Computation & Deduction, Spring 1997
Frank Pfenning

Suggested Projects

The project should consist of a term paper and an implementation in Elf. The term paper should summarize and explain a piece of research in the theory of programming languages or logics. Typically, the informal development follows a given research paper or report. The Elf implementation should formalize the language, judgments, algorithms, and metatheory to the extent that this is feasible for a term project.

Each student should propose his or her own term project. Some topics are suggested below, but they should not be considered as canonical or exhaustive.

1. A calculus of explicit substitutions [ACCL91] [Abadi, Cardelli, Curien, & Levy 91]

2. A calculus of objects [AC94] [Abadi & Cardelli 94]

3. Coherence of a language with subtyping and implicit coercions [BTCGS91] [Breazu-Tannen, Coquand, Gunter, & Scedrov 91]

4. A lambda-calculus with intersection types [CDCV81, Rey91] [Coppo, Dezani-Ciancaglini, & Venneri 81] [Reynolds 91]

5. A logic for parametricity [PA93] [Plotkin & Abadi 93]

6. A lambda-calculus with names [Ode94] [Odersky 94]

7. First-order unification [Sny91] [Snyder & Gallier 88]

8. Higher-order logic programming [MNPS91] [Miller, Nadathur, Pfenning, & Scedrov 91]

9. Program analysis and optimization (each topic below is a possible separate project)

    (a) Binding-time analysis [Nielson & Nielson 92] [NN92], [Davies & Pfenning 96] [Davies 96] [Hatcliff 96]

    (b) Abstract interpretation [Cousot & Cousot 77] [CC77]

    (c) Strictness analysis [Hankin & Metayer 94] [HM94]

10. The Categorical Abstract Machine [CCM87] [Cousineau, Curien, & Mauny 87]
11. Closure conversion and lambda-lifting [Rey72] [Reynolds 72]
12. Extending Mini-ML (each topic below is a possible separate project) [MTH90]
    (a) with exceptions
    (b) with references
    (c) with general datatype definitions
    (d) overloading and typecase

    [Milner, Tofte, & Harper 90]
13. Mini-Forsythe [Rey88] [Reynolds 88]
14. Elementary notions of rewrite systems [Bac91] [Bachmair 91]
15. Modal logic and Kripke semantics [Basin, Matthews, Vigano 96]
16. Polymorphic recursion and semi-unification [Hen93] [Henglein 93, TOPLAS]
17. Lazy evaluation [Lau93] [Launchbury 93]
18. Pure functional programming and monads [Fil94]
19. Evaluation based on evaluation contexts [Felleisen]
20. Logical interpretations [Goedel, Friedman]
21. Linear logic [Girard 87, Pfenning 96]
22. Computational interpretations of classical logic [Ong 96, Ong & Stewart 97]
23. Pi-calculus [Milner]
24. Theorem proving
    (a) via tactics
    (b) via iterative deepening
    (c) other methods
    (d) induction
    (e) Presburger arithmetic

(f) decision procedures for fragments

25. Elementary category theory [Rydeheard & Burstall, Gehrke]
26. Type-directed compilation [Morrisett 95]
27. Shape types [Fradet & Metayer 97]
28. Relational parametricity and units of measure [Kennedy 97]
29. Type directed partial evaluation [Danvy 96]
30. Typed closure conversion [Minamide, Morrisett, Harper 96]
31. Call-by-value lambda-calculus [Plotkin 75, Sabry & Wadler 96]
32. Call-by-need lambda-calculus [Ariola et al 95]
33. Recursive types [Abadi & Fiore 96]
34. "Logical" logic program compilation [Pfenning 97]
35. LF in Elf
36. Elf in Elf

References

[AC94] Martín Abadi and Luca Cardelli. A theory of primitive objects. Draft paper available from gatekeeper.pa.dec.com, February 1994.

[ACCL91] Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, October 1991.

[Bac91] Leo Bachmair. *Canonical Equational Proofs.* Birkäuser, Boston, 1991.

[BTCGS91] Val Breazu-Tannen, Thierry Coquand, Carl Gunter, and Andre Scedrov. Inheritance as implicit coercion. *Information and Computation*, 93:172–221, 1991.

[CC77] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages*, pages 238–252. ACM, 1977.

[CCM87] Guy Cousineau, Pierre-Louis Curien, and Michel Mauny. The categorical abstract machine. *Science of Computer Programming*, 8, May 1987.

[CDCV81] Mario Coppo, Maria Dezani-Ciancaglini, and B. Venneri. Functional character of solvable terms. *Zeitschrift für mathematische Logic und Grundlagen der Mathematik*, 27:45–58, 1981.

[Fil94] Andrzej Filinski. Representing monads. In H.-J. Boehm, editor, *Conference Record of the 21st Symposium on Principles of Programming Languages*, pages 446–457, Portland, Oregon, January 1994. ACM Press.

[Hen93] Fritz Henglein. Type inference with polymorphic recursion. *ACM Transactions on Programming Languages and Systems*, 15(2):253–289, April 1993.

[HM94] Chris Hankin and Daniel Le Métayer. Deriving algorithms from type inference systems: Application to strictness analysis. In *Proceedings of the Twenty-First Annual ACM Symposium on Principles of Programming Languages, Portland*, pages 202–212. ACM, January 1994.

[Lau93] John Launchbury. A natural semantics for lazy evaluation. In S. Graham, editor, *Conference Record of the 20th Annual Symposium on Principles of Programming Languages*, pages 144–154, Charleston, South Carolina, January 1993. ACM Press.

[MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[MTH90] Robin Milner, Mads Tofte, and Robert Harper. *The Definition of Standard ML.* MIT Press, Cambridge, Massachusetts, 1990.

[NN92] Flemming Nielson and Hanne Riis Nielson. *Two-Level Functional Languages.* Cambridge University Press, 1992.

[Ode94] Martin Odersky. A functional theory of local names. In H.-J. Boehm, editor, *Conference Record of the 21st Symposium on Principles of Programming Languages*, pages 48–59, Portland, Oregon, January 1994. ACM Press.

[PA93] Gordon Plotkin and Martín Abadi. A logic for parametric polymorphism. In M. Bezem and J.F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93*, pages 361–375, Utrecht, The Netherlands, March 1993. Springer-Verlag LNCS 664.

[Rey72] John C. Reynolds. Definitional interpreters for higher-order programming languages. In *Proceedings of the 25th ACM National Conference*, pages 717–740, New York, 1972. ACM.

[Rey88] John C. Reynolds. Preliminary design of the programming language Forsythe. Technical Report CMU-CS-88-159, Carnegie Mellon University, Pittsburgh, Pennsylvania, June 1988.

[Rey91] John C. Reynolds. The coherence of languages with intersection types. In T. Ito and A. R. Meyer, editors, *International Conference on Theoretical Aspects of Computer Software*, pages 675–700, Sendai, Japan, September 1991. Springer-Verlag LNCS 526.

[Sny91] Wayne Snyder. *A Proof Theory for General Unification.* Birkhäuser, 1991.