1 Timeline

1969: LCF (Logic for Computable Functions), Dana Scott [8]

1972: Stanford LCF (LCF Proof-Checker), Robin Milner [3]

1973: ML (Meta Language), Robin Milner (Edinburgh) [4]

1977: Edinburgh LCF, Robin Milner [2]

1985: Cambridge LCF, Larry Paulson

1986: Isabelle, Larry Paulson [6]

1988: HOL88, Mike Gordon

1989: Simplifier, Tobias Nipkow [5]

1993: Isabelle/HOL

2 Recent History

1994: Inductive Predicates, Larry Pauslon

1998: Recursive Functions, Konrad Slind

1998: Record Types, Wenzel and Naraschewski

1999: (Co)Inductive Datatypes

1999: Isar, Wenzel

1999/2006: Locales, Ballarin, Kammüller, Paulson, Wenzel

2007: Function Package, Krauss

2007: Generic Code Generator, Haftmann

2007: Parallelism, Wenzel

2011: Isabelle/PIDE/jEdit, Wenzel

2013: (Co)Datatype Package based on Bounded Natural Functors, Dmitriy Traytel, Andrei Popescu, Jasmin Blanchette

3 Quotes

> The language PCF itself was introduced by Scott in what is probably the most well-known unpublished manuscript in Programming Language Theory.
>
> – C. A. Gunter, *Semantics of Programming Languages*

> This is a tale of errors, not of grand design.
>
> – L. Paulson, *The Next 700 Theorem Provers*

4 Features

- (Co)inductive sets [7]
- inductive datatypes [1]
- recursive functions [9]

References

[1] S. Berghofer and M. Wenzel. Inductive datatypes in HOL - Lessons learned in formal-logic engineering. In *Theorem Proving in Higher Order Logics, TPHOLs 1999*, volume 1690 of *Lecture Notes in Computer Science*, pages 19–36, 1999. doi:10.1007/3-540-48256-3_3.

[2] M. Gordin, R. Milner, and C. P. Wadsworth. *Edinburgh LCF: A Mechanized Logic of Computation*, volume 78 of *Lecture Notes in Computer Science*. 1979. doi:10.1007/3-540-09724-4.

[3] R. Milner. Logic for computable functions: Description of a machine implementation. Technical Report AIM-169, Computer Science Department, Stanford University, 1972.

[4] R. Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17(3):348–375, 1978. doi:10.1016/0022-0000(78)90014-4.

[5] T. Nipkow. Equational reasoning in Isabelle. *Science of Computer Programming*, 12(2):123–149, 1989. doi:10.1016/0167-6423(89)90038-5.

[6] L. C. Paulson. *Logic and Computer Science*, chapter Isabelle: The Next 700 Theorem Provers, pages 361–386. Academic Press, 1990.

[7] L. C. Paulson. A fixedpoint approach to implementing (co)inductive definitions. In *International Conference on Automated Deduction, CADE 1994*, volume 814 of *Lecture Notes in Computer Science*, pages 148–161, 1994. doi:10.1007/3-540-58156-1_11.

[8] D. S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*, 121(1-2):411–440, 1993. doi:10.1016/0304-3975(93)90095-B.

[9] K. Slind. Function definition in higher-order logic. In *Theorem Proving in Higher Order Logics, TPHOLs 1996*, volume 1125 of *Lecture Notes in Computer Science*, pages 381–397, 1996. doi:10.1007/BFb0105417.