A Structured Set of Higher-Order Problems

Christoph E. Benzmüller and Chad E. Brown

FR Informatik

Universität des Saarlandes

Saarbrücken, Germany

Dagstuhl Seminar 05431, Schloss Dagstuhl, Germany

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems
... for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi
- Integration: interactive TP, proof planning, agent-oriented TP

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi
- Integration: interactive TP, proof planning, agent-oriented TP
- Verifiable proof objects

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi
- Integration: interactive TP, proof planning, agent-oriented TP
- Verifiable proof objects
- Internal & external specialist (sub)systems: e.g. FO/HO-ATPs

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi
- Integration: interactive TP, proof planning, agent-oriented TP
- Verifiable proof objects
- Internal & external specialist (sub)systems: e.g. FO/HO-ATPs
- Foundational framework(s) for mathematics

Research Topics of the ΩMEGA Group

Integrated mathematics assistance systems

. . . for formal methods, mathematics, and e-learning

- Mediation between “mathural” and "maschine mathematics”
- Publication oriented wysiwyg user interface (TeXmacs)
- Human oriented and cognitively adequate proof calculi
- Integration: interactive TP, proof planning, agent-oriented TP
- Verifiable proof objects
- Internal & external specialist (sub)systems: e.g. FO/HO-ATPs
- Foundational framework(s) for mathematics

Talk: Mechanization of Church’s simple type theory; test problems

Test Problems for Theorem Provers

- Test problems for FOL theorem provers

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs
- Test problems for HOL theorem provers

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs
- Test problems for HOL theorem provers
  - common library missing

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs
- Test problems for HOL theorem provers
  - common library missing
- This talk: example problems from our paper [TPHOLS-05]

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs
- Test problems for HOL theorem provers
  - common library missing
- This talk: example problems from our paper [TPHOLS-05]
- Are we proposing challenging HOL benchmark problems?

Test Problems for Theorem Provers

- Test problems for FOL theorem provers
  - [McCharenOverbeekWos76], [WilsonMinker79], [Pelletier86], etc.
  - TPTP [PelletierSutcliffeSuttner02]
  - significantly fostered the development of FOL ATPs
- Test problems for HOL theorem provers
  - common library missing
- This talk: example problems from our paper [TPHOLS-05]
- Are we proposing challenging HOL benchmark problems?
  - No!!!

Our Examples

- Examples are simple

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP
- Examples are structured

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP
- Examples are structured
  - quick indicators for completeness and soundness wrt to HOL model classes from [Benzm.BrownKohlhase-JSL-04]

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP
- Examples are structured
  - quick indicators for completeness and soundness wrt to HOL model classes from [Benzm.BrownKohlhase-JSL-04]
  - shall precede formal soundness / completeness analysis

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP
- Examples are structured
  - quick indicators for completeness and soundness wrt to HOL model classes from [Benzm.BrownKohlhase-JSL-04]
  - shall precede formal soundness / completeness analysis
  - many are collected from experience with LEO and TPS

Our Examples

- Examples are simple
  - highlight the essence of some semantical or technical point
  - easy to understand and easy to encode
  - relevant for both: automated and interactive TP
- Examples are structured
  - quick indicators for completeness and soundness wrt to HOL model classes from [Benzm.BrownKohlhase-JSL-04]
  - shall precede formal soundness / completeness analysis
  - many are collected from experience with LEO and TPS
- (Some more challenging examples are also added)

Outline of Talk

- HOL (notion and syntax)

Outline of Talk

- HOL (notion and syntax)
- HOL-CUBE: different model classes for HOL (semantics)

Outline of Talk

- HOL (notion and syntax)
- HOL-CUBE: different model classes for HOL (semantics)
  - Boolean extensionality
  - functional extensionality
    - η-equality
    - ξ-equality

Outline of Talk

- HOL (notion and syntax)
- HOL-CUBE: different model classes for HOL (semantics)
  - Boolean extensionality
  - functional extensionality
    - η-equality
    - ξ-equality
- HOL-Problems: simple problems structured along HOL-CUBE

Outline of Talk

- HOL (notion and syntax)
- HOL-CUBE: different model classes for HOL (semantics)
  - Boolean extensionality
  - functional extensionality
    - η-equality
    - ξ-equality
- HOL-Problems: simple problems structured along HOL-CUBE

Outline of Talk

- HOL (notion and syntax)
- HOL-CUBE: different model classes for HOL (semantics)
  - Boolean extensionality
  - functional extensionality
    - η-equality
    - ξ-equality
- HOL-Problems: simple problems structured along HOL-CUBE
- Conclusion

HOL: Simple Types

Simple Types $\mathcal{T}$:

$$
\begin{array}{ll}
o & \text{(truth values)} \\
\iota & \text{(individuals)} \\
(\alpha\beta) & \text{(functions from } \beta \text{ to } \alpha\text{)}
\end{array}
$$

HOL: Simple Types

Simple Types $\mathcal{T}$:	o	(truth values)
	ι	(individuals)
	$(\alpha\beta)$	(functions from β to α)

$(\alpha\beta\gamma)$ abbreviates $((\alpha\beta)\gamma)$

HOL: Simply Typed λ-Terms

Terms:

X_α	Variables ($\mathcal{V}$)
a_α	Parameters ($\mathcal{P}$)
c_α	Logical constants ($\mathcal{S}$)
$[F_{\alpha\beta}\, B_\beta]_\alpha$	Application
$[\lambda Y_\beta\, A_\alpha]_{\alpha\beta}$	λ-abstraction

HOL: Simply Typed λ-Terms

	X_α	Variables ($\mathcal{V}$)
	a_α	Parameters ($\mathcal{P}$)
Terms:	c_α	Logical constants ($\mathcal{S}$)
	$[F_{\alpha\beta}\, B_\beta]_\alpha$	Application
	$[\lambda Y_\beta\, A_\alpha]_{\alpha\beta}$	λ-abstraction

Equality of terms: $\alpha\beta\eta$

HOL: Simply Typed λ-Terms

Terms:	X_α	Variables ($\mathcal{V}$)
	a_α	Parameters ($\mathcal{P}$)
	c_α	Logical constants ($\mathcal{S}$)
	$[F_{\alpha\beta}\, B_\beta]_\alpha$	Application
	$[\lambda Y_\beta\, A_\alpha]_{\alpha\beta}$	λ-abstraction

Equality of terms: $\alpha\beta\eta$

α-conversion	Changing Bound Variables	
β-reduction	$[[\lambda Y_\beta\, A_\alpha]\, B] \xrightarrow{\beta} [B/Y]A$	
η-reduction	$[\lambda Y_\beta\, [F_{\alpha\beta}\, Y]] \xrightarrow{\eta} F$	$(Y_\beta \notin \mathbf{Free}(F))$

HOL: Simply Typed λ-Terms

Terms:

X_α	Variables ($\mathcal{V}$)
a_α	Parameters ($\mathcal{P}$)
c_α	Logical constants ($\mathcal{S}$)
$[F_{\alpha\beta}\, B_\beta]_\alpha$	Application
$[\lambda Y_\beta\, A_\alpha]_{\alpha\beta}$	λ-abstraction

Equality of terms: $\alpha\beta\eta$

Every term has a unique $\beta\eta$-normal form (up to α-conversion).

HOL: Logical Constants

Some logical constants which may be in $\mathcal{S}$:

- $\top_o$ – true
- $\bot_o$ – false
- $\neg_{oo}$ – negation
- $\vee_{ooo}$ – disjunction
- $\wedge_{ooo}$ – conjunction
- $\Rightarrow_{ooo}$ – implication
- $\Leftrightarrow_{ooo}$ – equivalence

HOL: Logical Constants

Some logical constants which may be in $\mathcal{S}$:

- $=^{\alpha}_{o\alpha\alpha}$ – equality at type α
- $\Pi^{\alpha}_{o(o\alpha)}$ – universal quantification over type α
- $\Sigma^{\alpha}_{o(o\alpha)}$ – existential quantification over type α

Intuition:

- $[\Sigma^{\alpha}\,[\lambda X_{\alpha}\, C_{o}]]$ is true iff $\{X_{\alpha} \mid C\}$ is nonempty.

HOL: Logical Constants

- $[A_o \lor B_o]$ means $[\lor_{ooo} A_o B_o]$
- $[A_o \land B_o]$ means $[\land_{ooo} A_o B_o]$
- $[A_o \Rightarrow B_o]$ means $[\Rightarrow_{ooo} A_o B_o]$
- $[A_o \Leftrightarrow B_o]$ means $[\Leftrightarrow_{ooo} A_o B_o]$
- $[A_\alpha =^\alpha B_\alpha]$ means $[=^\alpha_{o\alpha\alpha} A_\alpha B_\alpha]$
- $[\forall X_\alpha A_o]$ means $[\Pi^\alpha_{o(o\alpha)} \blacksquare \lambda X_\alpha A_o]$
- $[\exists X_\alpha A_o]$ means $[\Sigma^\alpha_{o(o\alpha)} \blacksquare \lambda X_\alpha A_o]$

HOL: Equality and Extensionality

Some important conventions and reminders for the talk:

- $=$ denotes primitive equality

HOL: Equality and Extensionality

Some important conventions and reminders for the talk:

- $=$ denotes primitive equality
- $\doteq$ denotes Leibniz equality: $A_\alpha \doteq^\alpha B_\alpha := \forall P_{o\alpha}.\ (PA) \Rightarrow (PB)$

HOL: Equality and Extensionality

Some important conventions and reminders for the talk:

- $=$ denotes primitive equality
- $\doteq$ denotes Leibniz equality: $A_\alpha \doteq^\alpha B_\alpha := \forall P_{o\alpha}. (PA) \Rightarrow (PB)$
- $\ddot{=}$ . . . other definition of equality (e.g., see [Andrews02])

HOL: Equality and Extensionality

Some important conventions and reminders for the talk:

- $=$ denotes primitive equality
- $\doteq$ denotes Leibniz equality: $A_\alpha \doteq^\alpha B_\alpha := \forall P_{o\alpha}.\, (P A) \Rightarrow (P B)$
- $\ddot{=}$ ... other definition of equality (e.g., see [Andrews02])

We use $\overset{*}{=}$ in the following to refer to any of the above

HOL: Church’s Type Theory

Church’s Type Theory:

- Simply typed λ-calculus with the signature

  $$\mathcal{S} := \{\neg, \vee\} \cup \{\Pi^\alpha \mid \alpha \in \mathcal{T}\}$$

  (and perhaps a description or choice operator).

HOL: Church's Type Theory

Church's Type Theory:

- Simply typed λ-calculus with the signature

  $$\mathcal{S} := \{\neg, \vee\} \cup \{\Pi^{\alpha} \mid \alpha \in \mathcal{T}\}$$

  (and perhaps a description or choice operator).
- Axioms of extensionality

HOL: Church's Type Theory

Church's Type Theory:

- Simply typed λ-calculus with the signature

  $\mathcal{S} := \{\neg, \vee\} \cup \{\Pi^{\alpha} \mid \alpha \in \mathcal{T}\}$

  (and perhaps a description or choice operator).
- Axioms of extensionality
- Axiom of description or choice

HOL: Church’s Type Theory

Church’s Type Theory:

- Simply typed λ-calculus with the signature

  $\mathcal{S} := \{\neg, \vee\} \cup \{\Pi^\alpha \mid \alpha \in \mathcal{T}\}$

  (and perhaps a description or choice operator).
- Axioms of extensionality
- Axiom of description or choice
- Axiom of infinity

HOL: Other Fragments

$\mathcal{S}$ Fragment of Elementary Type Theory:

$\mathcal{S}$ Fragment of Extensional Type Theory:

HOL: Other Fragments

$\mathcal{S}$ Fragment of Elementary Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$

$\mathcal{S}$ Fragment of Extensional Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$

HOL: Other Fragments

$\mathcal{S}$ Fragment of Elementary Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- No extensionality

$\mathcal{S}$ Fragment of Extensional Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- Extensionality

HOL: Other Fragments

$\mathcal{S}$ Fragment of Elementary Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- No extensionality
- No axiom of description or choice

$\mathcal{S}$ Fragment of Extensional Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- Extensionality
- No axiom of description or choice

HOL: Other Fragments

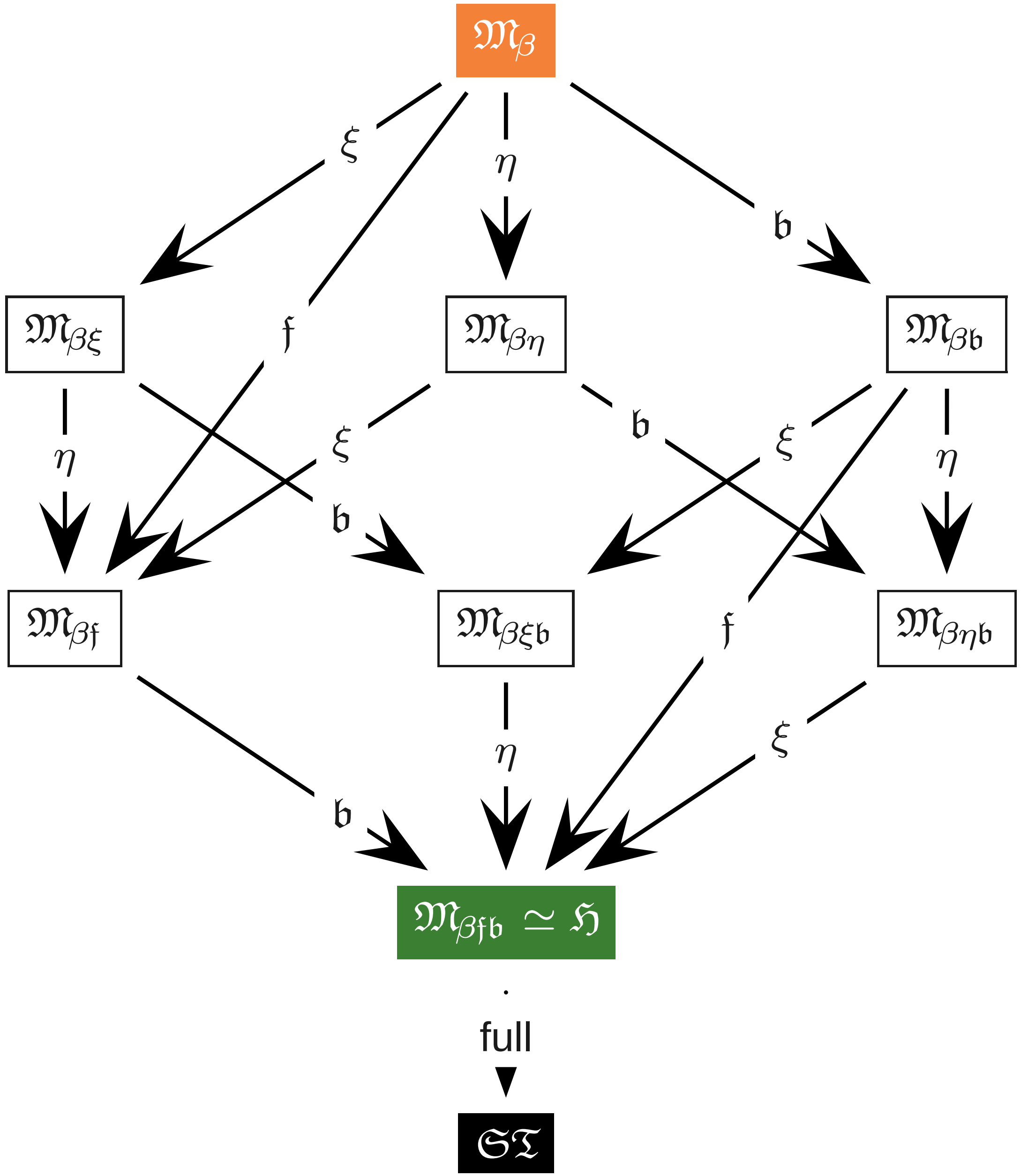
$\mathcal{S}$ Fragment of Elementary Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- No extensionality
- No axiom of description or choice
- No axiom of infinity

$\mathcal{S}$ Fragment of Extensional Type Theory:

- Simply typed λ-calculus with the signature $\mathcal{S}$
- Extensionality
- No axiom of description or choice
- No axiom of infinity

Semantics: HOL-CUBE



Landscape of HOL model classes

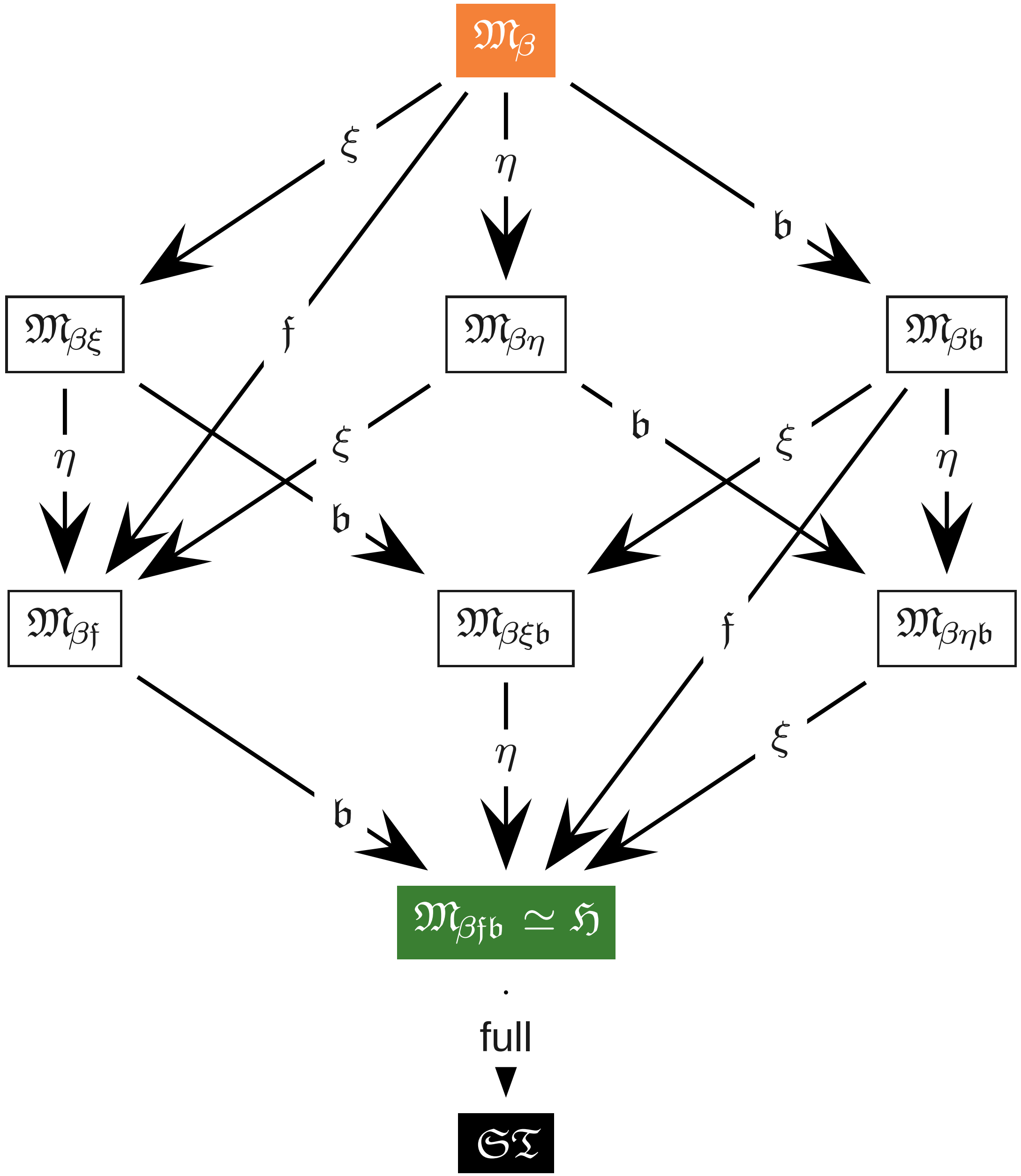
[Kohlhase-PhD-94]

[Benzmüller-PhD-99]

[Brown-PhD-04]

[Benzm.BrownKohlhase-JSL-04]

Semantics: HOL-CUBE



Landscape of HOL model classes

[Kohlhase-PhD-94]

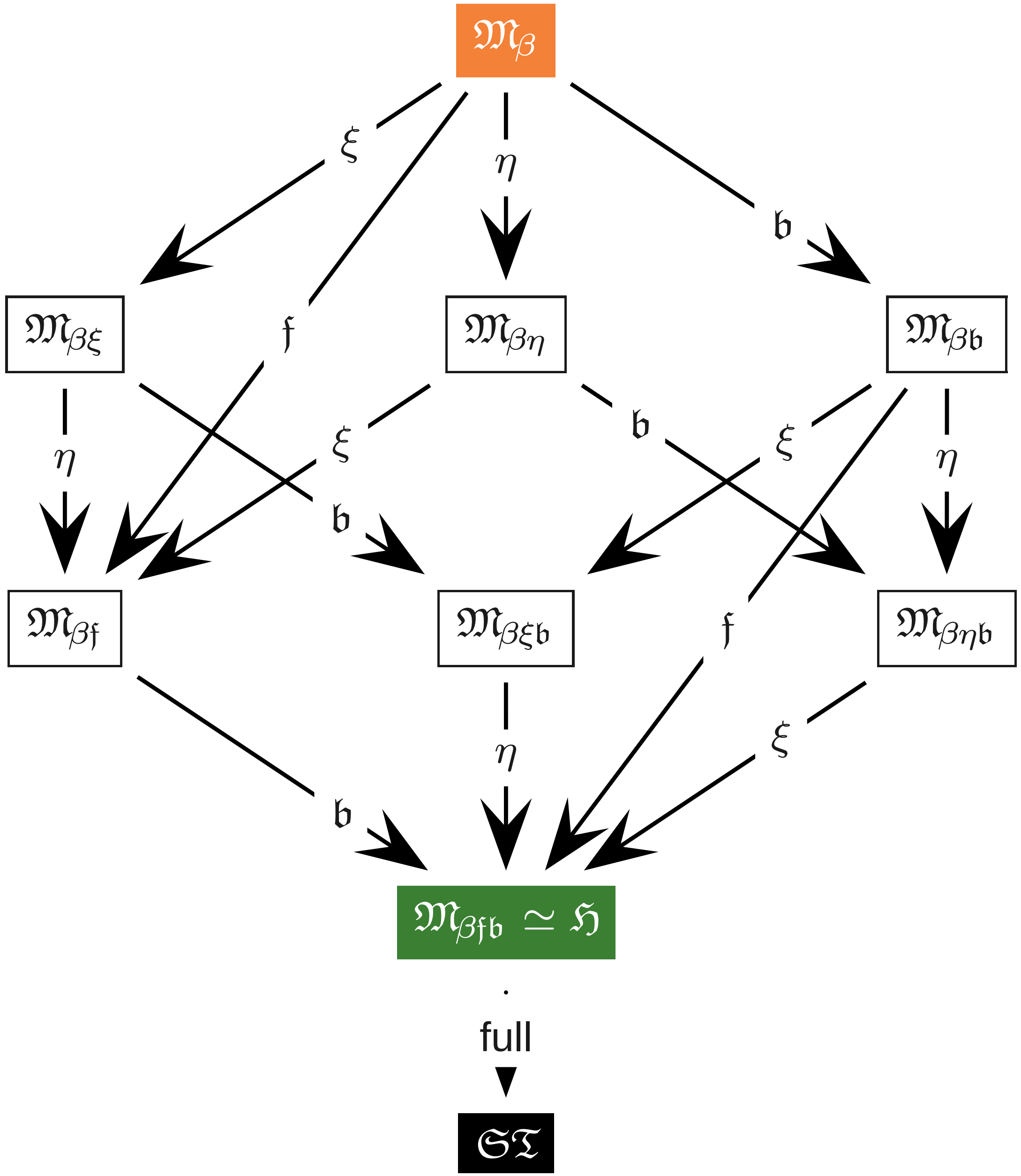
[Benzmüller-PhD-99]

[Brown-PhD-04]

[Benzm.BrownKohlhase-JSL-04]

$\mathfrak{M}_\beta$ model class for $\mathcal{S}$ fragment of elementary type theory

Semantics: HOL-CUBE



Landscape of HOL model classes

[Kohlhase-PhD-94]

[Benzmüller-PhD-99]

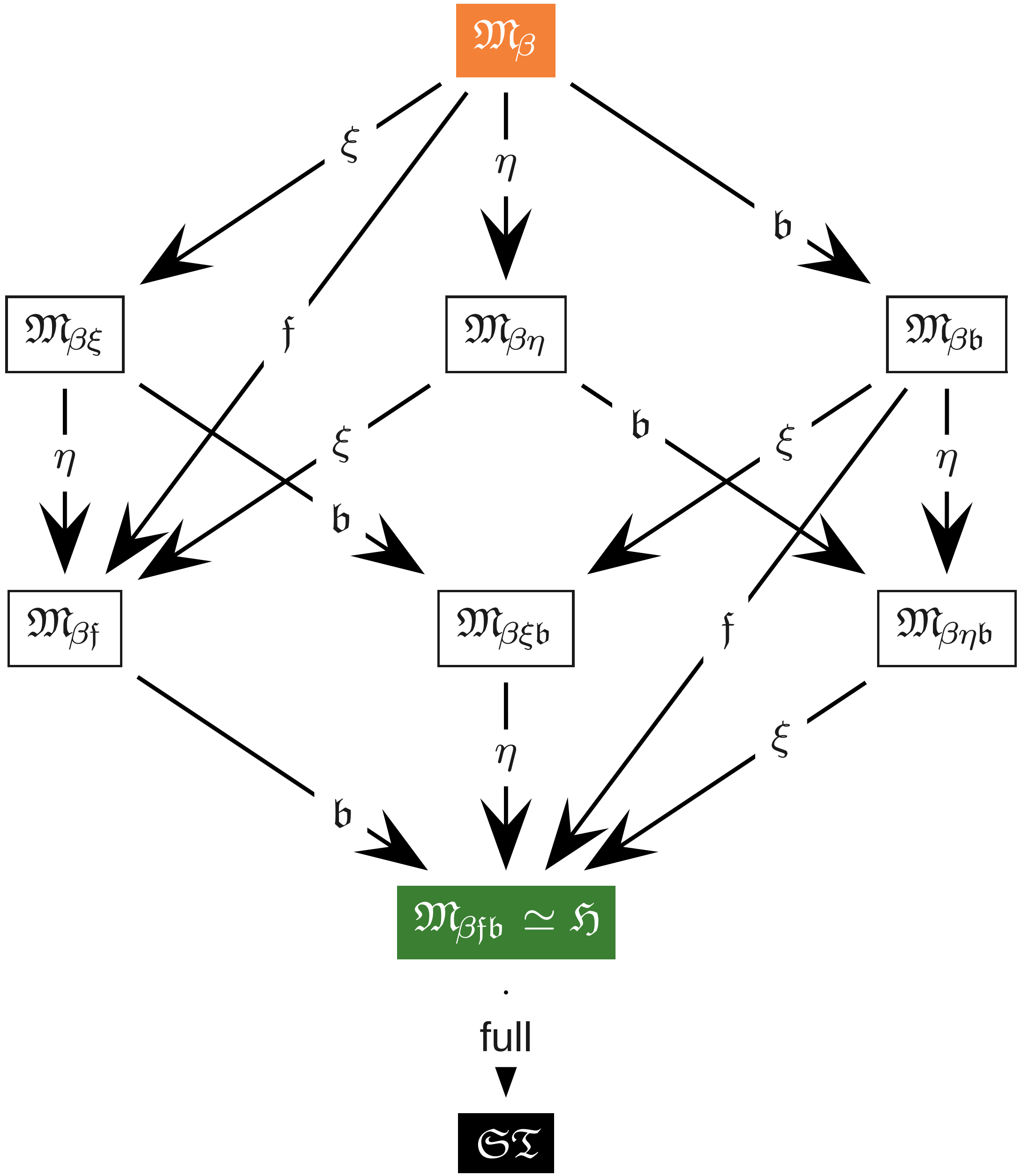
[Brown-PhD-04]

[Benzm.BrownKohlhase-JSL-04]

$\mathfrak{M}_{\beta}$ model class for $\mathcal{S}$ fragment of elementary type theory

$\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ model class for $\mathcal{S}$ fragment of extensional type theory (Henkin models)

Semantics: HOL-CUBE



Landscape of HOL model classes

[Kohlhase-PhD-94]

[Benzmüller-PhD-99]

[Brown-PhD-04]

[Benzm.BrownKohlhase-JSL-04]

$\mathfrak{M}_\beta$ model class for $\mathcal{S}$ fragment of elementary type theory

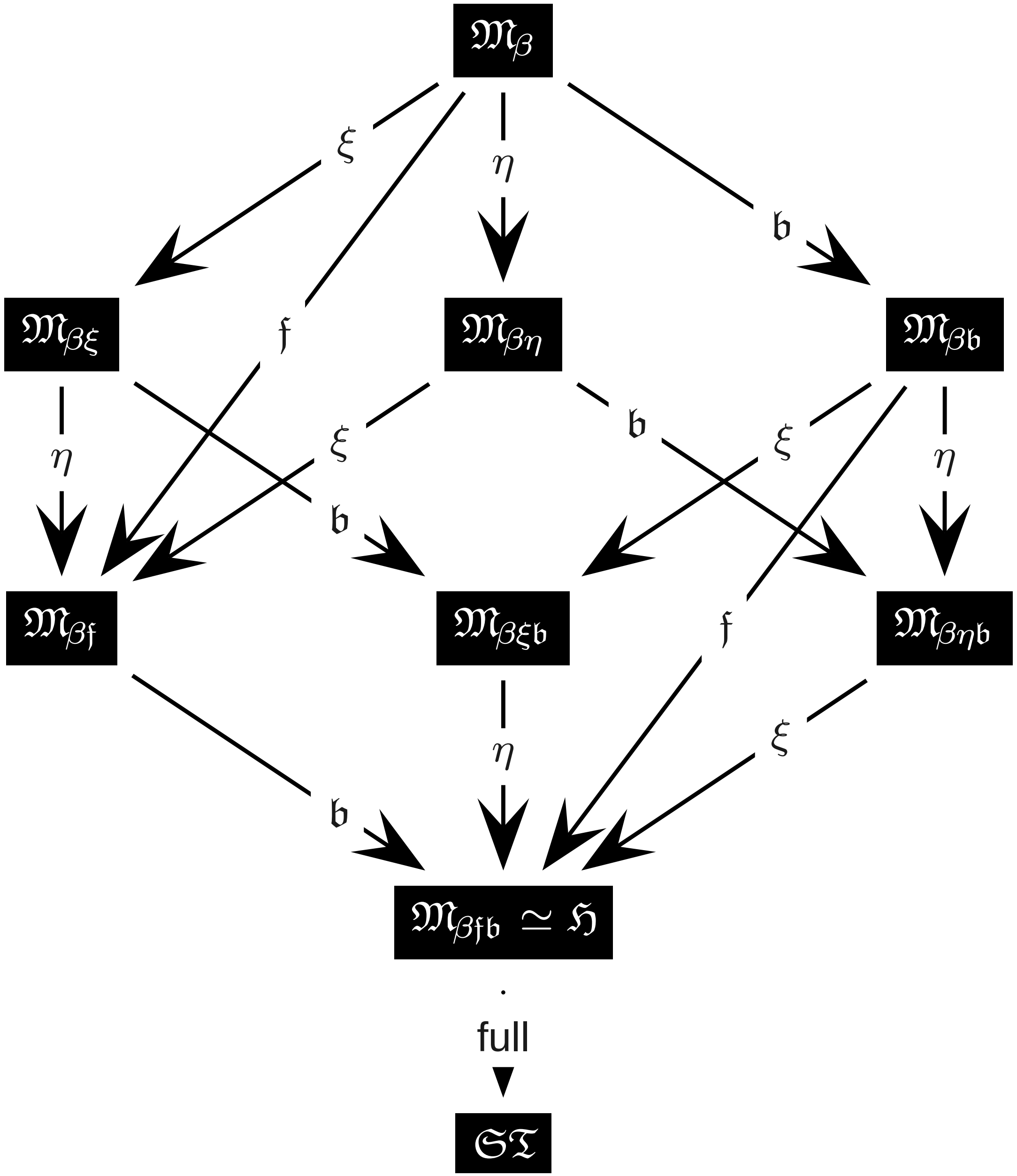
$\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ model class for $\mathcal{S}$ fragment of extensional type theory (Henkin models)

Signature $\mathcal{S}$ defined as

$$\{\top, \bot, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\} \cup \{\Pi^\alpha, \Sigma^\alpha, =^\alpha\}$$

(less logical connectives are possible)

Semantics: HOL-CUBE



β: models support β-equality

$\mathfrak{q}$: models provide identity relations

$\forall\alpha : \mathsf{id} \in \mathcal{D}_{\alpha\to\alpha\to o}$

Semantics: HOL-CUBE

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

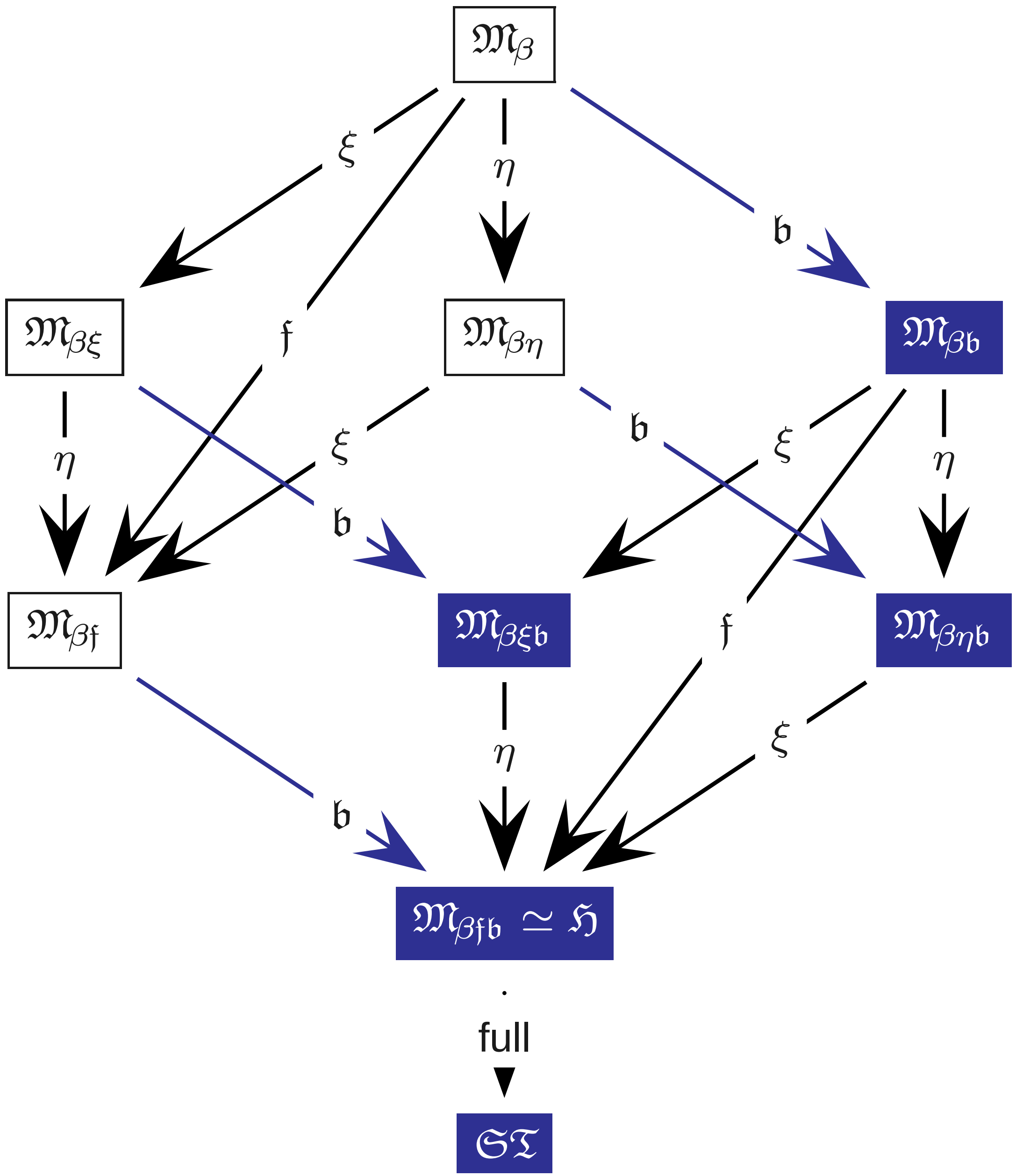
β: models support β-equality

$\mathfrak{q}$: models provide identity relations

$$\boxed{\forall\alpha : \mathsf{id} \in \mathcal{D}_{\alpha\to\alpha\to o}}$$

- [Andrews72]: without property $\mathfrak{q}$ Leibniz equality $\doteq$ not necessarily evaluates to identity relation — even in Henkin semantics

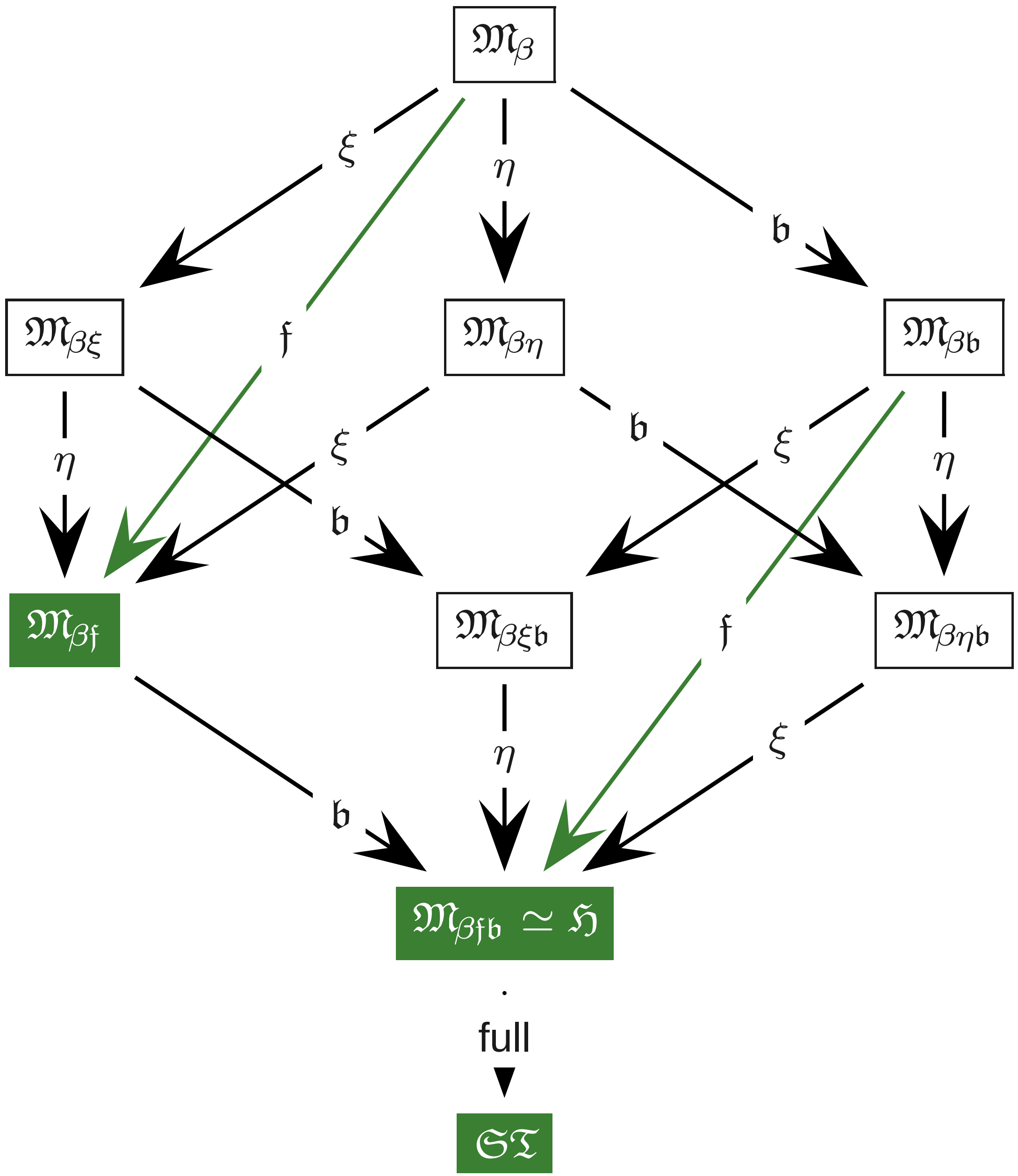
Semantics: HOL-CUBE



$\mathfrak{b}$: models are Boolean extensional

$$\mathcal{D}_o \equiv \{\bot, \top\}$$

Semantics: HOL-CUBE



$\mathfrak{f}$: models are functional

$$\forall f, g \in \mathcal{D}_{\beta\alpha} :$$
$$f \equiv g \text{ iff } f@a \equiv g@a \ (\forall a \in \mathcal{D}_\alpha)$$

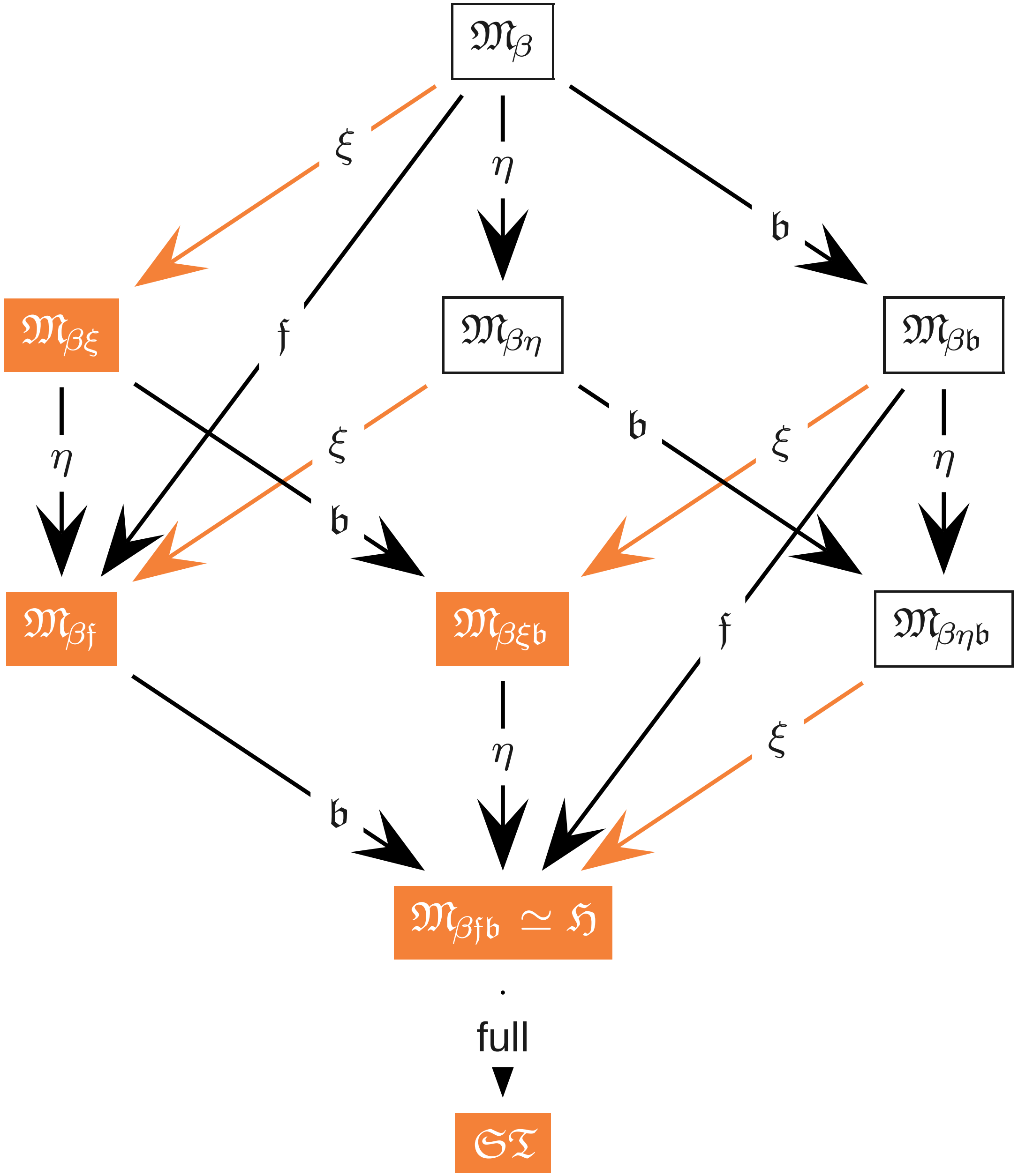
Semantics: HOL-CUBE

$$\begin{tikzcd}
 & & \mathfrak{M}_{\beta} \arrow[dll, "\xi"'] \arrow[ddll, "\mathfrak{f}"] \arrow[d, "\eta"] \arrow[drr, "\mathfrak{b}"] & & \\
\mathfrak{M}_{\beta\xi} \arrow[d, "\eta"] \arrow[drr, "\mathfrak{b}"] & & \mathfrak{M}_{\beta\eta} \arrow[dll, "\xi"'] \arrow[drr, "\mathfrak{b}"] & & \mathfrak{M}_{\beta\mathfrak{b}} \arrow[dll, "\xi"'] \arrow[ddll, "\mathfrak{f}"] \arrow[d, "\eta"] \\
\mathfrak{M}_{\beta\mathfrak{f}} \arrow[drr, "\mathfrak{b}"] & & \mathfrak{M}_{\beta\xi\mathfrak{b}} \arrow[d, "\eta"] & & \mathfrak{M}_{\beta\eta\mathfrak{b}} \arrow[dll, "\xi"'] \\
 & & \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}} \simeq \mathfrak{H} \arrow[d, dotted, "\text{full}"] & & \\
 & & \mathfrak{ST} & &
\end{tikzcd}$$

η: models are η-functional

$$\mathcal{E}_{\varphi}(\mathsf{A}) \equiv \mathcal{E}_{\varphi}(\mathsf{A}\downarrow_{\eta})$$

Semantics: HOL-CUBE



ξ: models are ξ-functional

$$\mathcal{E}_\varphi(\lambda X_\alpha\text{.}\ M_\beta) \equiv \mathcal{E}_\varphi(\lambda X_\alpha\text{.}\ N_\beta) \text{ iff}$$
$$\mathcal{E}_{\varphi,[a/X]}(M) \equiv \mathcal{E}_{\varphi,[a/X]}(N)\ (\forall a \in \mathcal{D}_\alpha)$$

HOL-CUBE: Abstract Consistency

- Abstract consistency properties are provided in [Benzm.BrownKohlhase-JSL-04]

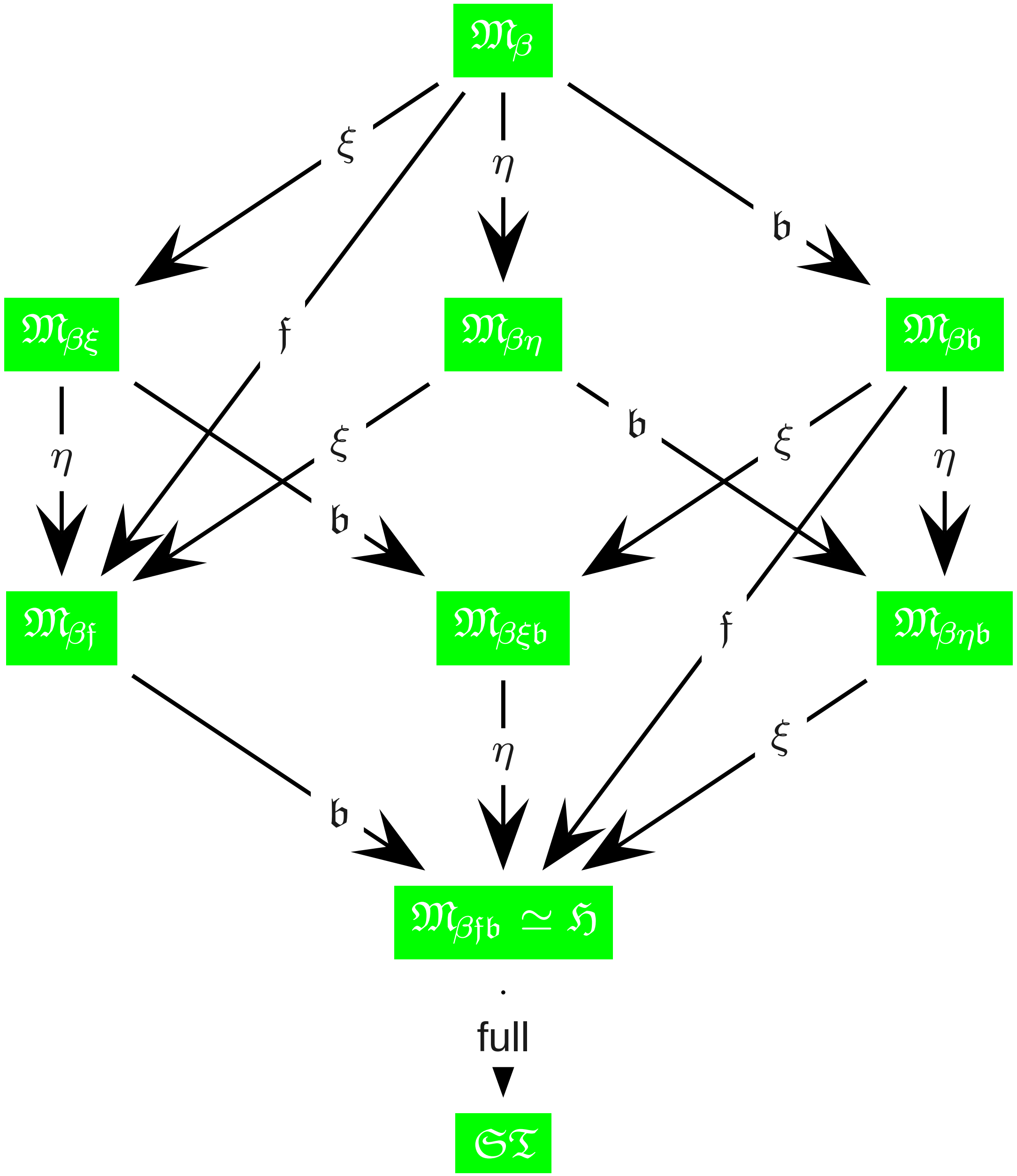
HOL-CUBE: Abstract Consistency

- Abstract consistency properties are provided in [Benzm.BrownKohlhase-JSL-04]
- They support completeness and soundness analysis of calculi by syntactical means for the HOL-CUBE

HOL-CUBE: Abstract Consistency

- Abstract consistency properties are provided in [Benzm.BrownKohlhase-JSL-04]
- They support completeness and soundness analysis of calculi by syntactical means for the HOL-CUBE
- Proposal:
  use the examples of this paper before trying a formal analysis

HOL-Problems requiring β



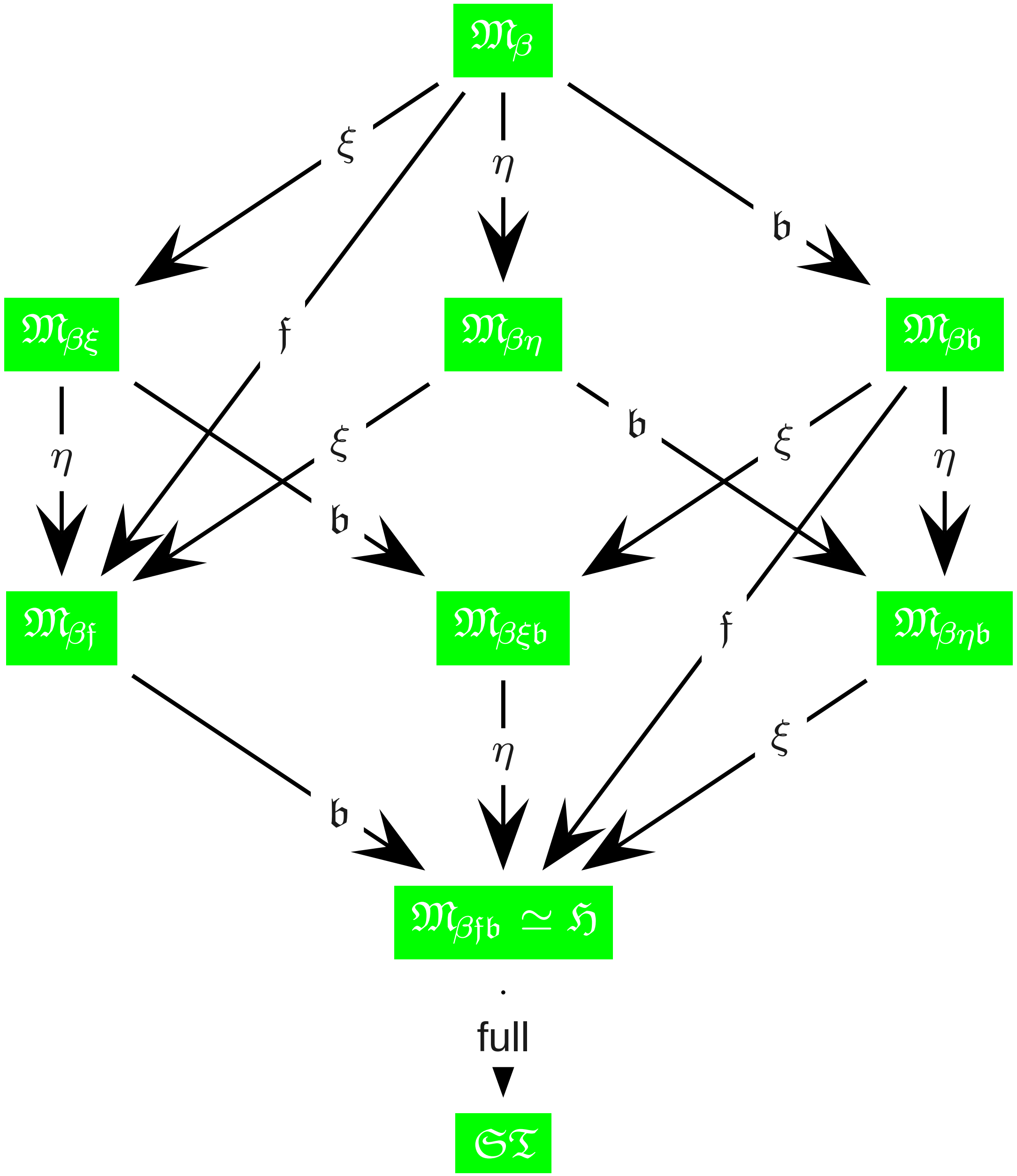
Church numerals:

$\overline{\mathsf{n}}^\alpha := (\lambda F_{\alpha\alpha}\lambda Y_\alpha.\ (F^n\ Y))$

$\overline{+} := \lambda M \lambda N \lambda F \lambda Y.\ MF(NFY)$

$\overline{\times} := \lambda M \lambda N \lambda F \lambda Z.\ N(MF)Z$

HOL-Problems requiring β



Church numerals:

$\overline{n}^{\alpha} := (\lambda F_{\alpha\alpha}\lambda Y_{\alpha}.\ (F^{n}\ Y))$

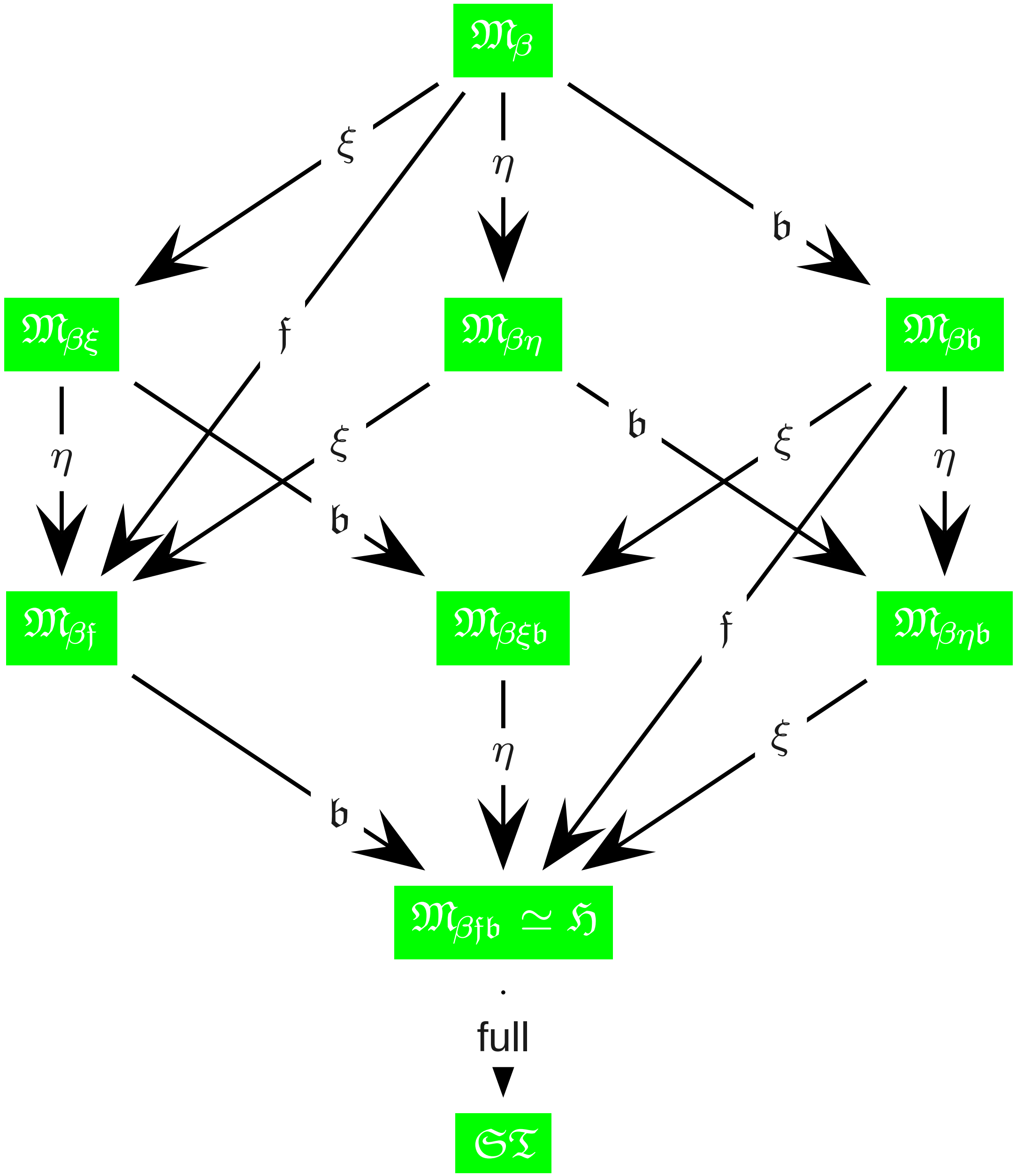
$\overline{+} := \lambda M\lambda N\lambda F\lambda Y.\ MF(NFY)$

$\overline{\times} := \lambda M\lambda N\lambda F\lambda Z.\ N(MF)Z$

Efficiency of β-conversion:

- $\overline{3}\overline{\times}\overline{4} \stackrel{*}{\doteq} \overline{5}\overline{+}\overline{7}$
- $(\overline{10}\overline{\times}\overline{10})\overline{\times}\overline{10} \stackrel{*}{\doteq} ((\overline{10}\overline{\times}\overline{5})\overline{+}(\overline{5}\overline{\times}\overline{10}))\overline{\times}\overline{10}$

HOL-Problems requiring β



Church numerals:

$\overline{n}^{\alpha} := (\lambda F_{\alpha\alpha} \lambda Y_{\alpha}.\, (F^{n}\ Y))$

$\overline{+} := \lambda M \lambda N \lambda F \lambda Y.\, MF(NFY)$

$\overline{\times} := \lambda M \lambda N \lambda F \lambda Z.\, N(MF)Z$

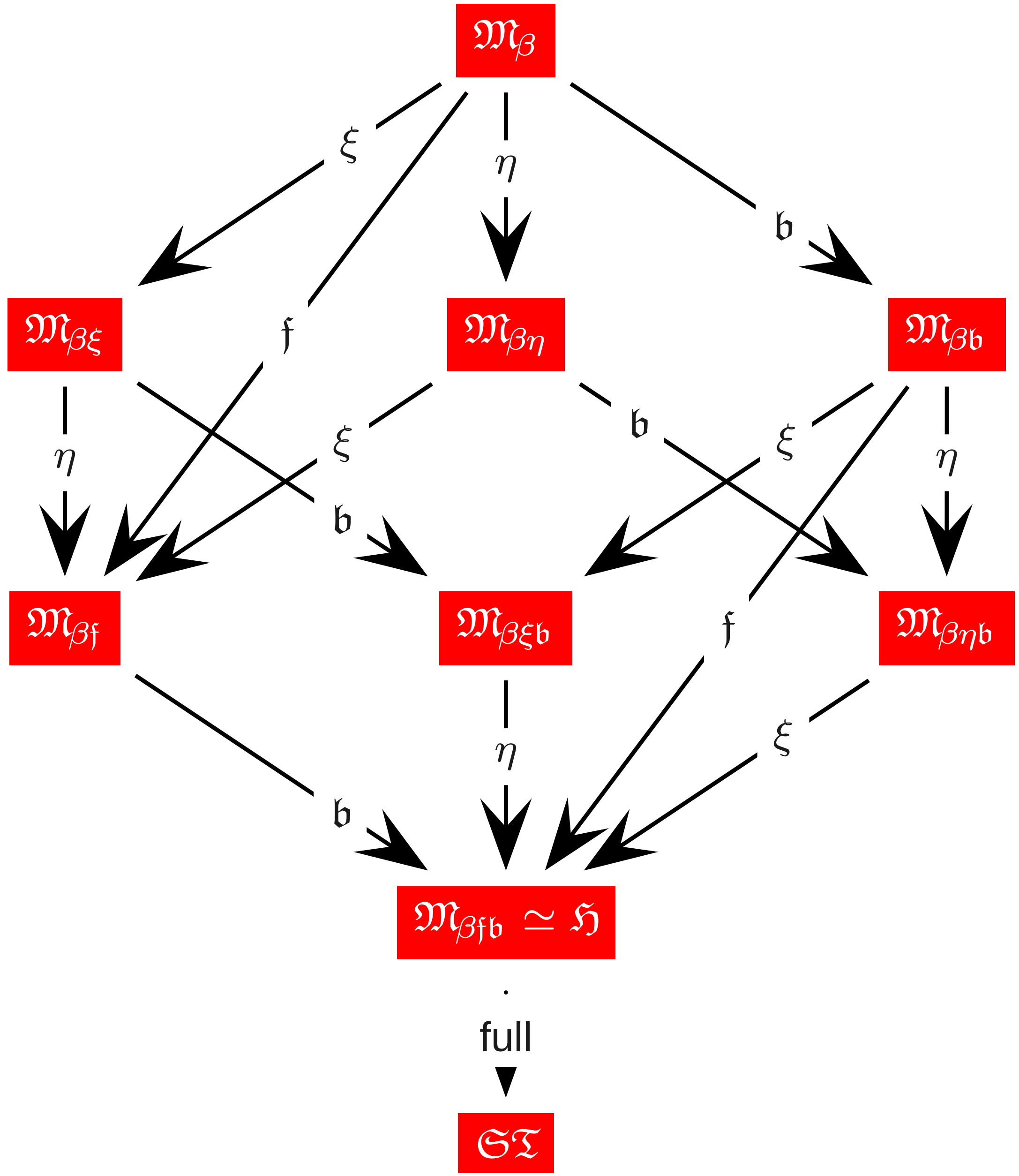
Efficiency of β-conversion:

- $\overline{3}\overline{\times}\overline{4} \stackrel{*}{=} \overline{5}\overline{+}\overline{7}$
- $(\overline{10}\overline{\times}\overline{10})\overline{\times}\overline{10} \stackrel{*}{=} ((\overline{10}\overline{\times}\overline{5})\overline{+}(\overline{5}\overline{\times}\overline{10}))\overline{\times}\overline{10}$

Pre-unification with β-conversion:

- $\exists N_{(\iota\rightarrow\iota)\rightarrow\iota\rightarrow\iota}.\, ((N\overline{\times}\overline{1}) \stackrel{*}{=} \overline{1})$
  (two solutions if only β;
  one solution if $\beta\eta$)
- $\exists N.\, N\overline{\times}\overline{4} \stackrel{*}{=} \overline{5}\overline{+}\overline{7}$
- $\exists H.\, ((H\ \overline{2})\overline{3}) \stackrel{*}{=} \overline{6} \wedge ((H\ \overline{1})\overline{2}) \stackrel{*}{=} \overline{2}$
- $\exists N, M.\, N\overline{\times}\overline{4} \stackrel{*}{=} \overline{5}\overline{+}M$
  (infinitely many solutions!)

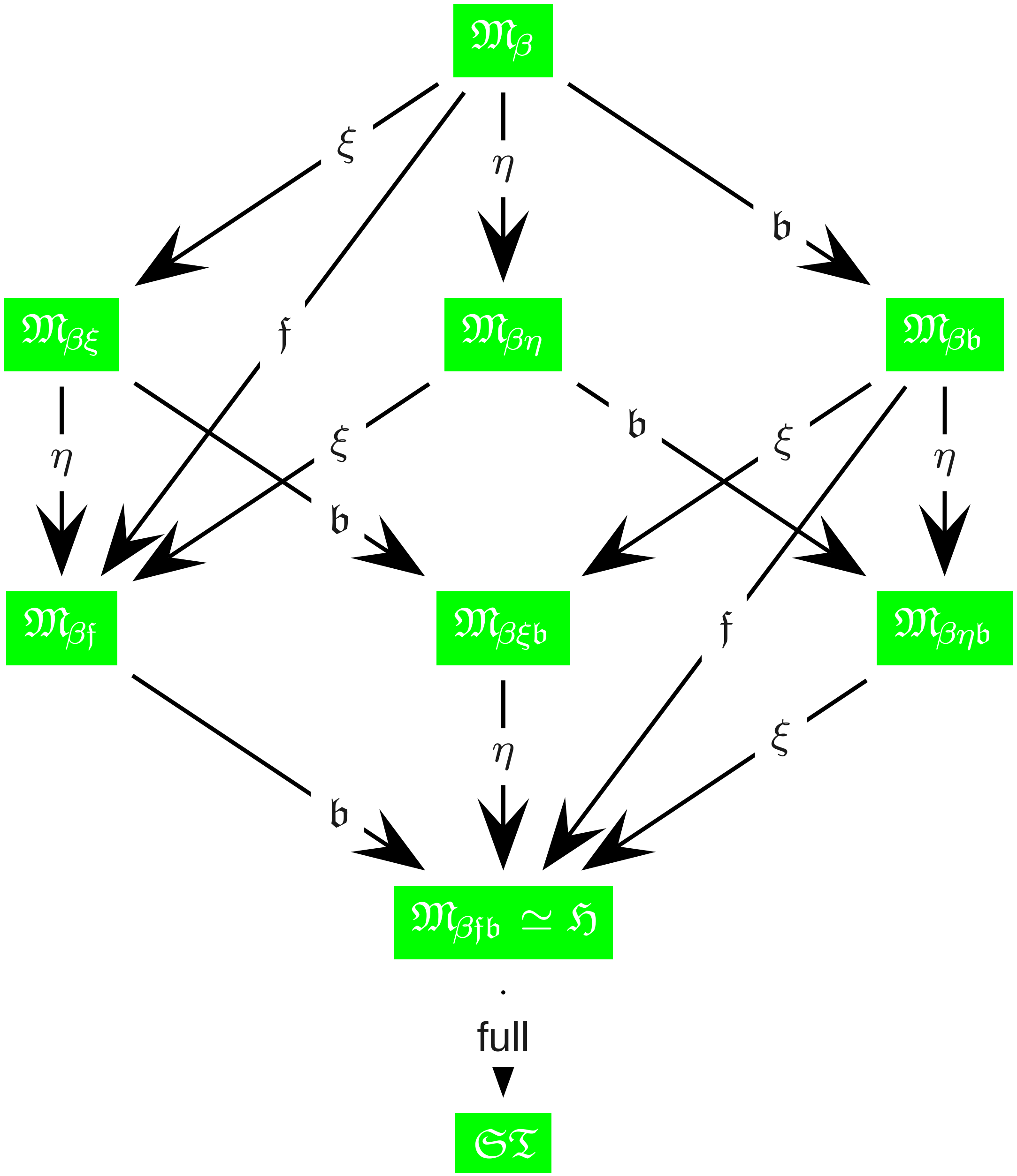
HOL-Non-Problems



Some non-theorems:

- essentially FOL
- apply to all model classes
- address
  - Skolemization
  - axiom of choice

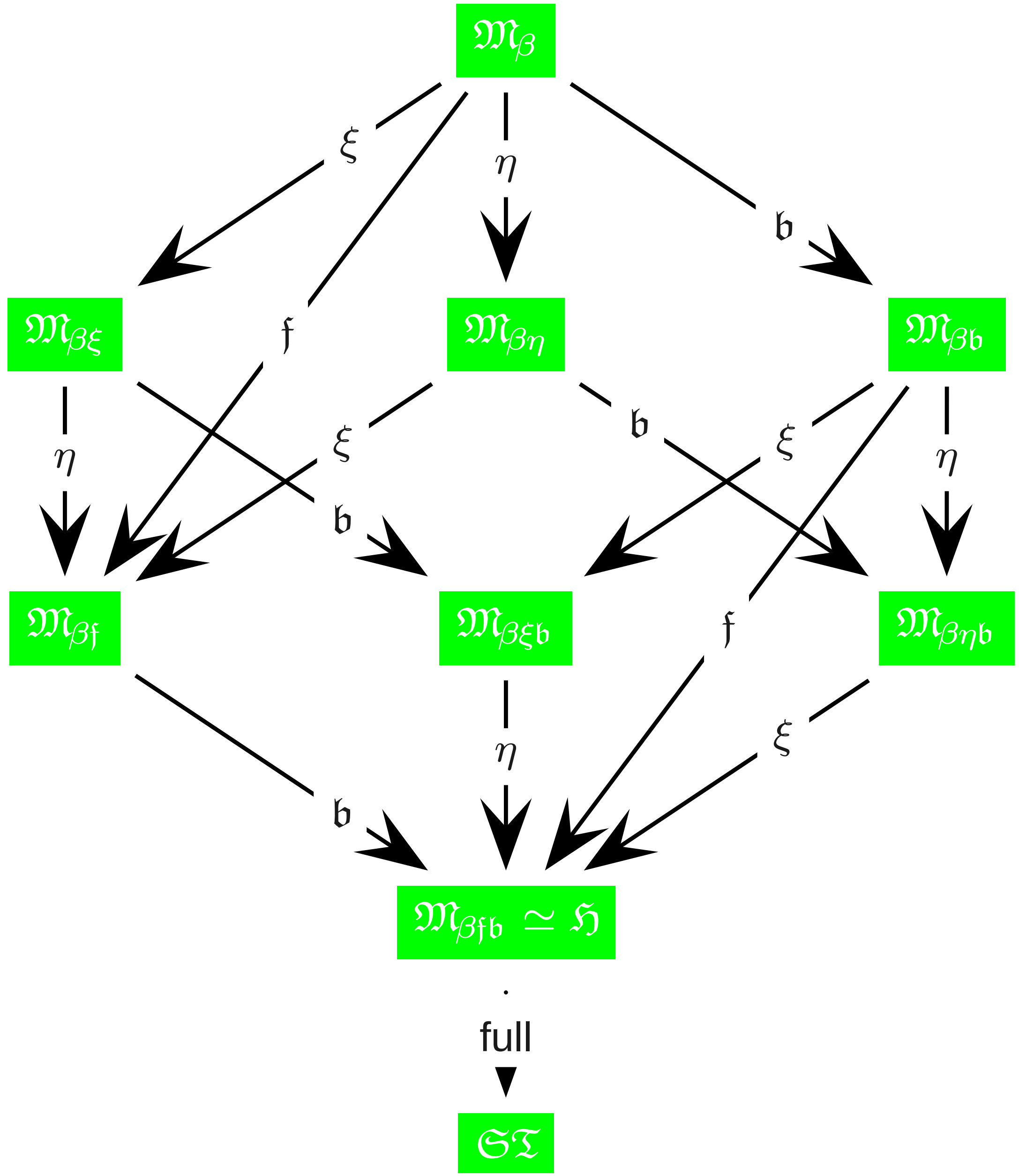
HOL-Problems requiring β



$\stackrel{*}{\doteq}$ is equivalence relation

- $\forall X_\alpha \text{.}\, X \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha \text{.}\, X \stackrel{*}{\doteq} Y \Rightarrow Y \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha, Z_\alpha \text{.}\, (X \stackrel{*}{\doteq} Y \wedge Y \stackrel{*}{\doteq} Z) \Rightarrow X \stackrel{*}{\doteq} Z$

HOL-Problems requiring β



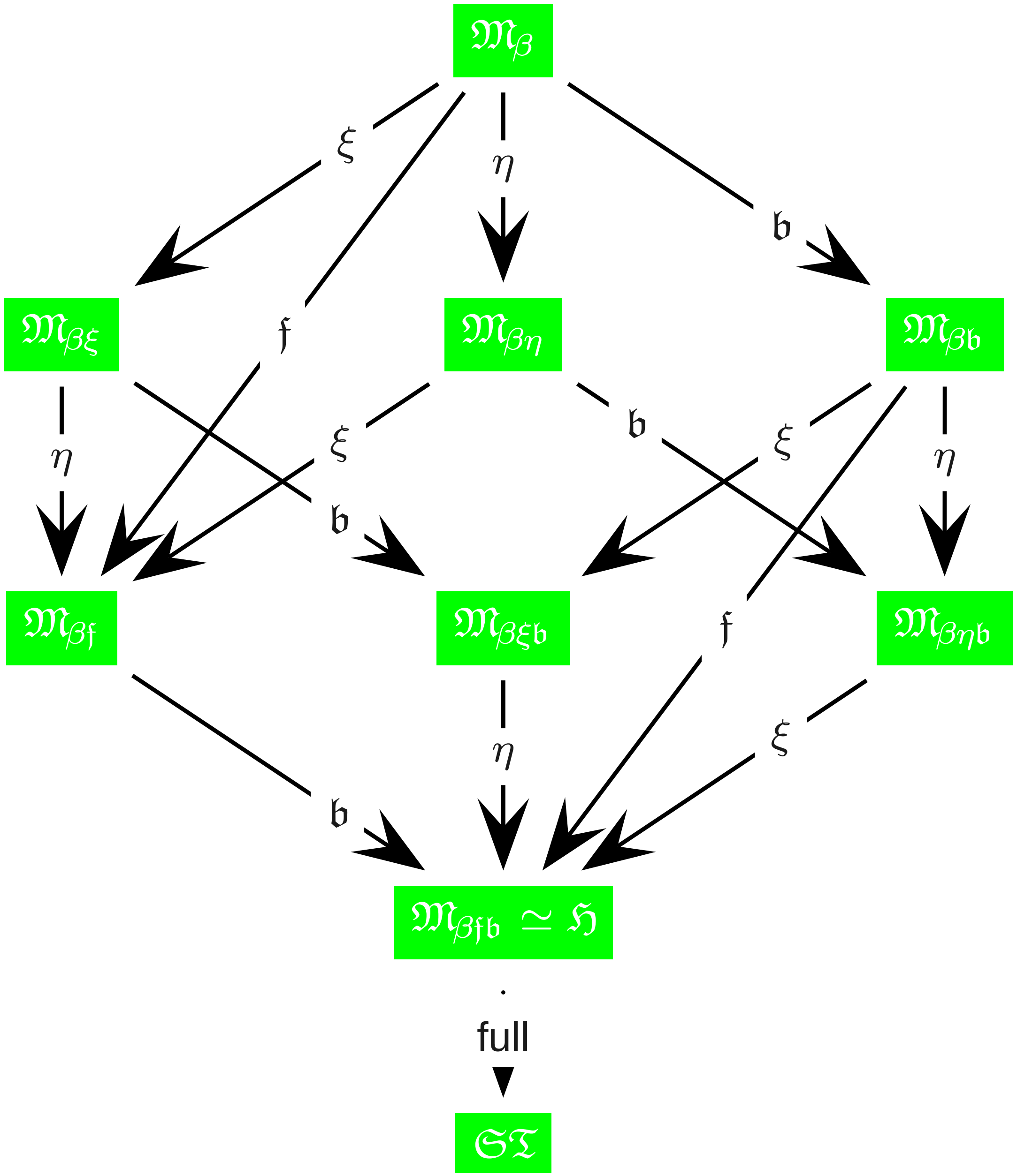
$\stackrel{*}{\doteq}$ is equivalence relation

- $\forall X_\alpha.\, X \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha.\, X \stackrel{*}{\doteq} Y \Rightarrow Y \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha, Z_\alpha.\, (X \stackrel{*}{\doteq} Y \wedge Y \stackrel{*}{\doteq} Z) \Rightarrow X \stackrel{*}{\doteq} Z$

$\stackrel{*}{\doteq}$ is congruence relation

- $\forall X_\alpha, Y_\alpha, F_{\alpha\alpha}.\, X \stackrel{*}{\doteq} Y \Rightarrow (FX) \stackrel{*}{\doteq} (FY)$
- $\forall X_\alpha, Y_\alpha, P_{o\alpha}.\, X \stackrel{*}{\doteq} Y \wedge (PX) \Rightarrow (PY)$

HOL-Problems requiring β



$\stackrel{*}{\doteq}$ is equivalence relation

- $\forall X_\alpha.\, X \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha.\, X \stackrel{*}{\doteq} Y \Rightarrow Y \stackrel{*}{\doteq} X$
- $\forall X_\alpha, Y_\alpha, Z_\alpha.\, (X \stackrel{*}{\doteq} Y \wedge Y \stackrel{*}{\doteq} Z) \Rightarrow X \stackrel{*}{\doteq} Z$

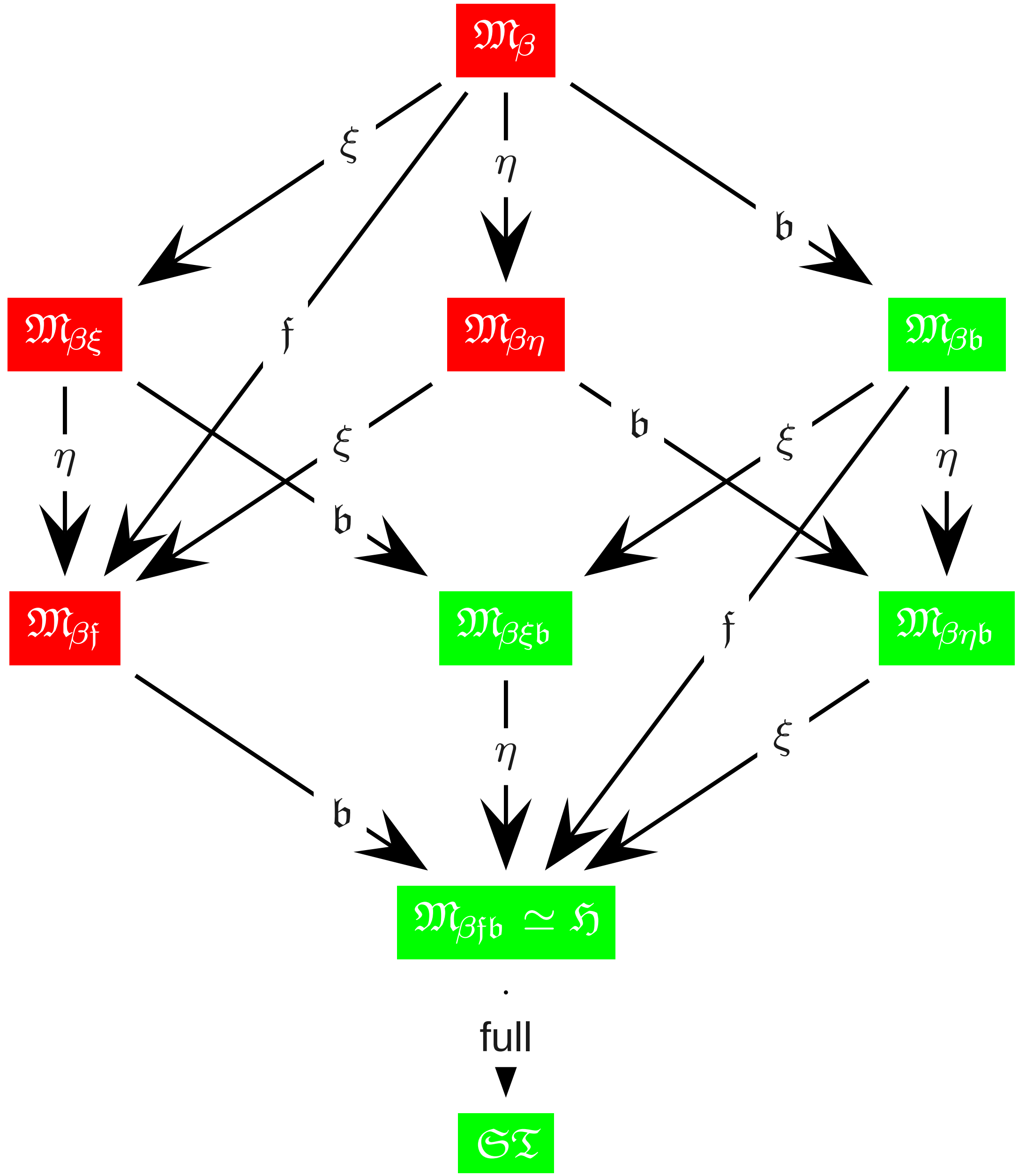
$\stackrel{*}{\doteq}$ is congruence relation

- $\forall X_\alpha, Y_\alpha, F_{\alpha\alpha}.\, X \stackrel{*}{\doteq} Y \Rightarrow (FX) \stackrel{*}{\doteq} (FY)$
- $\forall X_\alpha, Y_\alpha, P_{o\alpha}.\, X \stackrel{*}{\doteq} Y \wedge (PX) \Rightarrow (PY)$

Trivial directions of Boolean and functional extensionality

- $\forall A_o, B_o.\, A \stackrel{*}{\doteq} B \Rightarrow (A \Leftrightarrow B)$
- $\forall F_{\beta\alpha}, G_{\beta\alpha}.\, F \stackrel{*}{\doteq} G \Rightarrow (\forall X_\alpha.\, FX \stackrel{*}{\doteq} GX)$

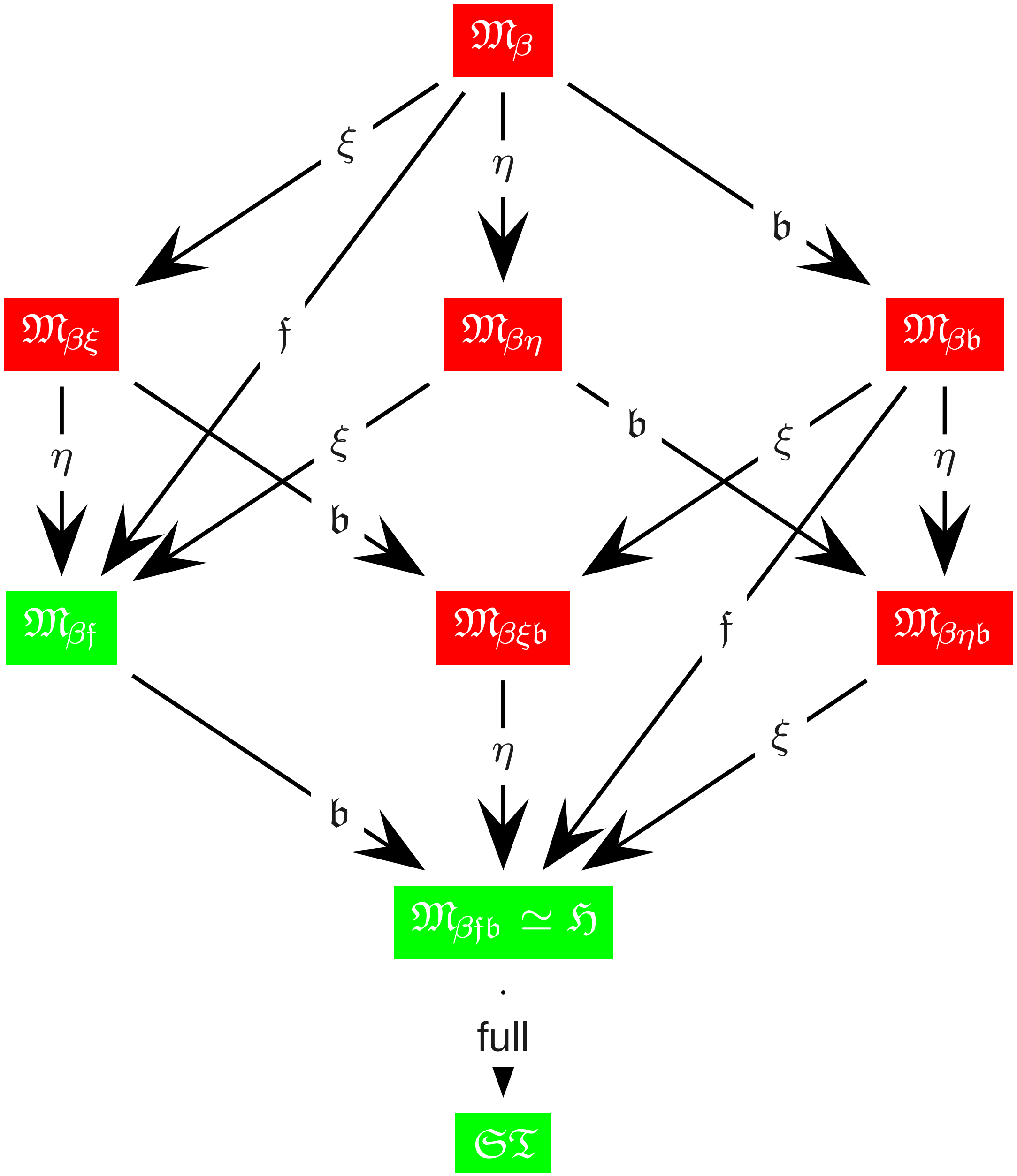
HOL-Problems requiring $\mathfrak{b}$



Non-trivial direction of Boolean extensionality

- $\forall A_o, B_o.\ (A \Leftrightarrow B) \Rightarrow A \stackrel{*}{=} B$

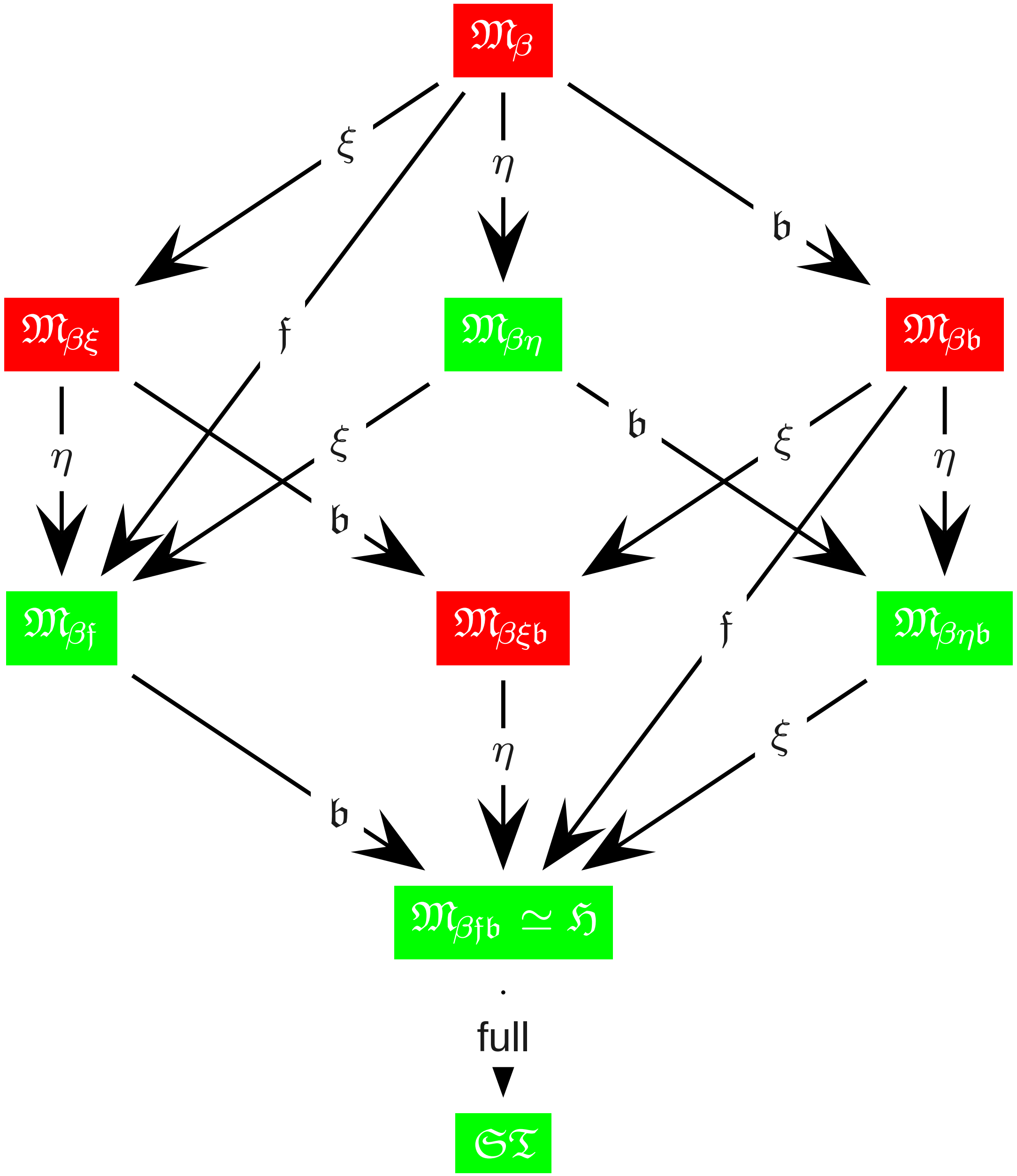
HOL-Problems requiring $\mathfrak{f}$



Non-trivial direct. of functional extensionality

- $\forall F_{\beta\alpha}, G_{\beta\alpha}.\ (\forall X_{\alpha}.\ FX \stackrel{*}{\doteq} GX) \Rightarrow F \stackrel{*}{\doteq} G$

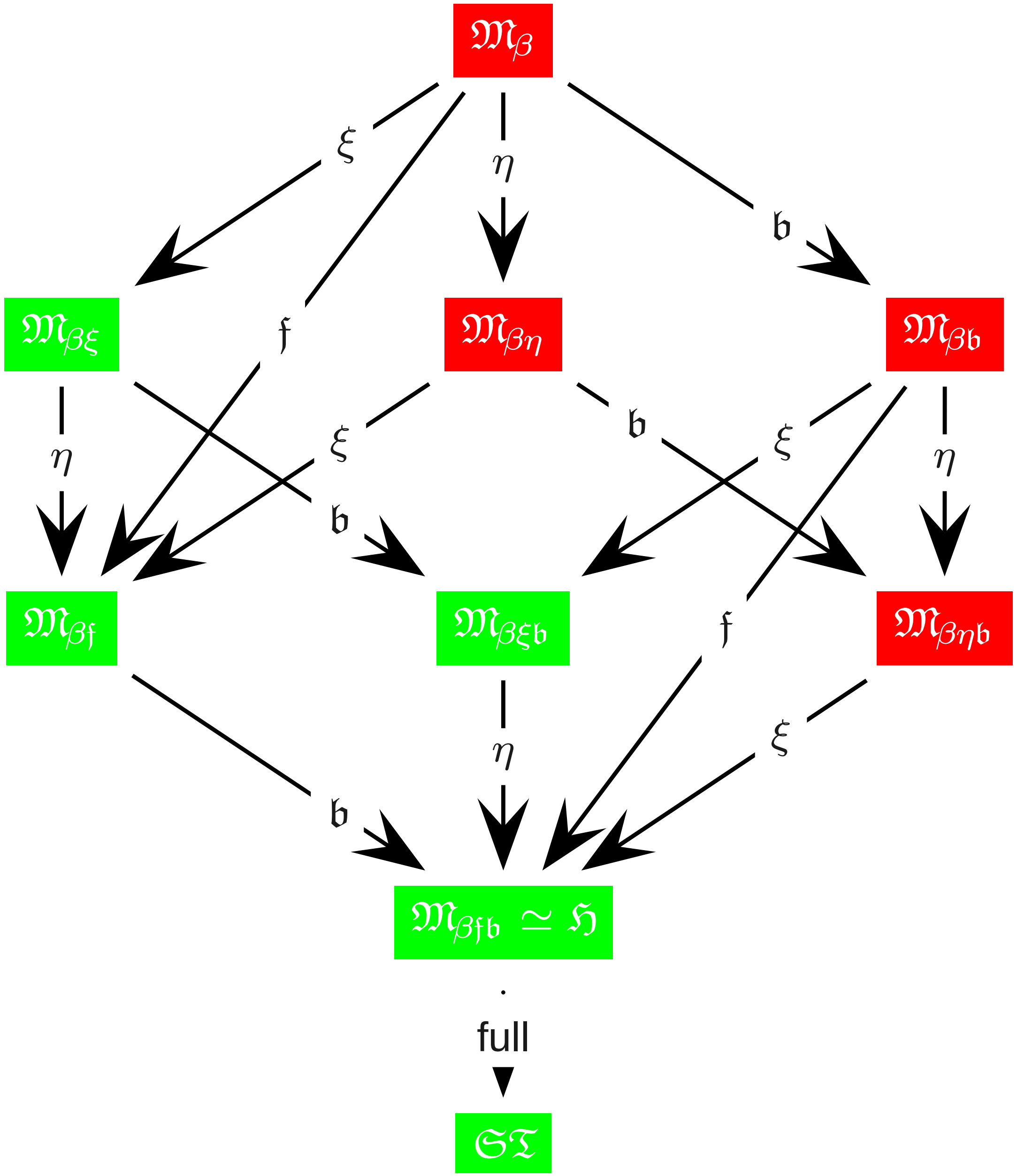
HOL-Problems requiring η



Example requiring property η

- $(p_{o(\iota\iota)}(\lambda X_\iota \text{.}\, f_{\iota\iota} X)) \Rightarrow (p\ f)$

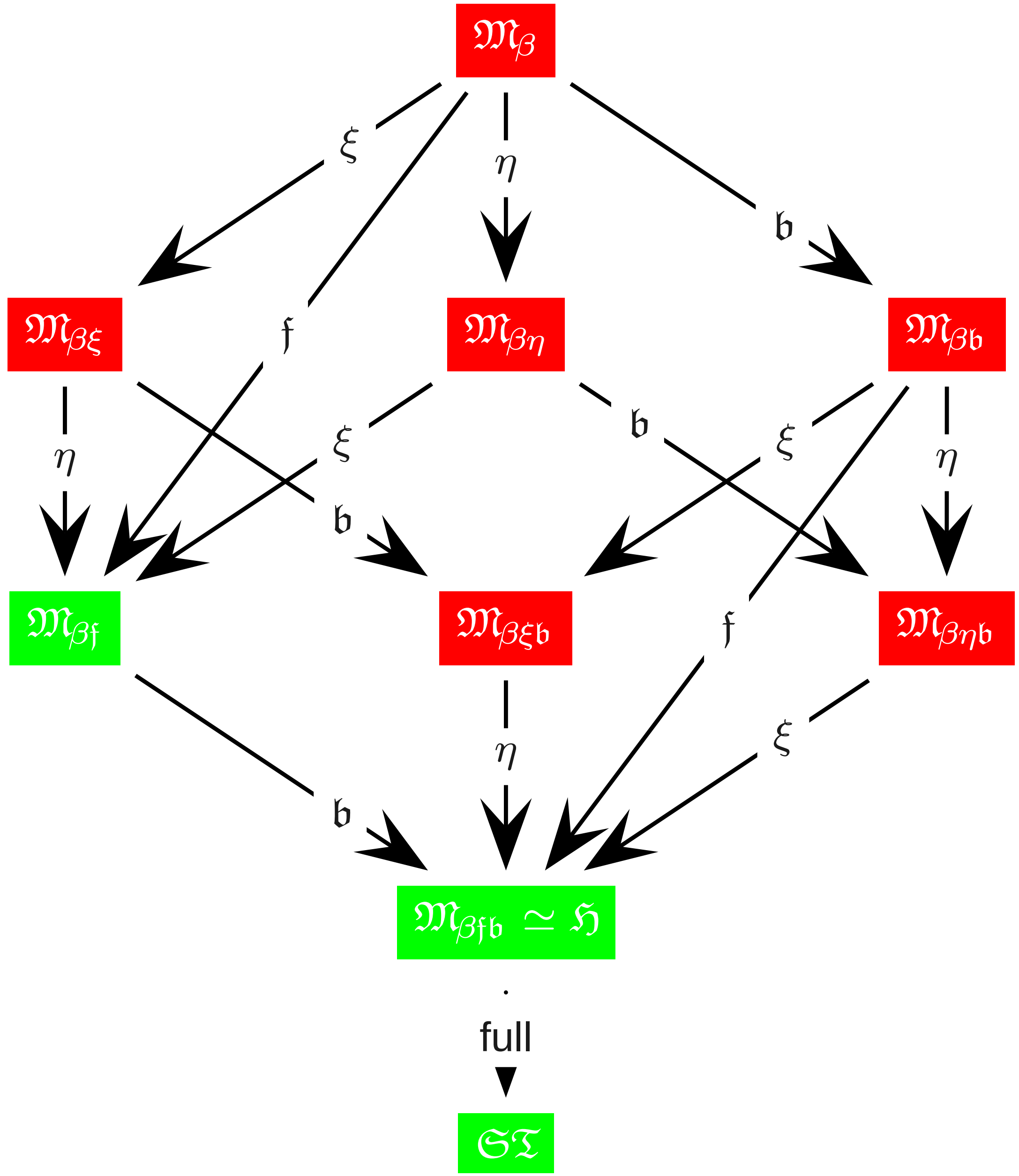
HOL-Problems requiring ξ



Example requiring property ξ (and $\mathfrak{q}$!)

- $(\forall X_\iota.\ (f_{\iota\iota} X) \stackrel{*}{\doteq} X) \wedge p_{o(\iota\iota)}(\lambda X_\iota.\ X)$
  $\Rightarrow p(\lambda X_\iota.\ f X)$

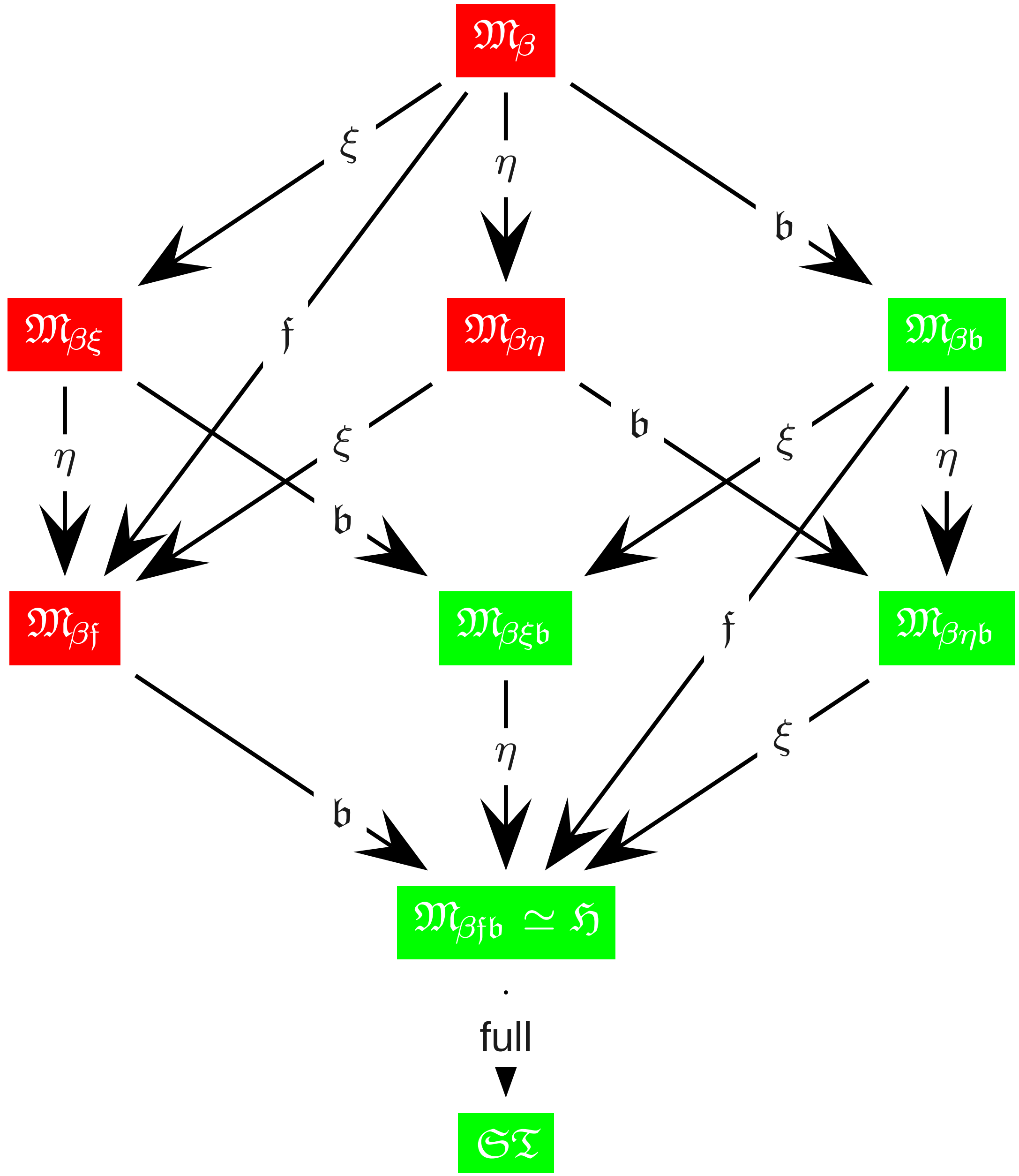
HOL-Problems requiring $\mathfrak{f}$



Example requiring property $\mathfrak{f}$ (and $\mathfrak{q}$!)

- $(\forall X_\iota \text{.}\, (f_{\iota\iota} X) \stackrel{*}{\doteq} X) \wedge p_{o(\iota\iota)} (\lambda X_\iota \text{.}\, X)$
  $\Rightarrow (p\ f)$

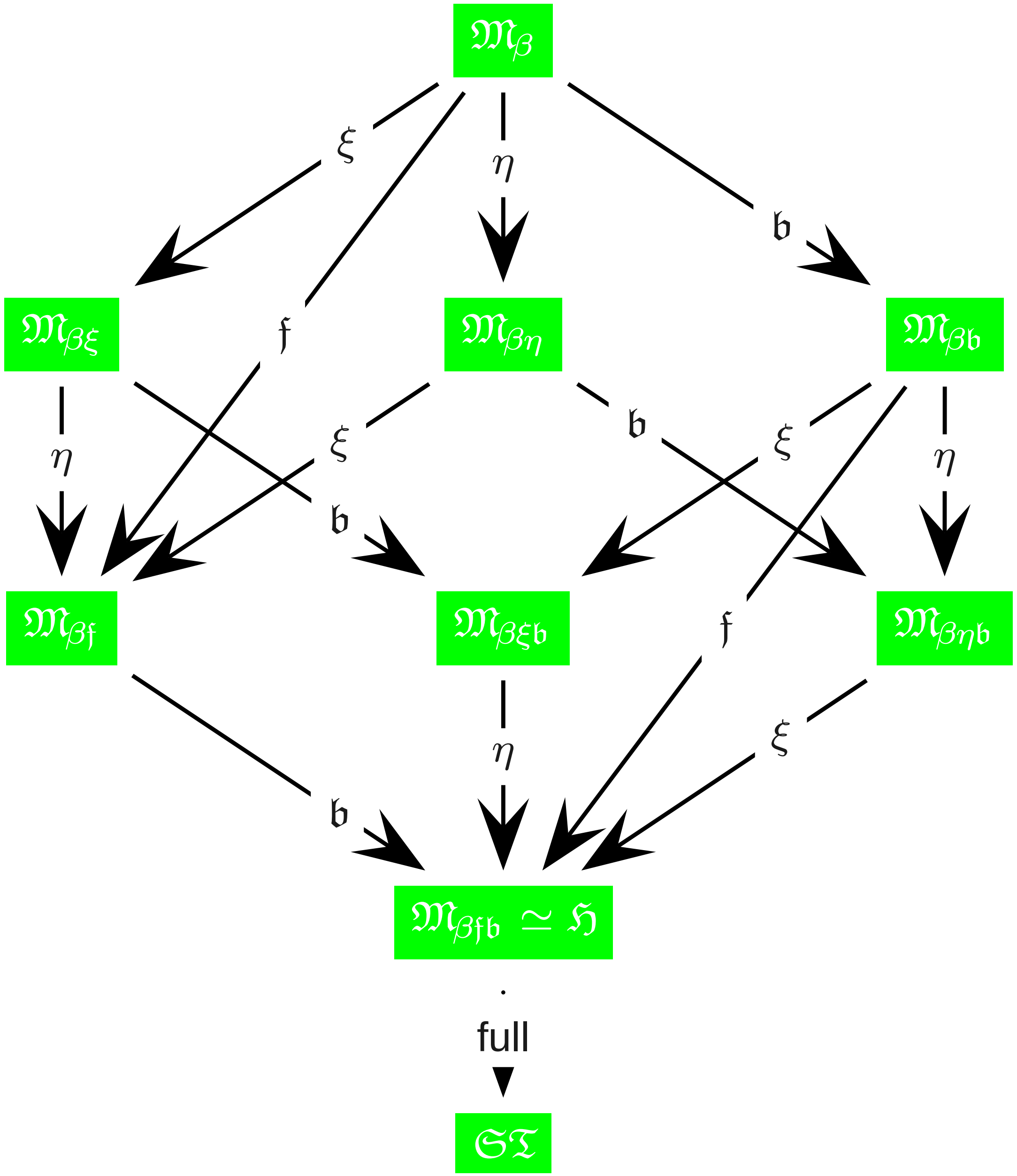
HOL-Problems requiring $\mathfrak{b}$



Examples requiring property $\mathfrak{b}$

- $(p_{oo}\ a_o) \wedge (p\ b_o) \Rightarrow (p\ (a \wedge b))$
- $\neg(a \stackrel{*}{=} \neg a)$ (in particular $\neg(a = \neg a)$)
- $(h_{\iota o}((h\top) \stackrel{*}{=} (h\bot))) \stackrel{*}{=} (h\bot)$

HOL-Problems: Other Examples

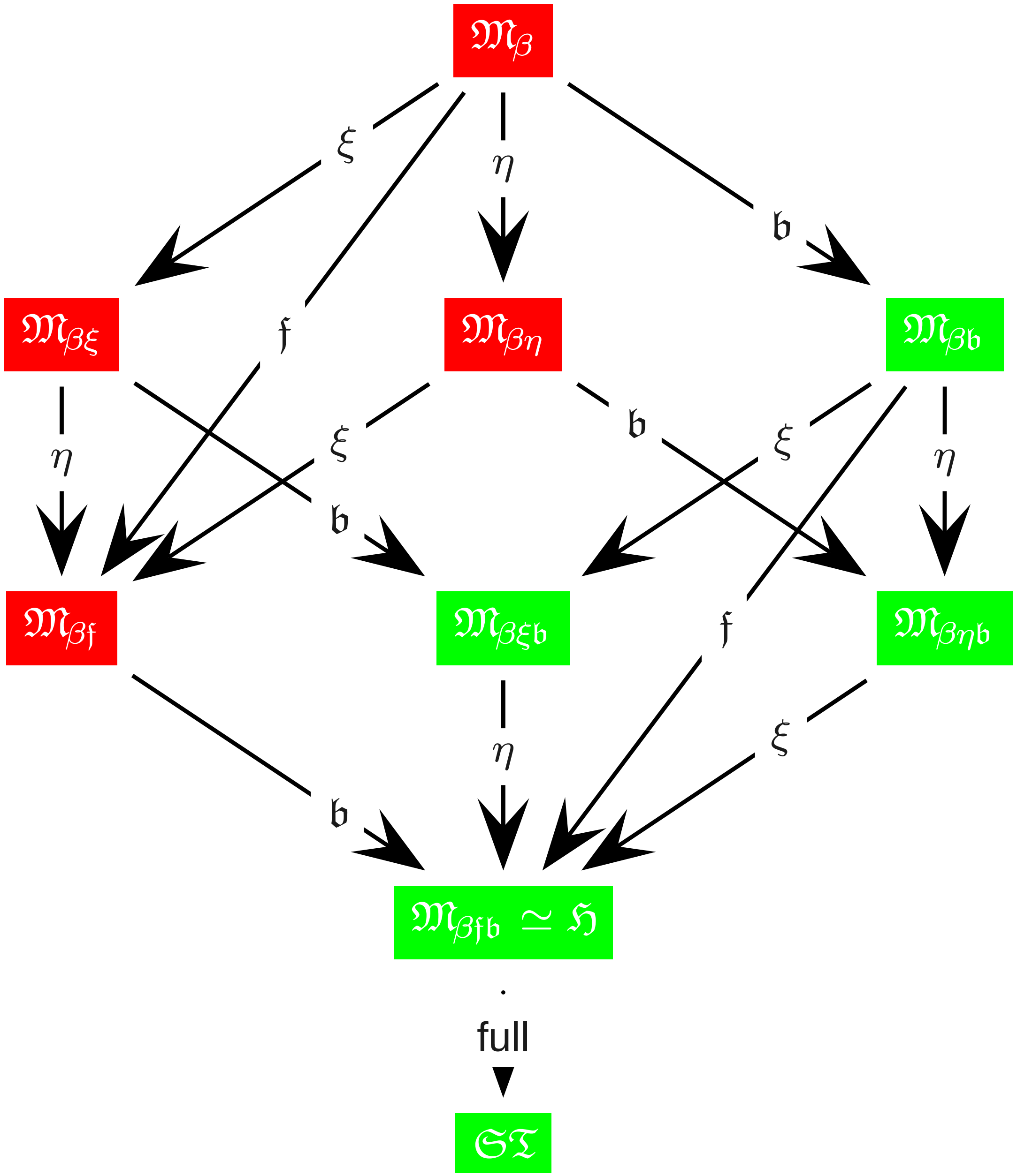


Playing with DeMorgan's Law:

- $\forall X, Y. X \wedge Y \Leftrightarrow \neg(\neg X \vee \neg Y)$

'Ok' for all model classes

HOL-Problems: DeMorgan's Law

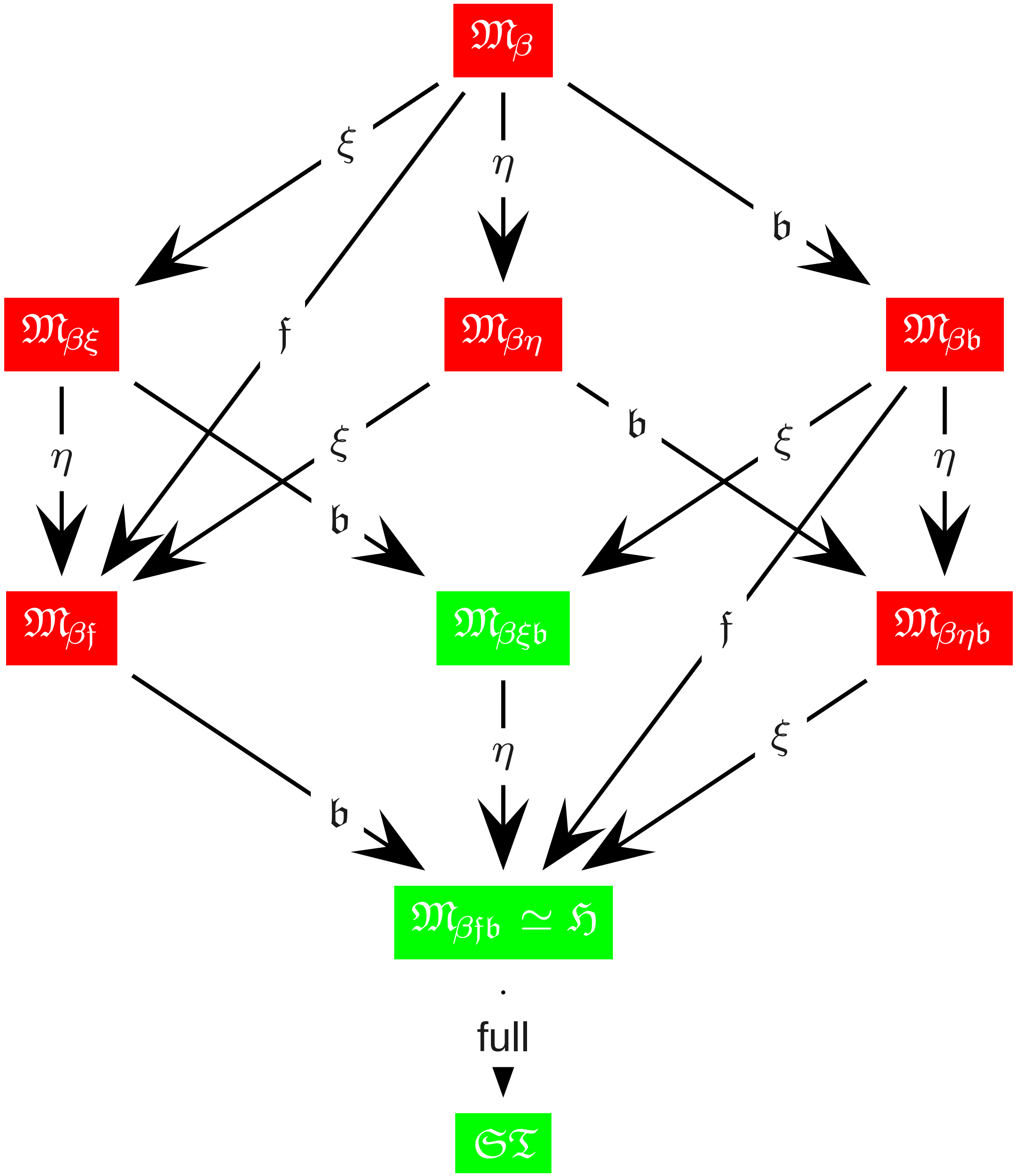


Playing with DeMorgan's Law:

- $\forall X, Y.\ X \land Y \Leftrightarrow \neg(\neg X \lor \neg Y)$
- $\forall X, Y.\ X \land Y \stackrel{*}{=} \neg(\neg X \lor \neg Y)$

requires $\mathfrak{b}$

HOL-Problems: DeMorgan’s Law

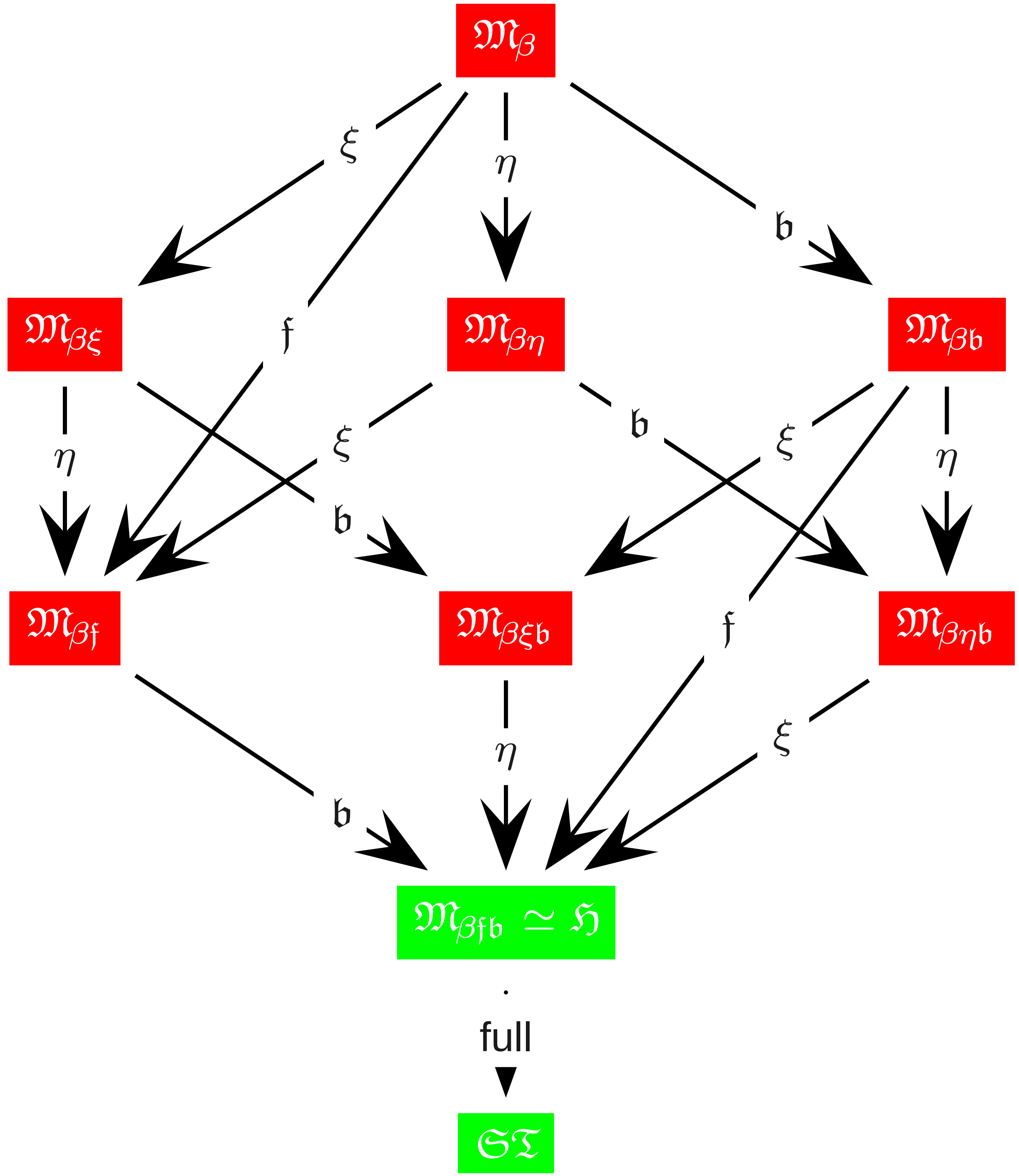


Playing with DeMorgan’s Law:

- $\forall X, Y.\ X \wedge Y \Leftrightarrow \neg(\neg X \vee \neg Y)$
- $\forall X, Y.\ X \wedge Y \stackrel{*}{\doteq} \neg(\neg X \vee \neg Y)$
- $(\lambda U \lambda V.\ U \wedge V) \stackrel{*}{\doteq} (\lambda X \lambda Y.\ \neg(\neg X \vee \neg Y))$

requires $\mathfrak{b}$ and ξ

HOL-Problems: DeMorgan's Law

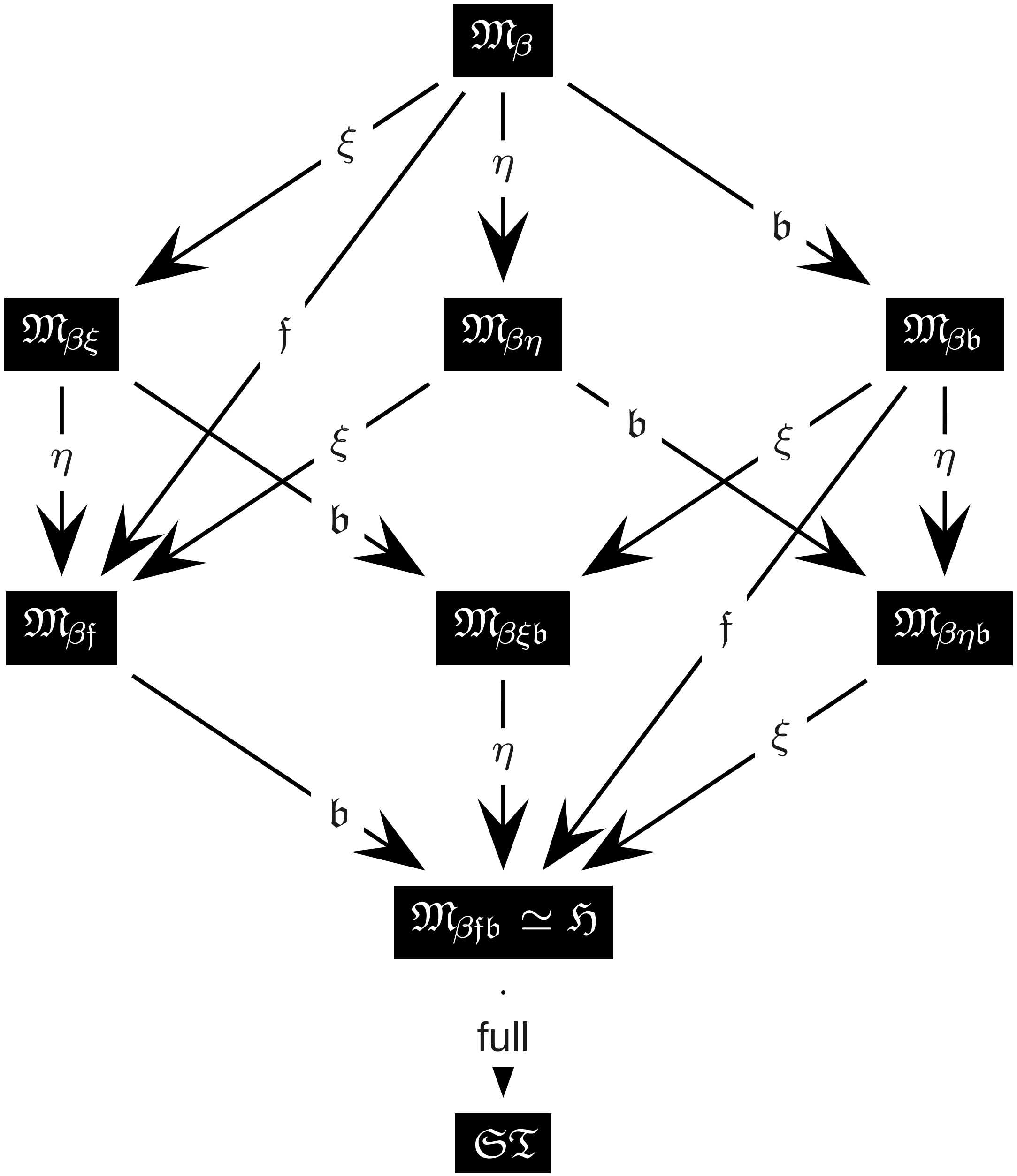


Playing with DeMorgan's Law:

- $\forall X, Y.\, X \wedge Y \Leftrightarrow \neg(\neg X \vee \neg Y)$
- $\forall X, Y.\, X \wedge Y \stackrel{*}{\doteq} \neg(\neg X \vee \neg Y)$
- $(\lambda U \lambda V.\, U \wedge V) \stackrel{*}{\doteq} (\lambda X \lambda Y.\, \neg(\neg X \vee \neg Y))$
- $\wedge \stackrel{*}{\doteq} (\lambda X \lambda Y.\, \neg(\neg X \vee \neg Y))$

$\boxed{\text{requires } \mathfrak{b} \text{ and } \mathfrak{f}}$

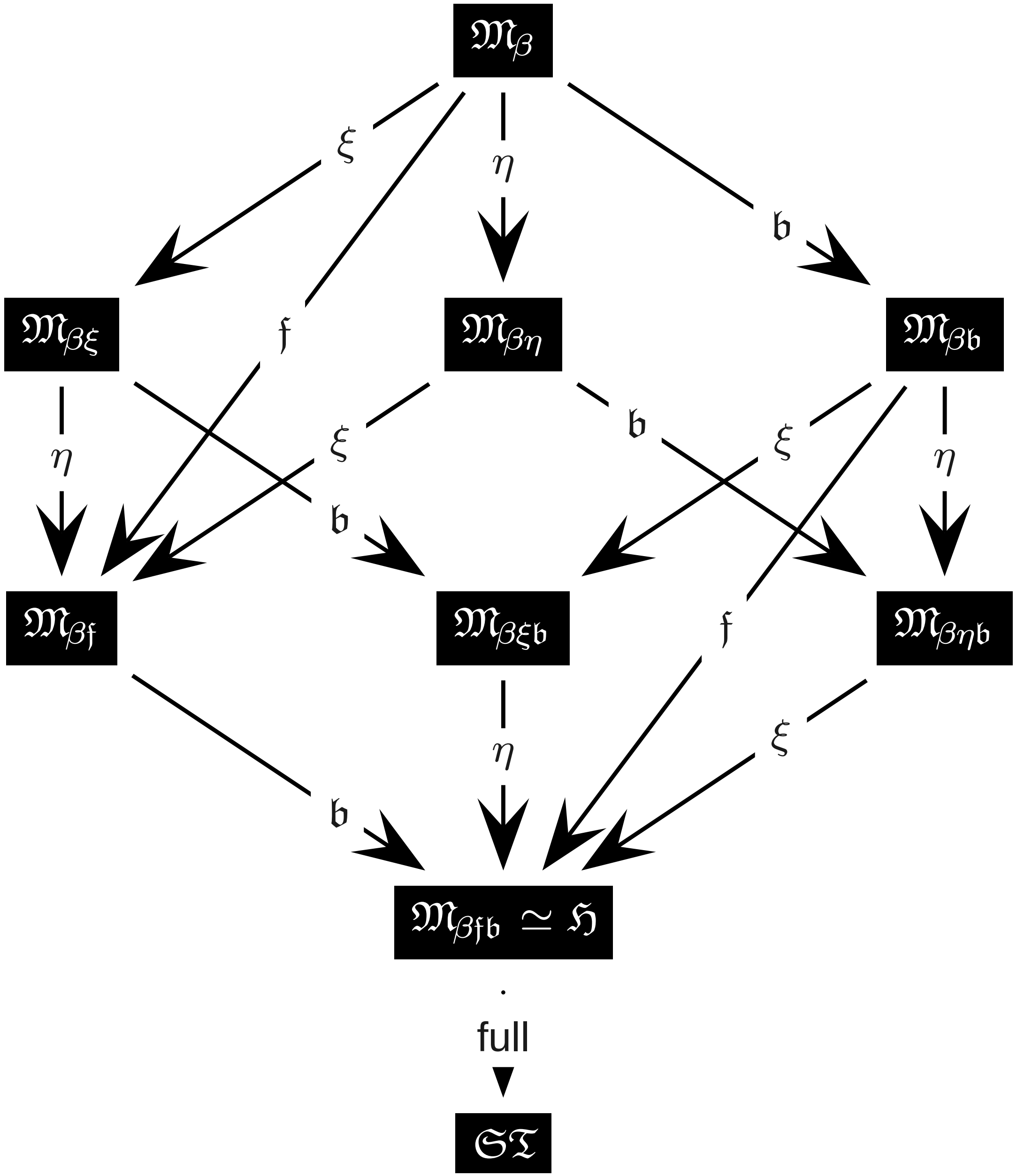
HOL-Problems: Set Comprehension



Set comprehension

- big challenge for automation
- [Benzm.BrownKohlhase-Draft-05] set instantiations can be used to simulate cut-rule if one of the following axioms is given: comprehension, induction, extensionality, choice, description
- dependend on logical constants in $\mathcal{S}$

HOL-Problems: Set Comprehension



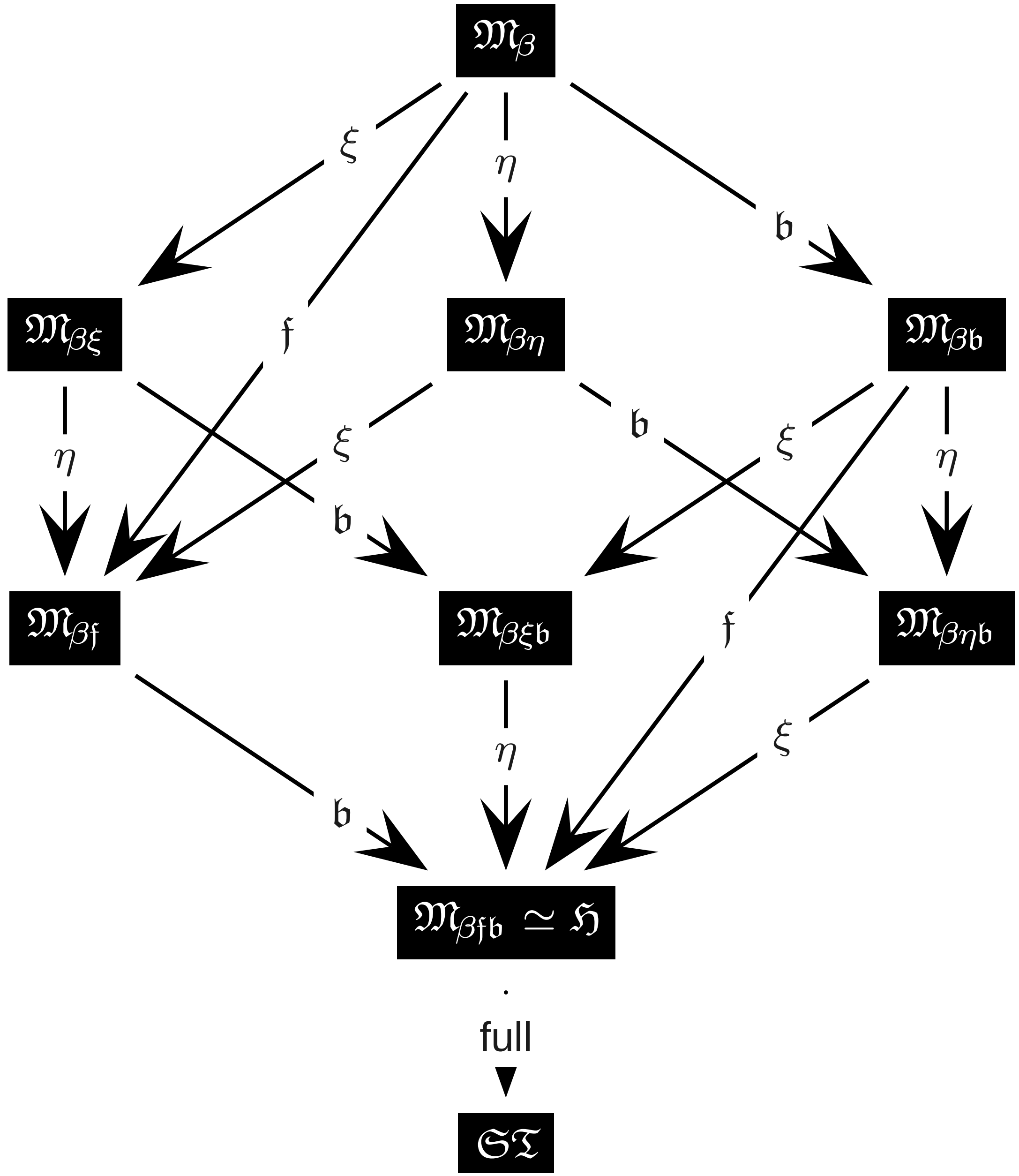
Set comprehension

- big challenge for automation
- [Benzm.BrownKohlhase-Draft-05] set instantiations can be used to simulate cut-rule if one of the following axioms is given: comprehension, induction, extensionality, choice, description
- dependend on logical constants in $\mathcal{S}$

In the remainder

- signature $\mathcal{S}$ varying
- no property $\mathfrak{q}$ assumed

HOL-Problems: Set Comprehension



Set comprehension

- big challenge for automation
- [Benzm.BrownKohlhase-Draft-05] set instantiations can be used to simulate cut-rule if one of the following axioms is given: comprehension, induction, extensionality, choice, description
- dependend on logical constants in $\mathcal{S}$

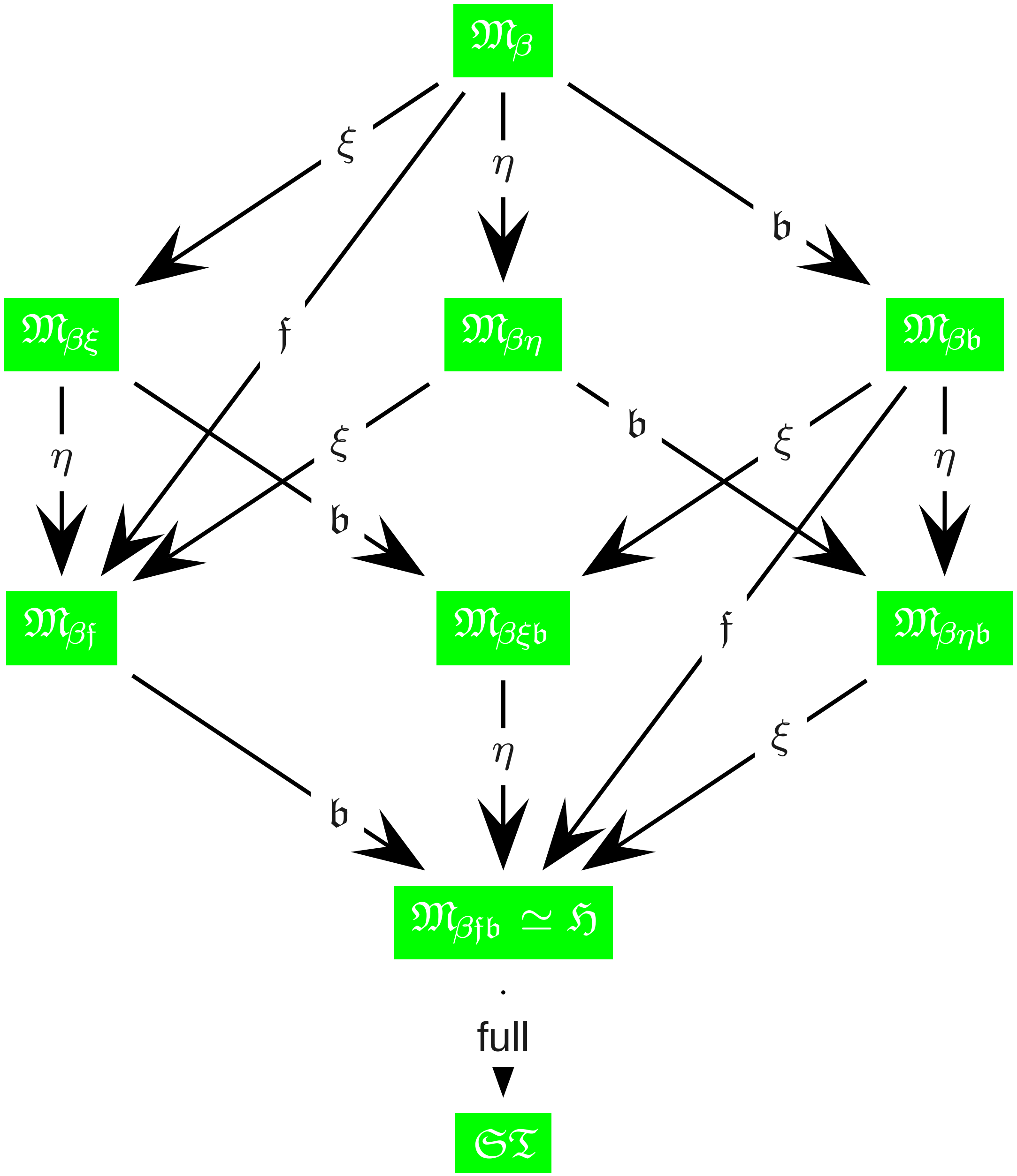
In the remainder

- signature $\mathcal{S}$ varying
- no property $\mathfrak{q}$ assumed

External vs. internal logical constants

- if $\neg \notin \mathcal{S}$:
  $\neg$ refers to 'external' symbol
  $\mathcal{M} \models \neg A$ means $\mathcal{M} \not\models A$

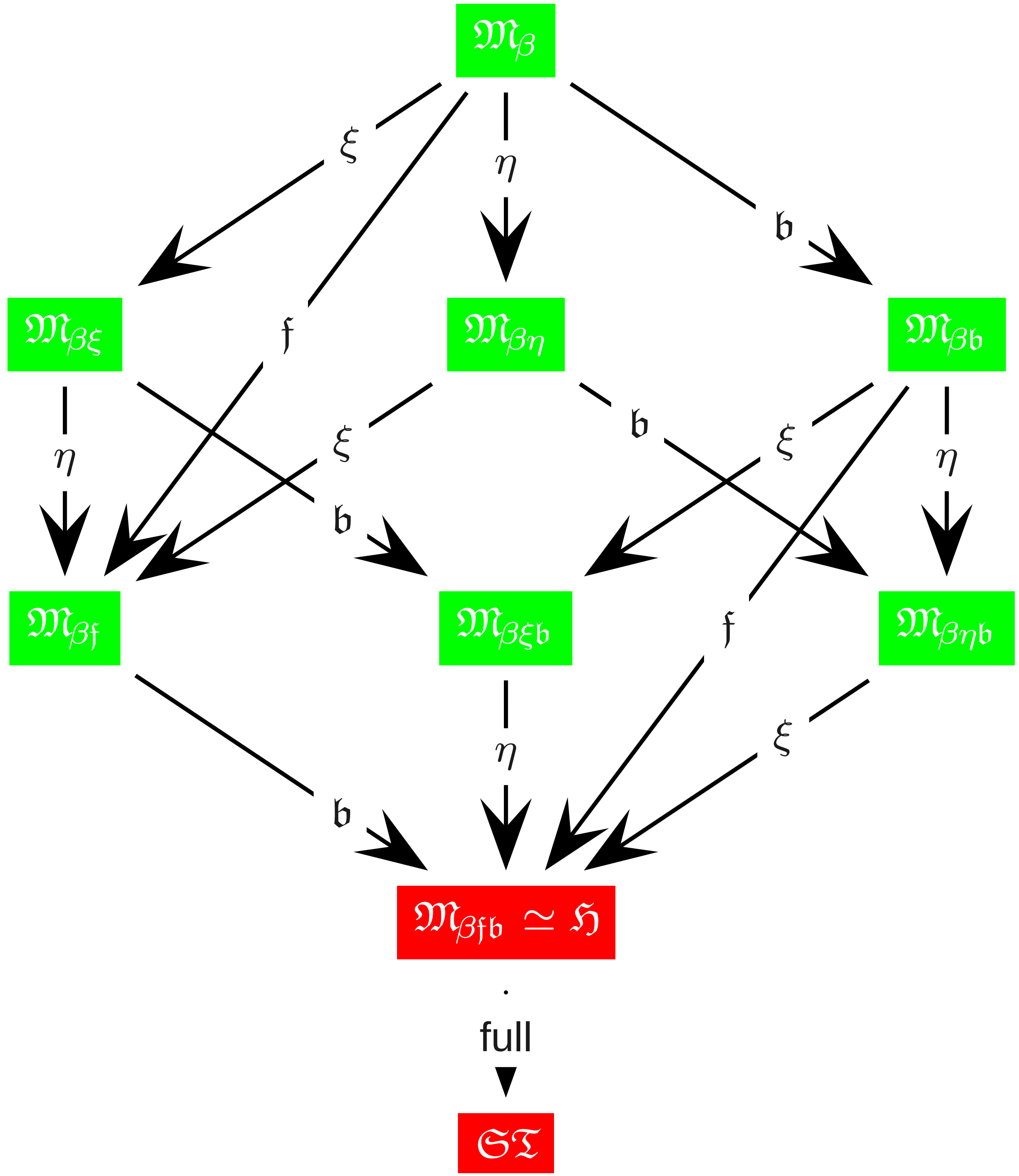
HOL-Problems: Set Comprehension



Set comprehension

- $\exists N_{oo} \forall P_o . \, NP \Leftrightarrow \neg P$
  - if $\neg \in \mathcal{S}$ or $\{\bot, \Rightarrow\} \subseteq \mathcal{S}$ or $\{\bot, \Leftrightarrow\} \subseteq \mathcal{S}$
  - e.g.: $N_{oo} \longleftarrow \lambda X_o . \, \neg X$
    e.g.: $N_{oo} \longleftarrow \lambda X_o . \, X \Rightarrow \bot$

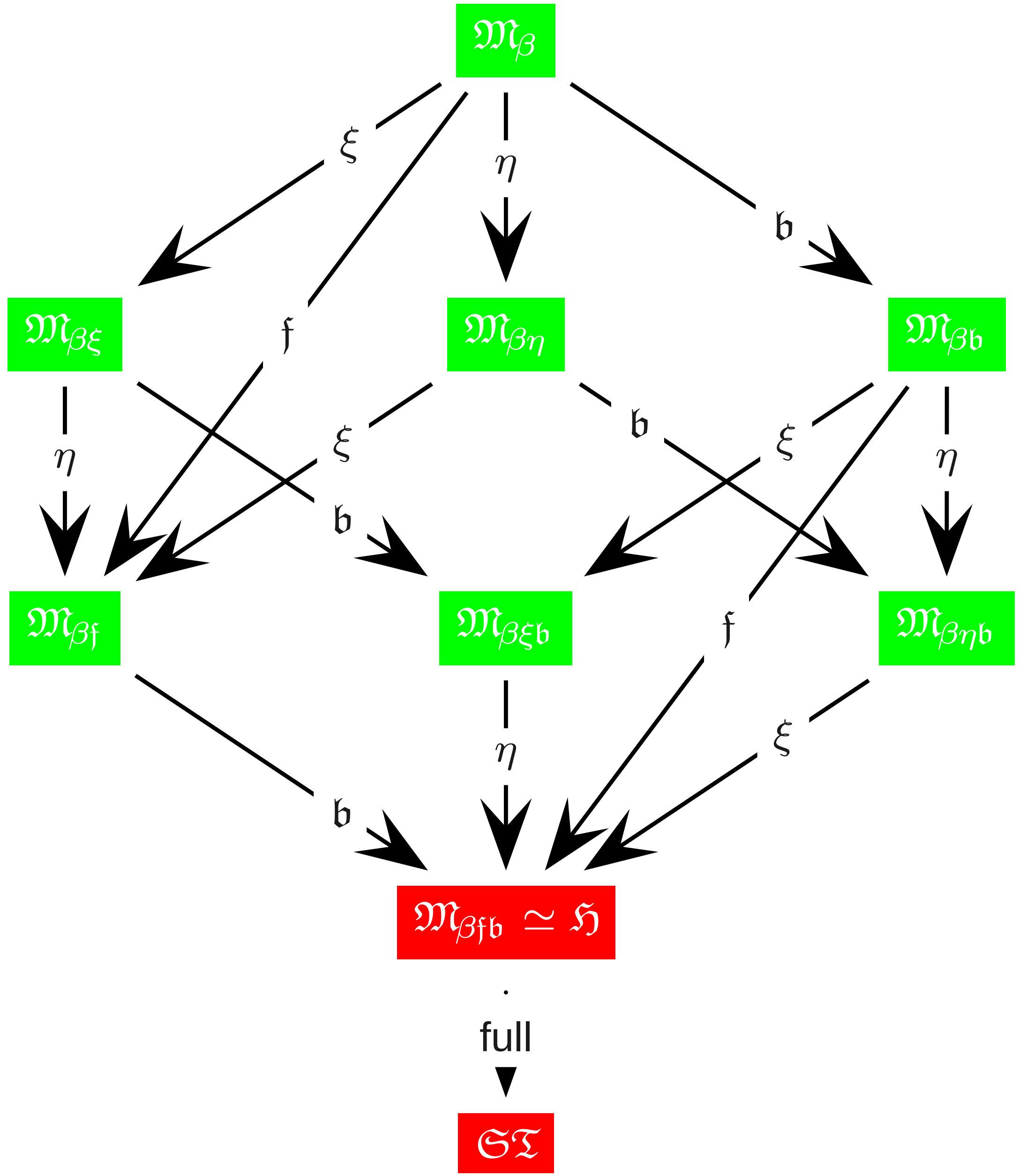
HOL-Problems: Set Comprehension



Set comprehension

- $\exists N_{oo} \forall P_o . \, NP \Leftrightarrow \neg P$
  - if $\neg \notin \mathcal{S}$ and $\{\bot, \Rightarrow\} \not\subseteq \mathcal{S}$ or $\{\bot, \Leftrightarrow\} \not\subseteq \mathcal{S}$

HOL-Problems: Set Comprehension



Set comprehension

- $\exists N_{oo} \forall P_o. NP \Leftrightarrow \neg P$
  - if $\neg \notin \mathcal{S}$ and $\{\bot, \Rightarrow\} \not\subseteq \mathcal{S}$ or $\{\bot, \Leftrightarrow\} \not\subseteq \mathcal{S}$

Other examples from [Brown-PhD-04]

- Surjective Cantor Theorem
- Injective Cantor Theorem

Conclusion

- Presented simple examples

Conclusion

- Presented simple examples
  - highlight some semantical or technical point

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality
  - mediocre and challenge ones to built up a HOL TPTP

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality
  - mediocre and challenge ones to built up a HOL TPTP
- Outlook

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality
  - mediocre and challenge ones to built up a HOL TPTP
- Outlook
  - Refine model classes for: description, choice, etc.

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality
  - mediocre and challenge ones to built up a HOL TPTP
- Outlook
  - Refine model classes for: description, choice, etc.
  - Build powerful HOL ATPs (see e.g. [LPAR-04])

Conclusion

- Presented simple examples
  - highlight some semantical or technical point
  - easy to understand / encode
  - relevant for automated and interactive TP
  - shall precede formal soundness / completeness analysis
- Further examples needed
  - to illustrate properties of different forms of equality
  - mediocre and challenge ones to built up a HOL TPTP
- Outlook
  - Refine model classes for: description, choice, etc.
  - Build powerful HOL ATPs (see e.g. [LPAR-04])
  - Integrate them with proof assistants

Thank You

[LPAR-04] HOL versus FOL

SET171+3	$\forall X_{o\alpha}, Y_{o\alpha}, Z_{o\alpha}. X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$
SET611+3	$\forall X_{o\alpha}, Y_{o\alpha}. (X \cap Y = \emptyset) \Leftrightarrow (X \setminus Y = X)$
SET624+3	$\forall X_{o\alpha}, Y_{o\alpha}, Z_{o\alpha}. \texttt{Meets}(X, Y \cap Z) \Leftrightarrow \texttt{Meets}(X, Y) \vee \texttt{Meets}(X, Z)$
SET646+3	$\forall x_\alpha, y_\beta. \texttt{Subrel}(\texttt{Pair}(x, y), (\lambda u_\alpha.\top) \times (\lambda v_\beta.\top))$
SET670+3	$\forall Z_{o\alpha}, R_{o\beta\alpha}, X_{o\alpha}, Y_{o\beta}. \texttt{IsRelOn}(R, X, Y) \Rightarrow \texttt{IsRelOn}(\texttt{RestrictRDom}(R, Z), Z, Y)$

$\_ \in \_$	:=	$\lambda x_\alpha, A_{o\alpha}.[Ax]$
$\emptyset$	:=	$[\lambda x_\alpha.\bot]$
$\_ \cap \_$	:=	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha. x \in A \wedge x \in B]$
$\_ \cup \_$	:=	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha. x \in A \vee x \in B]$
$\_ \setminus \_$	:=	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha. x \in A \vee x \notin B]$
$\texttt{Meets}(\_, \_)$	:=	$\lambda A_{o\alpha}, B_{o\alpha}.[\exists x_\alpha. x \in A \wedge x \in B]$
$\texttt{Pair}(\_, \_)$	:=	$\lambda x_\alpha, y_\beta.[\lambda u_\alpha, v_\beta. u = x \wedge v = y]$
$\_ \times \_$	:=	$\lambda A_{o\alpha}, B_{o\beta}.[\lambda u_\alpha, v_\beta. u \in A \wedge v \in B]$
$\texttt{Subrel}(\_, \_)$	:=	$\lambda R_{o\beta\alpha}, Q_{o\beta\alpha}.[\forall x_\alpha, y_\beta. Rxy \Rightarrow Qxy]$
$\texttt{IsRelOn}(\_, \_, \_)$	:=	$\lambda R_{o\beta\alpha}, A_{o\alpha}, B_{o\beta}.[\forall x_\alpha, y_\beta. Rxy \Rightarrow x \in A \wedge y \in B]$
$\texttt{RestrictRDom}(\_, \_)$	:=	$\lambda R_{o\beta\alpha}, A_{o\alpha}.[\lambda x_\alpha, y_\beta. x \in A \wedge Rxy]$