IPSJ Transactions on Programming Vol.14 No.2 32 (May 2021)

Presentation Abstract

On a Generalization of the Curry Correspondence

Yosuke Fukuda[1,2,a)]

Presented: October 30, 2020

Combinatory logic discovered by Schönfinkel (1924) and Curry (1930) is a formal system of computation that does not explicitly make use of bound variables. Unlike λ-calculus, which uses bound variables as primitive concepts in its function notation, combinatory logic realizes various computations by means of "combinators" (e.g., using SKI combinators), and this framework was often used as an implementation technique for functional programming languages since the late 1970s. An interesting aspect of it is its relation to proof theory. As Curry observed (1958), a typed-version of combinatory logic corresponds to a Hilbert-style deductive system (i.e., a kind of formal system of proof theory) in the sense that there is a one-to-one correspondence between well-typed programs and well-formed proofs. Thanks to this Curry correspondence, we can investigate the proof-theoretical meaning of computations and the computational meaning of proofs. However, although the correspondence is famous as the origin of the Curry–Howard correspondence, it is often mentioned in the context of the implicational fragment of intuitionistic propositional logic. Its extensions to other logical fragments are not widely known. In this presentation, we review related work that mentions some extensions of the Curry correspondence, and then discuss its generalization. If time permits, we also discuss a connection to the logical abstract machine developed by Ohori (1999).

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

[1] STAIR Lab, Tsudanuma Campus, Chiba Institute of Technology, Narashino, Chiba 275–0016, Japan

[2] Graduate School of Informatics, Kyoto University, Kyoto 606–8501, Japan

[a)] yfukuda@fos.kuis.kyoto-u.ac.jp