Towards Logic of Combinatory Logic

Simona Kašterović[1] Silvia Ghilezan[1, 2]

[1]Faculty of Technical Sciences, University of Novi Sad
[2]Mathematical Institute SASA, Belgrade, Serbia

Logic and Applications 2021
September 20-24, 2021

1. Syntax
2. Axiomatization
3. Semantics
4. Results: Soundness and Completeness
5. Related work and future work

Contents

1. Syntax
2. Axiomatization
3. Semantics
4. Results: Soundness and Completeness
5. Related work and future work

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

$$\textbf{Terms:}\ \ M, N ::= x \mid \mathsf{S} \mid \mathsf{K} \mid \mathsf{I} \mid MN$$

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

$$\textbf{Terms:}\ \ M, N ::= x \mid \mathsf{S} \mid \mathsf{K} \mid \mathsf{I} \mid MN$$

$$\textbf{Types:}\ \ \sigma, \tau := a \mid \sigma \to \tau$$

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

Terms: $M, N ::= x \mid \mathsf{S} \mid \mathsf{K} \mid \mathsf{I} \mid MN$

Types: $\sigma, \tau := a \mid \sigma \to \tau$

Type assignment statement

$$M : \sigma$$

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

$$\textbf{Terms:}\ M, N ::= x \mid \mathsf{S} \mid \mathsf{K} \mid \mathsf{I} \mid MN$$

$$\textbf{Types:}\ \sigma, \tau := a \mid \sigma \to \tau$$

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma} \qquad \frac{}{\Gamma \vdash \mathsf{S} : (\sigma \to (\rho \to \tau)) \to (\sigma \to \rho) \to (\sigma \to \tau)}$$

$$\frac{}{\Gamma \vdash \mathsf{K} : \sigma \to (\tau \to \sigma)} \qquad \frac{}{\Gamma \vdash \mathsf{I} : \sigma \to \sigma}$$

$$\frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

Syntax

- Logic of combinatory logic (LCL) - classical propositional logic over simply typed combinatory logic
- Syntax of simply typed combinatory logic

$$\textbf{Terms: } M, N ::= x \mid \mathsf{S} \mid \mathsf{K} \mid \mathsf{I} \mid MN$$

$$\textbf{Types: } \sigma, \tau := a \mid \sigma \to \tau$$

Type assignment statement

$$M : \sigma$$

Syntax of LCL

$$\alpha, \beta := M : \sigma \mid \neg\alpha \mid \alpha \wedge \beta$$

Equational theory $\mathcal{EQ}$

Equational theory $\mathcal{EQ}$

$$M = M \quad (id) \qquad \mathsf{S}MNL = (ML)(NL) \quad (\mathsf{S})$$

$$\mathsf{K}MN = M \quad (\mathsf{K}) \qquad \mathsf{I}M = M \quad (\mathsf{I})$$

$$\frac{M = N}{N = M}\,(\text{sym}) \qquad \frac{M = N \qquad N = L}{M = L}\,(\text{trans})$$

$$\frac{M = N}{MP = NP}\,(\text{app-l}) \qquad \frac{M = N}{PM = PN}\,(\text{app-r})$$

Equational theory $\mathcal{EQ}$

Equational theory $\mathcal{EQ}$

$$M = M \quad (id) \qquad SMNL = (ML)(NL) \quad (S)$$

$$\mathsf{K}MN = M \quad (K) \qquad \mathsf{I}M = M \quad (I)$$

$$\frac{M = N}{N = M}\,(\text{sym}) \qquad \frac{M = N \qquad N = L}{M = L}\,(\text{trans})$$

$$\frac{M = N}{MP = NP}\,(\text{app-l}) \qquad \frac{M = N}{PM = PN}\,(\text{app-r})$$

Rule: $$\frac{Mx = Nx \qquad x \notin FV(M) \cup FV(N)}{M = N}\,(\text{ext})$$

Equational theory $\mathcal{EQ}$

Equational theory $\mathcal{EQ}$

$$M = M \quad (id) \qquad SMNL = (ML)(NL) \quad (S)$$

$$\mathsf{K}MN = M \quad (K) \qquad \mathsf{I}M = M \quad (I)$$

$$\frac{M = N}{N = M}\,(\mathrm{sym}) \qquad \frac{M = N \qquad N = L}{M = L}\,(\mathrm{trans})$$

$$\frac{M = N}{MP = NP}\,(\mathrm{app\text{-}l}) \qquad \frac{M = N}{PM = PN}\,(\mathrm{app\text{-}r})$$

Rule: $$\frac{Mx = Nx \qquad x \notin FV(M) \cup FV(N)}{M = N}\,(\mathrm{ext})$$

- Equational theory $\mathcal{EQ}$ + rule (ext) = Equational theory $\mathcal{EQ}^{\eta}$.

Contents

1. Syntax
2. Axiomatization
3. Semantics
4. Results: Soundness and Completeness
5. Related work and future work

Axiomatization

- Combination of the axiomatic system for classical propositional logic and type assignment system for simply typed combinatory logic

Axiomatization

- Combination of the axiomatic system for classical propositional logic and type assignment system for simply typed combinatory logic
- Eight axiom schemes:

(Ax 1) $\mathsf{S} : (\sigma \to (\tau \to \rho)) \to ((\sigma \to \tau) \to (\sigma \to \rho))$
(Ax 2) $\mathsf{K} : \sigma \to (\tau \to \sigma)$
(Ax 3) $\mathsf{I} : \sigma \to \sigma$
(Ax 4) $(M : \sigma \to \tau) \Rightarrow ((N : \sigma) \Rightarrow (MN : \tau))$
(Ax 5) $M : \sigma \Rightarrow N : \sigma$, if $M = N$ is provable in $\mathcal{EQ}^\eta$
(Ax 6) $\alpha \Rightarrow (\beta \Rightarrow \alpha)$
(Ax 7) $(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$
(Ax 8) $(\neg\neg\alpha \Rightarrow \neg\beta) \Rightarrow ((\neg\neg\alpha \Rightarrow \beta) \Rightarrow \neg\alpha)$

Axiomatization

- Combination of the axiomatic system for classical propositional logic and type assignment system for simply typed combinatory logic
- Eight axiom schemes:

(Ax 1)	$\mathsf{S} : (\sigma \to (\tau \to \rho)) \to ((\sigma \to \tau) \to (\sigma \to \rho))$
(Ax 2)	$\mathsf{K} : \sigma \to (\tau \to \sigma)$
(Ax 3)	$\mathsf{I} : \sigma \to \sigma$
(Ax 4)	$(M : \sigma \to \tau) \Rightarrow ((N : \sigma) \Rightarrow (MN : \tau))$
(Ax 5)	$M : \sigma \Rightarrow N : \sigma$, if $M = N$ is provable in $\mathcal{EQ}^\eta$
(Ax 6)	$\alpha \Rightarrow (\beta \Rightarrow \alpha)$
(Ax 7)	$(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$
(Ax 8)	$(\neg\neg\alpha \Rightarrow \neg\beta) \Rightarrow ((\neg\neg\alpha \Rightarrow \beta) \Rightarrow \neg\alpha)$

- Inference rule:

$$\frac{\alpha \Rightarrow \beta \qquad \alpha}{\beta}\ (\mathrm{MP})$$

Contents

1. Syntax
2. Axiomatization
3. Semantics
4. Results: Soundness and Completeness
5. Related work and future work

Semantics

Applicative structure for LCL

Semantics

Applicative structure for *LCL*

$$\mathcal{M} = \langle D, \{A^{\sigma}\}_{\sigma}, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

Semantics

Applicative structure for LCL

$$\mathcal{M} = \langle \mathsf{D}, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

D is a non-empty set, called *domain*

Semantics

Applicative structure for LCL

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

$\{A^\sigma\}_\sigma$ is a family of sets indexed by σ such that $A^\sigma \subseteq D$ for all σ, $\{A^\sigma\}_\sigma$ is a short for $\{A^\sigma\}_{\sigma \in \mathsf{Types}}$

Semantics

Applicative structure for LCL

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

$\cdot$ is a binary operation on D

$$\cdot \; : \; D \times D \to D$$

It is extensional: for $d_1, d_2 \in D$,

if $(\forall e \in D)(d_1 \cdot e = d_2 \cdot e)$, then $d_1 = d_2$.

It holds that $\cdot \; : \; A^{\sigma \to \tau} \times A^\sigma \to A^\tau$

Semantics

Applicative structure for *LCL*

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i}\rangle$$

s is an element of the domain D, such that

- for every $\sigma, \tau, \rho \in$ Types,

$$\mathsf{s} \in A^{(\sigma\to(\tau\to\rho))\to((\sigma\to\tau)\to(\sigma\to\rho))}$$

- for every $d, e, f \in D$,

$$((\mathsf{s} \cdot d) \cdot e) \cdot f = (d \cdot f) \cdot (e \cdot f)$$

Applicative structure for LCL

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i}\rangle$$

k is an element of the domain D, such that

- for every $\sigma, \tau \in$ Types,

$$\mathsf{k} \in A^{\sigma\to(\tau\to\sigma)}$$

- for every $d, e \in D$,

$$(\mathsf{k} \cdot d) \cdot e = d$$

Applicative structure for *LCL*

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

i is an element of the domain D, such that

- for every $\sigma \in$ Types,

$$\mathsf{i} \in A^{\sigma \to \sigma}$$

- for every $d \in D$,

$$\mathsf{i} \cdot d = d$$

Semantics

Applicative structure for *LCL*

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

Semantics

Applicative structure for LCL

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

An environment ρ for $\mathcal{M}$ is a map from the set of term variables to the domain of the applicative structure $\mathcal{M}$, $\rho : V \to D$.

Semantics

Applicative structure for LCL

$$\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathsf{s}, \mathsf{k}, \mathsf{i} \rangle$$

An environment ρ for $\mathcal{M}$ is a map from the set of term variables to the domain of the applicative structure $\mathcal{M}$, $\rho : V \to D$.

A LCL-model is a tuple $\mathcal{M}_\rho = \langle \mathcal{M}, \rho \rangle$, where $\mathcal{M}$ is an applicative structure and ρ is an environment for $\mathcal{M}$.

Kašterović, S., Ghilezan, S., *Kripke semantics and completeness for full simply typed lambda calculus,* Journal of Logic and Computation, Volume 30, issue 8 (2020).

Mitchell, J. C., and E. Moggi, *Kripke-style models for typed lambda calculus*, Annals of Pure and Applied Logic, vol. 51, pp. 99–124, 1991.

The interpretation of a term

- $[\![x]\!]_\rho = \rho(x)$;
- $[\![\mathsf{S}]\!]_\rho = \mathsf{s}$;
- $[\![\mathsf{K}]\!]_\rho = \mathsf{k}$;
- $[\![\mathsf{I}]\!]_\rho = \mathsf{i}$;
- $[\![MN]\!]_\rho = [\![M]\!]_\rho \cdot [\![N]\!]_\rho$.

Satisfiability

- $\mathcal{M}_\rho \models M : \sigma$ if and only if $[\![M]\!]_\rho \in A^\sigma$;
- $\mathcal{M}_\rho \models \alpha \wedge \beta$ if and only if $\mathcal{M}_\rho \models \alpha$ and $\mathcal{M}_\rho \models \beta$;
- $\mathcal{M}_\rho \models \neg\alpha$ if and only if it is not true that $\mathcal{M}_\rho \models \alpha$.

Contents

1 Syntax

2 Axiomatization

3 Semantics

4 Results: Soundness and Completeness

5 Related work and future work

Soundness and completeness of equational theory $\mathcal{EQ}^\eta$

Soundness of $\mathcal{EQ}^\eta$

If $M = N$ in provable in $\mathcal{EQ}^\eta$, then $[\![M]\!]_\rho = [\![N]\!]_\rho$ for any LCL-model $\mathcal{M}_\rho = \langle \mathcal{M}, \rho \rangle$.

Soundness and completeness of equational theory $\mathcal{EQ}^\eta$

Soundness of $\mathcal{EQ}^\eta$

If $M = N$ in provable in $\mathcal{EQ}^\eta$, then $[\![M]\!]_\rho = [\![N]\!]_\rho$ for any LCL-model $\mathcal{M}_\rho = \langle \mathcal{M}, \rho \rangle$.

Completeness of $\mathcal{EQ}^\eta$

If $[\![M]\!]_\rho = [\![N]\!]_\rho$ in every LCL-model, then $M = N$ is provable in $\mathcal{EQ}^\eta$.

Soundness and completeness of the axiomatiation for *LCL*

Soundness and completeness of the axiomatiation for *LCL*

Soundness of Ax

If $T \vdash \alpha$, then $T \models \alpha$.

Soundness and completeness of the axiomatiation for *LCL*

Soundness of Ax

If $T \vdash \alpha$, then $T \models \alpha$.

Completeness of Ax

If $T \models \alpha$, then $T \vdash \alpha$.

Contents

1 Syntax

2 Axiomatization

3 Semantics

4 Results: Soundness and Completeness

5 Related work and future work

Related work

S. Feferman, “Constructive theories of functions and classes,” in Proceedings Logic Colloquium ’78, v. D. Boffa, D. and K. McAloon, Eds., vol. 97. Amsterdam: North-Holland, 1979, pp. 159–224, mons, Aug. 24–Sept. 1., 1978.

M. Beeson, “Lambda logic,” in Automated Reasoning - Second International Joint Conference, IJCAR 2004, Cork, Ireland, July 4-8, 2004, Proceedings, ser. Lecture Notes in Computer Science, D. A. Basin and M. Rusinowitch, Eds., vol. 3097. Springer, 2004, pp. 460–474. [Online]. Available: https://doi.org/10.1007/978-3-540-25984-8 34

H. Barendregt, M. W. Bunder, and W. Dekkers, “Systems of illative combinatory logic complete for first-order propositional and predicate calculus,” J. Symb. Log., vol. 58, no. 3, pp. 769–788, 1993. [Online]. Available: https://doi.org/10.2307/2275096

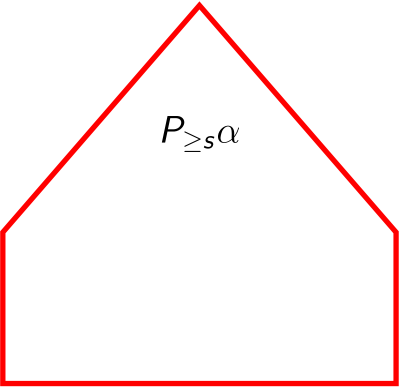
W. Dekkers, M. W. Bunder, and H. Barendregt, “Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus,” Arch. Math. Log., vol. 37, no. 5-6, pp. 327–341, 1998. [Online]. Available: https://doi.org/10.1007/s001530050102

Motivation and future work

LPP_2

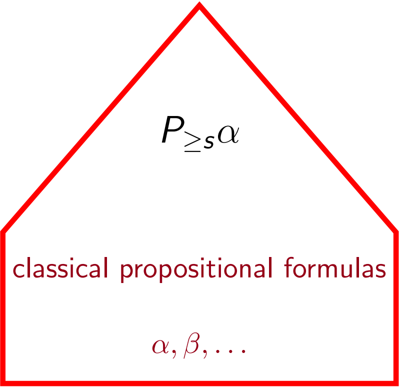
Motivation and future work

LPP_2



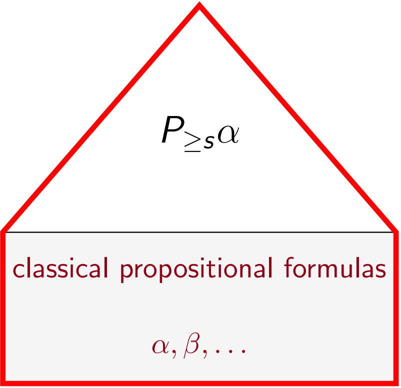
Motivation and future work

LPP_2



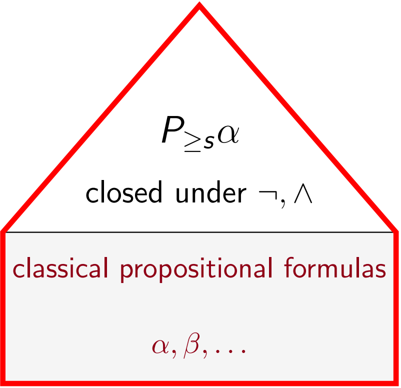
Motivation and future work

LPP_2

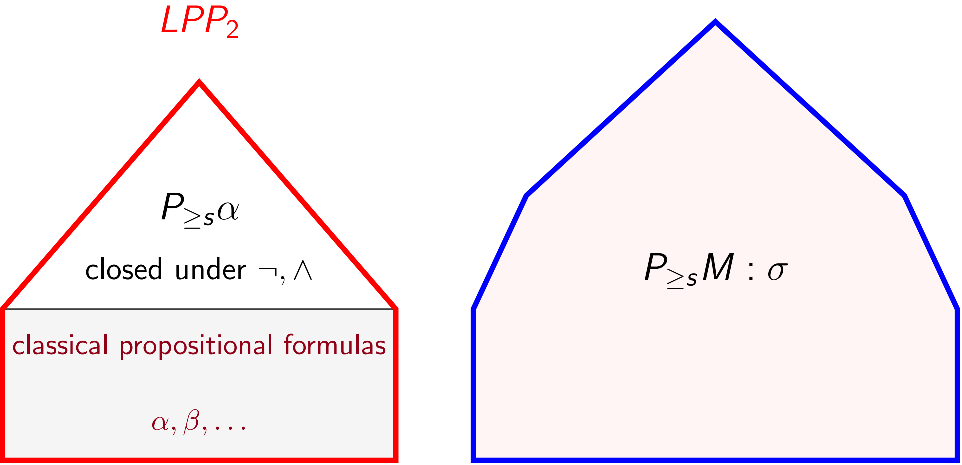


Motivation and future work

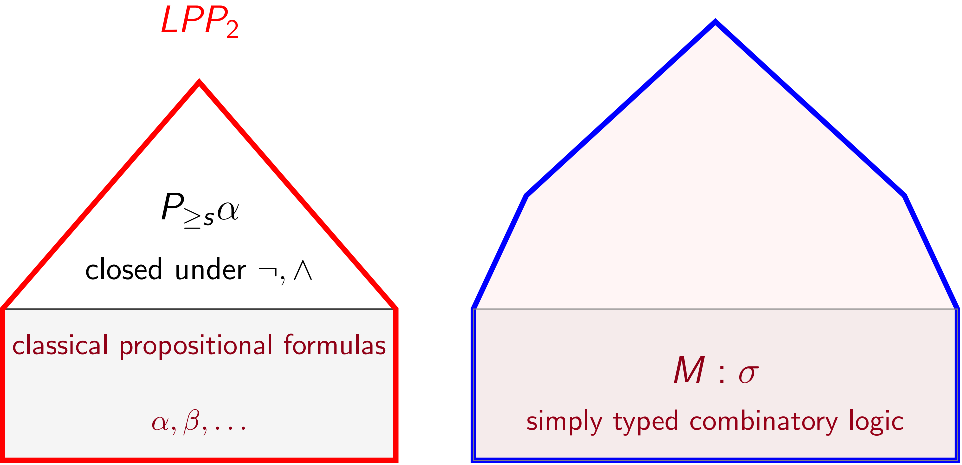
LPP_2



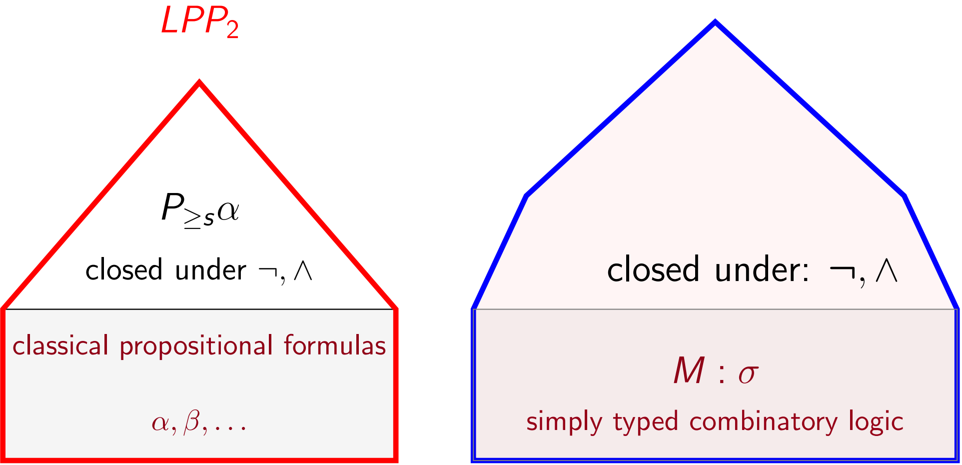
Motivation and future work



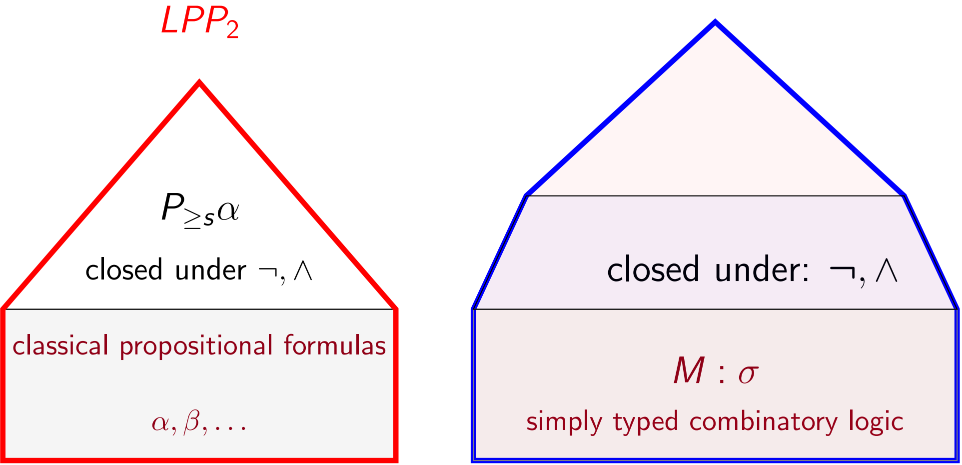
Motivation and future work



Motivation and future work

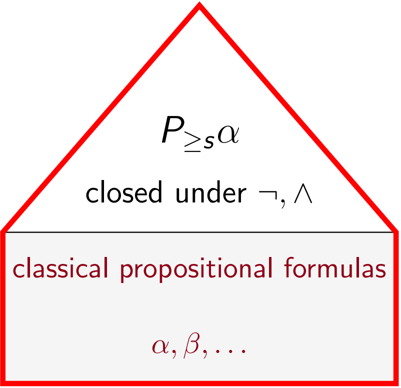


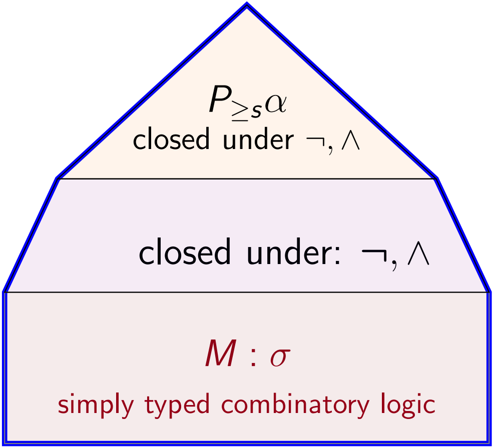
Motivation and future work



Motivation and future work

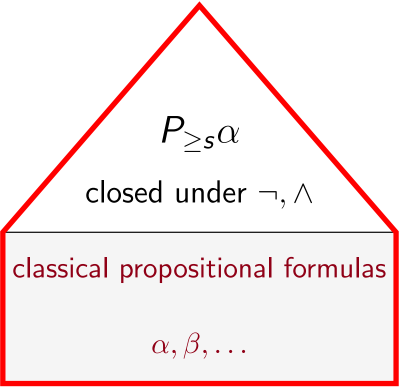
LPP_2

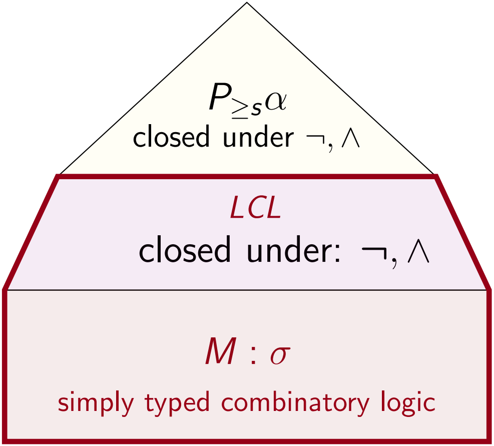




Motivation and future work

LPP_2





Logic of combinatory logic

Towards Logic of Combinatory Logic

Simona Kašterović[1] Silvia Ghilezan[1, 2]

[1]Faculty of Technical Sciences, University of Novi Sad
[2]Mathematical Institute SASA, Belgrade, Serbia

Logic and Applications 2021
September 20-24, 2021