\alpha_1} \mathbf{B}^1 \ \cdots \ \Delta, \mathbf{A}^n \doteq^{\alpha_n} \mathbf{B}^n \quad \ddagger}{\Delta, \mathsf{h}\overline{\mathbf{A}^n} \doteq^{\beta} \mathsf{h}\overline{\mathbf{B}^n}}\ \mathcal{G}(d)$$

$\dagger$ $\mathbf{A}, \mathbf{B}$ atomic $\qquad$ $\ddagger$ $\mathsf{n} \geq 1, \beta \in \{o, \iota\}, \mathsf{h}_{\overline{\alpha^n}\to\beta} \in \Sigma$ parameter

How to Avoid Cut-Simulation?

Key: Avoid cut-strong axioms (here extensionality)

Def.: We define calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}} := \mathcal{G}_{\beta} \cup \{\mathcal{G}(\mathfrak{f}), \mathcal{G}(\mathfrak{b}), \mathcal{G}(Init^{\dot{=}}), \mathcal{G}(d)\}$

$$\frac{\Delta, (\forall X_\alpha . \mathbf{F}X \doteq^{\beta} \mathbf{G}X){\downarrow_\beta}}{\Delta, (\mathbf{F} \doteq^{\alpha\to\beta} \mathbf{G})}\,\mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta, \neg\mathbf{A}, \mathbf{B} \quad \Delta, \neg\mathbf{B}, \mathbf{A}}{\Delta, (\mathbf{A} \doteq^{o} \mathbf{B})}\,\mathcal{G}(\mathfrak{b})$$

$$\frac{\Delta, \mathbf{A} \doteq^{o} \mathbf{B} \quad \dagger}{\Delta, \neg\mathbf{A}, \mathbf{B}}\,\mathcal{G}(Init^{\dot{=}}) \qquad \frac{\Delta, \mathbf{A}^1 \doteq^{\alpha_1} \mathbf{B}^1 \quad \cdots \quad \Delta, \mathbf{A}^n \doteq^{\alpha_n} \mathbf{B}^n \quad \ddagger}{\Delta, h\overline{\mathbf{A}^n} \doteq^{\beta} h\overline{\mathbf{B}^n}}\,\mathcal{G}(d)$$

$\dagger$ $\mathbf{A}, \mathbf{B}$ atomic $\qquad$ $\ddagger$ $n \geq 1, \beta \in \{o, \iota\}, h_{\overline{\alpha^n}\to\beta} \in \Sigma$ parameter

Thm.: The calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ is sound and complete for Henkin semantics.

How to Avoid Cut-Simulation?

Key: Avoid cut-strong axioms (here extensionality)

Def.: We define calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}} := \mathcal{G}_\beta \cup \{\mathcal{G}(\mathfrak{f}), \mathcal{G}(\mathfrak{b}), \mathcal{G}(Init^{\dot{=}}), \mathcal{G}(d)\}$

$$\frac{\Delta, (\forall X_\alpha . \mathbf{F}X \doteq^\beta \mathbf{G}X){\downarrow_\beta}}{\Delta, (\mathbf{F} \doteq^{\alpha\to\beta} \mathbf{G})}\ \mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta, \neg\mathbf{A}, \mathbf{B} \quad \Delta, \neg\mathbf{B}, \mathbf{A}}{\Delta, (\mathbf{A} \doteq^{o} \mathbf{B})}\ \mathcal{G}(\mathfrak{b})$$

$$\frac{\Delta, \mathbf{A} \doteq^{o} \mathbf{B} \quad \dagger}{\Delta, \neg\mathbf{A}, \mathbf{B}}\ \mathcal{G}(Init^{\dot{=}}) \qquad \frac{\Delta, \mathbf{A}^1 \doteq^{\alpha_1} \mathbf{B}^1 \quad \cdots \quad \Delta, \mathbf{A}^n \doteq^{\alpha_n} \mathbf{B}^n \quad \ddagger}{\Delta, h\overline{\mathbf{A}^n} \doteq^{\beta} h\overline{\mathbf{B}^n}}\ \mathcal{G}(d)$$

$\dagger$ $\mathbf{A}, \mathbf{B}$ atomic $\qquad$ $\ddagger$ $n \geq 1, \beta \in \{o, \iota\}, h_{\overline{\alpha^n}\to\beta} \in \Sigma$ parameter

Thm.: The calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ is sound and complete for Henkin semantics.

Claim: $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ does not support effective cut-simulation

Abstract Cut-Elimination Result

The rules $\mathcal{G}(Init^{\dot{=}}), \mathcal{G}(d)$ motivate corresponding

abstract consistency conditions for Henkin semantics

∇_m If $\mathbf{A}, \mathbf{B} \in cwff_o(\Sigma)$ are atomic and $\mathbf{A}, \neg\mathbf{B} \in \Phi$, then $\Phi * \neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Gamma_\Sigma$.

∇_d If $\neg(h\overline{\mathbf{A}^n} \doteq^\beta h\overline{\mathbf{B}^n}) \in \Phi$ for some types α_i where $\beta \in \{o, \iota\}$ and $h_{\overline{\alpha^n} \to \beta} \in \Sigma$ is a parameter, then there is an i ($1 \leq i \leq n$) such that $\Phi * \neg(\mathbf{A}^i \doteq^{\alpha^i} \mathbf{B}^i) \in \Gamma_\Sigma$.

which sufficiently replace the strong saturation condition ∇_{sat}

Abstract Cut-Elimination Result

The rules $\mathcal{G}(Init^{\dot=}), \mathcal{G}(d)$ motivate corresponding

abstract consistency conditions for Henkin semantics

∇_m If $\mathbf{A}, \mathbf{B} \in cwff_o(\Sigma)$ are atomic and $\mathbf{A}, \neg\mathbf{B} \in \Phi$, then $\Phi * \neg(\mathbf{A} \doteq^o \mathbf{B}) \in \Gamma_\Sigma$.

∇_d If $\neg(h\overline{\mathbf{A}^n} \doteq^\beta h\overline{\mathbf{B}^n}) \in \Phi$ for some types α_i where $\beta \in \{o, \iota\}$ and $h_{\overline{\alpha^n} \rightarrow \beta} \in \Sigma$ is a parameter, then there is an i $(1 \leq i \leq n)$ such that $\Phi * \neg(\mathbf{A}^i \doteq^{\alpha^i} \mathbf{B}^i) \in \Gamma_\Sigma$.

which sufficiently replace the strong saturation condition ∇_{sat}

- abstract cut-elimination result for Henkin-semantics

Conclusion

- Issues for the automation of IL:
  - avoid naive treatment of cut-strong axioms and formulas
  - only some first steps achieved: comprehension, extensionality, primitive equality

Conclusion

- Issues for the automation of IL:
  - avoid naive treatment of cut-strong axioms and formulas
  - only some first steps achieved: comprehension, extensionality, primitive equality
- Debatable:
  - How useful is ’cut-freeness’ criterion in IL? (without also considering cut-simulation)

Conclusion

- Issues for the automation of IL:
  - avoid naive treatment of cut-strong axioms and formulas
  - only some first steps achieved: comprehension, extensionality, primitive equality
- Debatable:
  - How useful is 'cut-freeness' criterion in IL? (without also considering cut-simulation)
- Further work:
  - . . . research is only at its very beginning . . .