# Towards Probabilistic Reasoning about Typed Combinatory Terms

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

[1] Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
simona.k@uns.ac.rs, gsilvia@uns.ac.rs
[2] Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia

The interest in probabilistic programming has increased over the past decades due to the role which reasoning with uncertainty has in computer science and artificial intelligence. The developments in computer science and artificial intelligence urge for formalizing uncertain reasoning. Combinatory logic found its application in computer science as a model of computation. In order to formalize reasoning with uncertainty about programs, we propose probabilistic model for reasoning about typed combinatory terms.

We present results of [8] and ongoing work that emerged from [4, 3, 8]. In [8] we introduced *Logic of Combinatory Logic* ($LCL$), a classical propositional logic for reasoning about simply typed combinatory logic. $LCL$ is obtained by defining the classical propositional logic over simply typed combinatory logic. The language of $LCL$ is defined by the following grammar:

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

where $M : \sigma$ is type assignment statement typable from some basis $\Gamma$ in simply typed combinatory logic, $M$ is a combinatory term and $\sigma$ is a simple type.

The axiomatic system of $LCL$ is obtained from the axiomatic system for classical propositional logic and the type assignment system for simply typed combinatory logic (Figure 1). Instances of axiom schemes can only be built up from formulas of the language, hence, $M : \sigma$ can be a subformula of some instance of an axiom scheme only if $M : \sigma$ is type assignment statement typable from some basis $\Gamma$ in simply typed combinatory logic. Besides the axiom schemes of classical propositional logic and the inference rule Modus Ponens, the proposed axiomatic system contains three non-logical axiom schemes for typing primitive combinators $\mathsf{S}, \mathsf{K}, \mathsf{I}$ and two axiom schemes that correspond to the typing rules of simply typed combinatory logic. The simple type system developed in [2] does not have an equality rule which would correspond to (Ax 5). However, in [5] Hindley has added this rule in order to obtain completeness of the type assignment system for lambda calculus. In the notation $=_{w,\eta}$, $w$ denotes that it is a transitive, reflexive and symmetric closure of the one-step reduction, and $\eta$ denotes that it is an extensional relation.

We proposed semantics for $LCL$, inspired by Kripke-style semantics for lambda calculus with types introduced in [9, 7]. The semantics for $LCL$ is based on an extensional applicative structure containing special elements that correspond to primitive combinators. The proposed semantics are not a Kripke-style semantics as the ones presented in [9, 7], still the definition of the applicative structure is inspired by the definition of the applicative structure presented in [9, 7]. An applicative structure for $LCL$ is a tuple $\mathcal{M} = \langle D, \{A^\sigma\}_\sigma, \cdot, \mathbf{s}, \mathbf{k}, \mathbf{i}\rangle$ where $D$ is a non-empty set, called domain; $\{A^\sigma\}_\sigma$ is a family of subsets of the domain $D$; $\cdot$ is an extensional binary operation on $D$ such that $\cdot : A^{\sigma\to\tau} \times A^\sigma \to A^\tau$ for every $\sigma, \tau \in \mathsf{Types}$; $\mathbf{s}, \mathbf{k}, \mathbf{i}$ are special elements of the domain $D$ that correspond to the primitive combinators, that is $\mathbf{s} \in A^{(\sigma\to\tau\to\rho)\to(\sigma\to\tau)\to(\sigma\to\rho)}$ and $((\mathbf{s}\cdot a)\cdot b)\cdot c = (a\cdot c)\cdot(b\cdot c)$, and similar for the elements $\mathbf{k}, \mathbf{i}$.

We have proved that the given axiomatic system is sound and complete with respect to the proposed semantics.

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 $M =_{w,\eta} N$ |
| (Ax 6) | $\alpha \Rightarrow (\beta \Rightarrow \alpha)$ |
| (Ax 7) | $(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$ |
| (Ax 8) | $(\neg\alpha \Rightarrow \neg\beta) \Rightarrow ((\neg\alpha \Rightarrow \beta) \Rightarrow \alpha)$ |

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

Figure 1: Axiom schemes and inference rule for $LCL$

Our goal is to develop a formal model for probabilistic reasoning about simply typed combinatory terms. In [4, 3], formal models are introduced for probabilistic reasoning about simply typed lambda terms and lambda terms with intersection types, respectively. These models are based on the well-known semantics for typed lambda calculus, namely, term models for simply typed lambda calculus ([5]) and filter models for lambda calculus with intersection types ([1]). However, these models are not well-suited for propositional reasoning about typed terms, thus we develop a model for probabilistic reasoning about typed combinatory terms based on the Logic of Combinatory Logic ([8]).

We propose a probabilistic system for simply typed combinatory terms, $PCL$ which is a probabilistic logic ([6, 10]) over $LCL$. We extend $LCL$ with the probabilistic operator $P_{\geq s}$, and obtain a system expressive enough to write formulas of the form $P_{\geq s}\alpha$ which has a meaning "probability that $\alpha$ is true is greater than or equal to $s$". The language of $PCL$ consists of two sets of formulas: *basic formulas* and *probabilistic formulas*. Basic formulas are formulas of $LCL$. Probabilistic formulas are formulas generated by the following grammar

$$\phi := P_{\geq s}\alpha \mid \neg\phi \mid \phi \wedge \phi,$$

where $\alpha$ is an $LCL$-formula and $s \in [0, 1] \cap \mathbb{Q}$.

We propose Kripke-style semantics, where $LCL$-models serve as possible worlds. To interpret probabilities we equip the set of possible worlds with a probability measure. We will give an infinitary axiomatic system, obtained from the axiomatic system of $LCL$ along with the axiomatic system for probability logic.

In order to obtain soundness and completeness results for $PCL$ we had to prove soundness and completeness results on the level of basic formulas. Kripke-style semantics for the proposed logic is built from models for $LCL$, by defining probability measure over the set of $LCL$-models. Thus, the completeness result for $LCL$ ([8]) will play a key role in proving completeness of the given axiomatization with respect to the proposed semantics.

**Acknowledgements** We wish to thank the anonymous reviewers for their valuable suggestions.

# References

[1] Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *J. Symb. Log.*, 48(4):931–940, 1983.

[2] Haskell Brooks Curry and Robert M. Feys. *Combinatory Logic Vol. 1.* Amsterdam, Netherlands: North-Holland Publishing Company, 1958.

[3] Silvia Ghilezan, Jelena Ivetić, Simona Kašterovć, Zoran Ognjanović, and Nenad Savić. Towards probabilistic reasoning in type theory - the intersection type case. In Andreas Herzig and Juha Kontinen, editors, *Foundations of Information and Knowledge Systems - 11th International Symposium, FoIKS 2020, Dortmund, Germany, February 17-21, 2020, Proceedings*, volume 12012 of *Lecture Notes in Computer Science*, pages 122–139. Springer, 2020.

[4] Silvia Ghilezan, Jelena Ivetić, Simona Kašterović, Zoran Ognjanović, and Nenad Savić. Probabilistic reasoning about simply typed lambda terms. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8-11, 2018, Proceedings*, volume 10703 of *Lecture Notes in Computer Science*, pages 170–189. Springer, 2018.

[5] J. Roger Hindley. The completeness theorem for typing lambda-terms. *Theor. Comput. Sci.*, 22:1–17, 1983.

[6] Nebojša Ikodinović, Zoran Ognjanović, Aleksandar Perović, and Miodrag Rašković. Hierarchies of probabilistic logics. *Int. J. Approx. Reason.*, 55(9):1830–1842, 2014.

[7] Simona Kašterović and Silvia Ghilezan. Kripke-style semantics and completeness for full simply typed lambda calculus. *J. Log. Comput.*, 30(8):1567–1608, 2020.

[8] Simona Kašterović and Silvia Ghilezan. Logic of combinatory logic, February 2021. Submitted for publication.

[9] John C. Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Ann. Pure Appl. Log.*, 51(1-2):99–124, 1991.

[10] Zoran Ognjanović, Miodrag Rašković, and Zoran Marković. *Probability Logics - Probability-Based Formalization of Uncertain Reasoning.* Springer, 2016.