Pola: A Language for PTIME Programming

Michael J. Burrell
Computer Science Department, University of Western Ontario,
London, Ontario N6A 5B7, Canada
Robin Cockett and Brian F. Redmond
Department of Computer Science, University of Calgary,
Calgary, Alberta T2N 1N4, Canada
robin@cpsc.ucalgary.ca

Pola is a functional style programming language—currently under development—whose type system guarantees that all its programs run in polynomial time. Indeed, as every polynomial time (PTIME) algorithm can be rendered in Pola, the language is complete for polynomial time programming. Furthermore, to someone familiar with functional programming, Pola enforces a style of programming which is—relative to what is being achieved—quite natural.

Pola was inspired by the realization that Bellantoni and Cook's system of safe recursion [1] can be viewed as the proof theory of a polarized logic [4]. As polarized logic was developed to model games, Pola has inherited some game theoretic terminology. In particular, rather than having "safe" and "normal" types, Pola has "player" and "opponent" types. The game theoretic view is that the opponent drives the iteration of computation while the player responds in constant time (and space) in the context he is given. This semantics has a particularly appealing and simple categorical presentation using a fibration whose fibers are affine categories together with a notion of "comprehended" inductive fixed points.

The development of *implicit* systems for PTIME programs has considerable history. In particular, Bellantoni and Cook's system of safe recursion is a simplification of a slightly earlier system of Leivant [8]. That system used tiered recursion and supported a general class of inductive data. Bellantoni and Cook, besides simplifying the system of tiers by safe recursion, also abandoned general inductive data in favor of modeling binary numbers: this limits their system as a basis for programming. Hofmann, in his habilitationsschrift [6], aware of all the above, developed a modal type system for PTIME programs which he modeled in a presheaf topos. In particular, Hofmann suggested that constant time affine computations could be used as the basis for stepping up to polynomial time. Pola's player world (in a given opponent context) is, as mentioned above, both affine and populated by constant time computations. Furthermore, in Pola these player worlds can additionally support arbitrary coinductive fixed points.

An important step in the development of Pola was to provide uniform recursion schemes both for inductive and coinductive data. Pola's recursion scheme derives from the circular proof systems of Luigi Santocanale [11], which also appeared earlier (with other schemes) in Varmo Vene's thesis [12]. This recursion scheme, as used in Pola, is interesting as it has some built-in higher-order content which allows one to avoid Colson's objection [5, 10] to first-order recursion schemes and to express recursive programs in a reasonably natural manner.

The original motivation behind Pola was driven by the investigation of implicit type systems for low complexity programs. However, the fact that there is a relatively simple categorical semantics for this system, suggests another important direction: perhaps, categorical techniques can be usefully employed to obtain a deeper structural understanding of PTIME?

References

[1] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity*, 2:97–110, 1992.

[2] V.-H. Caseiro. *Equations for Defining Poly-time Functions*. PhD thesis, University of Oslo, 1997.

[3] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, ed., *Proc. of 1964 International Congress for Logic, Methodology, and the Philosophy of Science*, pp. 24–30. North Holland, 1964.

[4] R. Cockett and R. Seely. Polarized category theory, modules and game semantics. *Theory and Appl. of Categ.*, 18:4–101, 2007.

[5] L. Colson. About primitive recursive algorithms. *Theor. Comput. Sci.*, 83(1):57–69, 1991.

[6] M. Hofmann. *Type Systems for Polynomial-Time Computation.* Habilitation thesis, University of Darmstadt, 1999.

[7] M. Hofmann. Linear types and non-size-increasing polynomial time computation. *Inform. and Comput.*, 183(1):57–85, 2003.

[8] D. Leivant. Stratified functional programs and computational complexity. In *Conf. Record of 20th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '93 (Charleston, SC, Jan. 1993)*, pp. 325–333. ACM Press, 1993.

[9] D. Leivant and J.-Y. Marion. Ramified recurrence and computational complexity II: substitution and polyspace. In L. Pacholski and J. Tiuryn, eds., *Selected Papers from 8th Int. Wksh. on Computer Science Logic, CSL '94 (Kazimierz, Sept. 1994)*, v. 933 of *Lect. Notes in Comput. Sci.*, pp. 486–500. Springer, 1995.

[10] Y. N. Moschovakis. On Colson's theorem. Invited talk at 2nd Panhellenic Logic Symp. (Delphi, July 1999). Available at `http://www.mathphys.ntua.gr/logic/symposium/articles/moschova.ps`.

[11] L. Santocanale. A calculus of circular proofs and its categorical semantics. In M. Nielsen and U. Engberg, eds., *Proc. of 5th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS 2002 (Grenoble, Apr. 2002)*, v. 2303 of *Lect. Notes in Comput. Sci.*, pp. 357–371. Springer, 2002.

[12] V. Vene. *Categorical Programming with Inductive and Coinductive Types.* PhD thesis, University of Tartu, 2000.