Recursive Programs in the Abstract

Masahito Hasegawa
Research Institute for Mathematical Sciences, Kyoto University
PRESTO, Japan Science and Technology Agency
hassei@kurims.kyoto-u.ac.jp

Semantics of recursive programs has been extensively studied for more than 30 years, and now there exist several well-established theories, most notably the fixed-point semantics and domain theory developped by Dana Scott and others. Is it then true that all the issues regarding recursive programs have now been settled in a satisfactory manner? Is it just a "classical" topic to which we can add nothing?

I think that it is not the case, in the sense that classical theories did not pay sufficient attention to how recursion is created, nor to how recursion interacts with other features of programming languages. I regard them as important and challenging questions, from both theoretical and practical perspectives. In fact, recent investigations on these issues have revealed quite rich — and abstract [9], I would say — mathematical structures behind recursive programs, which also have provided (or at least suggested) general — sometimes very concrete — principles for reasoning about recursion. I shall outline three such cases:

1. *Recursion from Cyclic Sharing* (in call-by-name languages) [3, 4, 5, 6]
   which explains recursion created from cyclic structures (like cyclic graph rewriting) using mathematics for cyclic structures (like knot theory) [10],

2. *Recursion and Control* (in call-by-value languages) [7, 8]
   which explains principles on recursion under the presence of side effects, especially control primitives like first-class continuations [2], and

3. *Recursion and Duality* (between call-by-name and call-by-value) [11, 12]
   which suggests a way to relate the seemingly isolated two cases explained above in terms of a mathematical duality [1, 13].

Through these stories, I shall try to advocate some views on the semantics-based approaches to programming languages which are now being carried out. In other words: "Programming Languages in the Abstract".

Part of this talk is based on joint work with Yoshihiko Kakutani.

References

[1] Filinski, A. (1989) Declarative continuations: an investigation of duality in programming language semantics. In *Proc. Category Theory and Computer Science*, Springer Lecture Notes in Comput. Sci. **389**, pp. 224–249.

[2] Filinski, A. (1994) Recursion from iteration. *Lisp and Symbolic Comput.* **7**(1): 11–38.

[3] †Hasegawa, M. (1997) Recursion from cyclic sharing: traced monoidal categories and models of cyclic lambda calculi. In *Proc. 3rd International Conference on Typed Lambda Calculi and Applications (TLCA'97)*, Springer Lecture Notes in Comput. Sci. **1210**, pp. 196–213.

[4] Hasegawa, M. (1997) *Models of Sharing Graphs: A Categorical Semantics of let and letrec*. PhD thesis, University of Edinburgh, 1997 / Distinguished Dissertation Series, Springer-Verlag, 1999.

[5] 長谷川真人 (1999) 再帰的プログラムの意味論とトレース付きモノイダルカテゴリ. **コンピュータソフトウェア 16**(2): 62–66.

[6] †Hasegawa, M. (2002) The uniformity principle on traced monoidal categories. In *Proc. 9th International Conference on Category Theory and Computer Science (CTCS'02)*, Electronic Notes in Theoretical Computer Science **69**. A revised and expanded version will appear in *Publ. Res. Inst. Math. Sci.*

[7] †Hasegawa, M. and Kakutani, Y. (2001) Axioms for recursion in call-by-value (extended abstract). In *Proc. 4th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2001)*, Springer Lecture Notes in Comput. Sci. **2030**, pp. 246–260.

[8] †Hasegawa, M. and Kakutani, Y. (2002) Axioms for recursion in call-by-value. *Higher-Order and Symbolic Computation* **15**(2/3): 235–264.

[9] Hyland, J.M.E. (2002) Proof theory in the abstract. *Ann. Pure Appl. Logic* **114**: 43–78.

[10] Joyal, A., Street, R. and Verity, D. (1996) Traced monoidal categories. *Math. Proc. Cambridge Phil. Soc.* **119**(3): 447–468.

[11] Kakutani, Y. (2002) Duality between call-by-name recursion and call-by-value iteration. In *Proc. 11th Annual Conference of the European Association for Computer Science Logic (CSL'02)*, Springer Lecture Notes in Comput. Sci. **2471**, pp. 506–521.

[12] †Kakutani, Y. and Hasegawa, M. (2003) Parameterizations and fixed-point operators on control categories. In *Proc. 6th International Conference on Typed Lambda Calculi and Applications (TLCA'03)*, Springer Lecture Notes in Comput. Sci. **2701**, pp. 180–194.

[13] Selinger, P. (2001) Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Math. Structures Comput. Sci.* **11**(2): 207–260.

†: available from

`http://www.kurims.kyoto-u.ac.jp/~hassei/papers/`