Jean-Jacques Lévy
9, rue José Maria de Heredia
75007, Paris
tel: +33 6 43 94 26 11
jean-jacques.levy@inria.fr
http://pauillac.inria.fr/~levy

Born on 06/02/1947 in Toulouse
French citizen; Married + 2 children

SUMMARY

Jean-Jacques Lévy is a senior researcher emeritus at Inria. He graduated from the Ecole polytechnique in 1968, and got a PhD in computer science at University of Paris 7 in 1978.

He joined Inria in 1970, served as a member of the research staff at Digital Equipment (DEC-PRL, 1987-1988), a full professor at Ecole polytechnique (1992-2006), the director of the new Inria-Microsoft Research Joint Centre (2006-2013) and a visiting professor at the Chinese Academy of Sciences (ISCAS, Beijing, 2013-2014).

He headed two Inria research-teams (Para, Moscova), and two european projects (Confer-1, Confer-2); he was scientific chairman at Inria-Rocquencourt (1994-1996), vice chairman of the Inria national evaluation committee (1997-2000).

He supervised 20 PhD theses, was a consultant at Xerox-PARC (1984) and DEC-SRC (1989) in Palo Alto (Ca). He received the CNRS Médaille de Bronze (1979).

Jean-Jacques Lévy worked on compilers, lambda-calculus, term rewriting systems, CAD for VLSI, system programming, programming languages for distributed applications, formal proofs of programs. He participated to the debugging of the embedded software for the Ariane 5 rocket (after its explosion) and headed the on-board code review for the ISS european module Columbus. He is a (co-)author of 50 publications, 5 software and 1 US patent.

EDUCATION & AFFILIATIONS

Jul 1964	Baccalauréat in Nancy,
1964 - 1966	MathSup & MathSpé at lycées Henri Poincaré (Nancy) + Louis-le-Grand (Paris)
1966 - 1968	Student at the École Polytechnique, Paris,
Jul 1969	Master in Computer Science (DEA d'informatique) at Univ. of Paris 6,
1969 - 1984	Inria Junior Researcher (Chargé de Recherche CR2 + CR1),
Jun 1974	Small thesis in Computer Science (Thèse de 3ème cycle) at Univ. of Paris 7,
Jan 1978	PhD (Thèse d'Etat) at Univ. of Paris 7,
Jan 1979	Bronze medal, CNRS.
1988 - 1989	Member of the Research Team, Digital Equipment Corp., PRL,
1984 - 2012	Inria Senior Researcher (Directeur de Recherche DR2, DR1, DR0),
1989 - 1992	Associate Professor at the École Polytechnique, Palaiseau
1992 - 2006	Professor of Computer Science at École Polytechnique, Palaiseau,
1992 - 1999	Coordinator of EU projects CONFER-1-2-3,
1993 - 1995	Inria Scientific Chairman (chef du comité des projets) at Inria-Rocquencourt,
1996 - 2000	Vice-Chairman of Inria national evaluation committee (Inria CE)
2006 - 2012	Managing Director of the new Inria-Microsoft Research Joint Centre (Saclay),
2008 - 2011	Member of the Scientific Council of the Fondation des Sciences Mathématiques de Paris (FSMP)
2013 - 2014	Visiting Professor at the Chinese Academy of Sciences (ISCAS, Beijing)
2012 - now	Inria Emeritus, member of Inria - PiR2 research team at Univ. of Paris

MISCELLANEOUS

Languages	French (native), English (fluent), Russian (scholar), Chinese (beginner)
Sports	Swimming (62s, 100m free style) at ASPTT-CNN (Nancy) + PUC (Paris)

RESEARCH

The lambda-calculus: This area is related to Mathematical Logic and Models of Computation. Church invented the lambda-calculus in 1935 as a model of computation with functions. For instance, the identity function I is such that I(x) = x for all x. Therefore I(I) = I. Similarly the constant function K is such that K(x,y) = x for all x and y. Hence K(K,I) = K. Again let D(x) = x(x) for all x, then D(D) = D(D) ! These are nonsenses in standard mathematics, since a function cannot be applied to itself. But in computer science, these equations look natural. For instance, the evaluation of D(D) is just looping. Church proved that the lambda-calculus is as expressive as other models of computation, which was a striking result for a calculus with the sole functions. This led to his famous Church-Turing thesis, claiming that all general models of computation have the same expressiveness. But the lambda-calculus remained an obscure object reserved to logicians during a long period. In 1970, Scott gave a renewal of this calculus with its first functional model (in the logical sense). It was the kick off for the studies of meanings of programs and the denotational semantics of programming languages. Scott's idea was to incorporate non-termination inside functions at any order of functionality and to restrict denotations to "continuous" functions. For the lambda-calculus, Scott's model was a strong hint for the discovery of new properties of its syntax.

Optimal reductions in the lambda-calculus: My PhD work focused on safe and optimal evaluation strategies of the lambda-calculus. Safe computations are terminating; optimal computations minimize the number of computation steps to the result. For the latter, the difficulty is to define and implement sharing in the evaluation of functions. In my PhD, I designed a theory of Redex Families which extends the classic notion of residuals and which characterizes the redexes (reductible expressions) to share. These families are invariant w.r.t. permutations of reductions which by themselves give another interesting theory. The so-called Levy's labeled calculus permits to give computable names to redex families. But optimal strategies were difficult to implement until 1990 when Lamping provided a solution. With Abadi and Gonthier, we simplified Lamping's algorithm by relating it to Girard's geometry of interaction. In practice, these optimal strategies correspond (for weaker calculi) to lazy evaluators of functional programming languages such as Haskell or LML. My work on reduction strategies is the basis of chapter 13 in Barendregt's reference book about the Syntax and Semantics of the lambda-calculus and chapter 8 in Klop's Terese book. I also worked on free models with Bohm trees and proved the continuity theorem, as a corollary of completeness of inside-out strategies.

From the lambda-calculus to other calculi: I was a promoter (with Abadi, Cardelli and Curien) of Explicit Substitutions[1990]. It is a refined calculus of the lambda-calculus, which was studied at length by several authors, mainly to model run-time environments of programming languages. Another calculus related to my labeled lambda-calculus was the calculus of dependencies and incremental computations with Abadi and Lampson[1997], which was implemented and patented in the DEC/SRC Vesta makefile system (Levin et al). This dependency calculus has also been used and extended in various works about Information Flow for Security of Computing Systems. With Huet, we ported these results about the lambda-calculus to the general theory of Term Rewriting Systems [1980] and developed strongly sequential systems, as a basis for further theories on efficient pattern-matching in programming languages such as ML or Haskell. Klop wrote a book on these sequential systerms. Finally, Boudol, Castellani and Laneve showed the connection of permutations of reductions to Winskel's theory of events structures in concurrent systems. Indeed the labeled calculus reflects causality within various models of computations (e.g. reversible calculi for biology by Danos and Krivine).

Concurrency and distributed systems: Milner invented calculi to model concurrent processes, namely programs with several processes running in parallel. All computer scientists know that concurrency is a nightmare where debugging is Mission Impossible. Therefore concurrency is a wide area of research to discover the right tools for writing correct programs. Our contribution was the Join Calculus with Fournet, Gonthier, Maranget and Rémy[1997]. The Join Calculus is claimed to be the right language for abstract distributed implementions of Milner's pi-calculus and to avoid the distributed consensus problem. In this calculus, synchronization is based on Join Patterns. The Join Calculus induced the Jocaml implementation by le Fessant and Maranget on top of Ocaml and the Polyphonic C# by Russo on top of C# 2.0 and Join Patterns in Visual Basic 9.0 included in Microsoft Visual Studio. Another output of the Join Calculus was the applied pi-calculus of Abadi and Fournet used for the study of Security protocols.

Real Programming languages: I luckily worked in a central area with applications to compilers, run-times and designs of programming languages. Functional languages (Lisp, Haskell, Ocaml, ML, F#, Vesta, etc) use pattern-matching and higher-order functions. The semantics of imperative and object-oriented programming languages is also influenced by the semantics of the lambda-calculus. The study of evaluation rules in the lambda-calculus hints

optimizations in the efficiency of modern proof-assistants such as Coq, HOL or Isabelle. More generally, the lambda-calculus makes a tight connection between computer science and mathematical logic. As a practical outcome, in my Para group at Inria, we developed several concurrent implementations of functional programming languages (Concurrent LML, Concurrent Caml, Jocaml). In 1996, Gilles Kahn asked us to work on the debugging of the Embedded Software of the Ariane 5 rocket after its explosion due to a software bug (-300 millions euros). At EADS (les Mureaux, France), we made a static analysis of concurrent accesses to the numerous shared variables of the on-board flight programs. This successful ADA code inspection was based on the fantastic alias analyzer of Deutsch and the outstanding review of Gonthier. We were declared experts of space programs; I headed an international review of the embedded code of the ISS (International Space Station) Columbus european module in 1997 at Matra Marconi Space (Toulouse) and DASA (Bremen); with Gonthier I also contributed to the new programming rules for CNES and ESA!

More general programming: A less important part of my research has been related to Operating Systems, Interactive Graphics and CAD. In my Inria early days, I wrote the LP70 compiler (LP70 was the implementation of the Esope time-sharing system) [1970] and a text editor for Esope [1971]. Later I wrote a raster-op graphics package for LeLisp [1983], the Luciole layout editors for VLSI circuits [1984] written in LeLisp, a Unix event driver (multiplexing keyboard and mouse events) for the french sm90 workstation [1986]. Although less prestigious than theoretical papers, I always considered that part of my work as my most difficult achievements.

Program verification checked by computer: In my research-teams at Inria and MSR-Inria, we have experience of very long proofs checked with computer assistance: correctness of the non-obstrusive concurrent garbage collector of C-Caml (proved with the Larch prover by Doligez and Gonthier), the four-color theorem in planar graph theory (proved with Coq by Gonthier), the Feit-Thompson theorem in finite groups theory (proved with Coq/Ssreflect by Gonthier and his Math-Components group). There are long proofs in mathematics, there are long proofs in computer science to show properties of programs or security of software. We definitely want to make critical software proved 100% correct. My current research plans to make correctness proofs of standard algorithms, totally checked by computer in Why3 + Coq/Ssreflect. Why3 is (to my opinion) the best system (with Frama-C, Spec#, F*) for generating proof obligations on top of a programming language with Hoare logic assertions. I thus study the feasability of mixing automatic tools and interactive systems to prove properties of standard algorithms (as already achieved in Filliâtre's repository). With Chen Ran, Cohen, Merz and Théry, we formally proved the correctness of Tarjan's algorithm fo strongly connected components in graphs by 3 different methods (Why3, Coq, Isabelle-Hol).

TEACHING

[Contents of courses are visible at `http://pauillac.inria.fr/~levy/courses`]

École Polytechnique: I was in charge (with Cori) of the first-year grand Programming Course ("Algorithmes et Programmation", 430 students/year) [1991-2000]; I taught the second-year course on Basics of Computer Science ("Informatique Fondamentale", 200 students/year) [2000-2006] and several third-year courses in the Majeures; I was the coordinator of the Entrance Examination in Computer Science during 13 years. I also gave graduate level courses at the MPRI (Univ. of Paris 7).

UBA, Buenos Aires: I gave a course at Univ. of Buenos Aires (ECI) on Reductions and Causality in summer 2013.

Tsinghua University, Beijing: I gave courses on Jocaml (2009), Lambda-Calculus (2010), Reductions and Causality (2011), Functions and Datatypes in Coq (2013).

IISc, Bangalore: I co-organized (with K.Gopinath) the CIMPA-UNESCO School on Security of Computer Systems and Networks (2005). I gave a course on Concurrency Theory.

PhD students: I had the chance of being the advisor of 22 excellent phD students.

Sébastien Ailleret, Renault
Jade Alglave, Researcher, Microsoft Research Cambridge (co-supervised with Maranget)
Tomasz Blanc, France Telecom
Sylvain Conchon, Professor, Univ. of Paris-Saclay
Pierre-Malo Deniélou, Google, Mountain View (co-supervised with Leifer)
Damien Doligez, Researcher Inria, Rocquencourt
Francis Dupont, Telecom Bretagne
Fabrice le Fessant, Researcher Inria, Rocquencourt

Cédric Fournet, Principal Researcher, Microsoft Research Cambridge
Oliver Guillaumin, co-founder of Netgem
Nataliya Guts, retired from Post-doc, Maryland (co-supervised with Zappa Nardelli)
Jean Krivine, Researcher CNRS-IRIF, Univ. of Paris
Luc Maranget, Researcher Inria, Rocquencourt
Paul-André Melliès, Researcher CNRS-IRIF, Univ. of Paris
Gilles Peskine, Trusted Logic (co-supervised with Leifer)
Jérémy Planul, Post-doc, Stanford (co-supervised with Fournet)
Marc Pouzet, Professor, Ecole Normale Supérieure, Paris
Ma Qin, Assistant Professor, Univ. of Luxembourg (co-supervised with Maranget)
Alan Schmitt, Researcher Inria, Rennes

and more remotely:

Zena Ariola [Harvard], Professor, Eugene Oregon (with Arvind, MIT)
Chen Ran [Iscas], Huawei, Beijing (with Zhang Wenhui, Iscas)
Cosimo Laneve [Pisa], Professor, Bologna Italy (with Montanari, Pisa)

ADMINISTRATION

I participated to the program committees of POPL'91 (Orlando), PLDI'92 (Toronto), FPCA'93 (Copenhague), LICS'94 (Paris), ACSC Asian'95 + Asian'96 (Pathumthani + Singapore), TACS'97 (Sendai), PLILP'98 (Pisa), ICALP'98 (Aalborg).

I co-chaired IFIP TCS 2004 (Toulouse) with E.Mayr; I organized ICALP'01 BOTH workshop (Heraklion).

I have been a member of the Scientific Council of the Fondation Sciences Mathématiques de Paris [2007-2010]

I was the coordinator of 3 european projects (Confer 1-2-3) gathering Inria-Rocquencourt, Inria-Sophia Antipolis, ECRC Munich, Univ. of Edinburgh, CWI Amsterdam, ENS Paris, Imperial College London, Univ. of Pisa and SICS Stockhlom.

I headed the Inria research-teams Para [1988-2000], Moscova [2004, 2012]; I chaired the Inria-Rocquencourt Comité des projets [1994-1997]; and was vice-chair of the Inria National Commission d'Évaluation [1997-2000].

I acted as Head of the Computer Science department at the École polytechnique in Palaiseau [1991-2000] and was then a member of both the Pure Math and CS hiring committees (Commissions de Recrutement).

I am also proud of having managed the new Microsoft Research-Inria Joint Centre in Saclay, France [2006-2012]. The Centre has been a long-term and productive cooperation between 50 researchers at Cambridge (UK) and Inria.

PUBLICATIONS

[PDF contents are visible at `http://pauillac.inria.fr/~levy/pubs`]

- Mechanizable proofs about parallel processes, with Jean-Marie Cadiou, 14th Annual IEEE Symposium on Switching and Automata Theory (1973), pp.34-48. USA [ISSN: 0272-4847].
- Réductions sures dans le lambda-calcul, Paris 7, thèse de 3ème cycle, June, 1974. [Pdf, 4.7MB] (In French)
- An algebraic interpretation of the lambda-beta-K-calculus; and an application of a labelled lambda-calculus, Proceedings of the Rome Symposium on the Lambda calculus, 1975; also in Theoretical Computer Science 2 (1976), North Holland, pp.97-114. [Pdf]
- A structure oriented program editor: a first step towards computer assisted programming, with Véronique Donzeau-Gouge, Gérard Huet, Gilles Kahn, Bernard Lang, JJL, International Computing Symposium, North Holland, 1975
- Minimal and Optimal Computations of Recursive Programs, with Gérard Berry, Journal of the ACM, Vol 26, 1, Jan 1979, pp.148-175. [Pdf]. Also presented at the Fourth Annual ACM Symposium on Principles of Programming Languages (POPL) 1977.

- Réductions correctes et optimales dans le lambda-calcul, Paris 7, thèse d'Etat, January, 1978. [Pdf, 11.3MB] (In French)
- Le problème du partage dans l'évaluation des lambda-expressions, 1er colloque AFCET-SMF de Mathématiques Appliquées, Palaiseau, 4-8 Sept 1978. (In French)
- Approximations et Arbres de Bohm dans le lambda-calcul, Lambda Calcul et Sémantique formelle des langages de programmation, Actes de la 6ème École de printemps d'Informatique théorique, La Châtre, 1978, LITP-ENSTA. (In French)
- A Survey of Some Syntactic Results in the Lambda-calculus, with Gérard Berry. Proc. Ann. Conf. on Mathematical Foundations of Computer Science, Olomouc, Tchecoslovaquia, Lecture Notes in Computer Science 74, Springer-Verlag (1979).
- Optimal reductions in the lambda-calculus, To H.B.Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, edited by J.P.Seldin and J.R.Hindley, Academic Press, 1980. [Pdf].
- Full Abstraction for Sequential Languages: the State of the Art, with Gérard Berry and Pierre-Louis Curien, in Algebraic Methods in Semantics, Cambridge University Press (1985) 89-132. [Pdf].
- Attemps for Generalising the Recursive Path Orderings, with Sam Kamin, Manuscript, 1980. [Pdf].
- "Kruskaleries", Manuscript, October 1981. [Pdf]. (In French)
- "Dershowitzeries", Manuscript, October 1981. [Pdf]. (In French)
- Computations in Orthogonal Rewriting Systems, I, with Gérard Huet, Computational Logic; Essays in Honor of Alan Robinson, edited by J.-L.Lassez and G.Plotkin, The MIT Press, 1991. [Pdf].
- Computations in Orthogonal Rewriting Systems, II, with Gérard Huet, Computational Logic; Essays in Honor of Alan Robinson, edited by J.-L.Lassez and G.Plotkin, The MIT Press, 1991. [Pdf]. (Also research report 359, Inria, 1979)
- On the Lucifer System, VLSI architecture, Univ. of Bristol, Edited by B.Randell and P.C.Treleaven. Prentice Hall, 1982. [Pdf].
- L'éditeur Luciole, Unpublished note, with Gérard Baudet, 1983. [Pdf]. (In French)
- Le système Lucifer d'aide à la conception de circuits intégrés, with Jérôme Chailloux, Jean-Marie Hullot and Jean Vuillemin. Rapport de Recherche 196, Mars 1983. [Pdf]. (In French)
- Sharing in the Evaluation of lambda Expressions, in Programming of Future Generation Computers II, edited by K. Fuchi and L.Kott, Elsevier, North Holland, 1988. [Pdf].
- Explicit substitutions, with Martin Abadi, Luca Cardelli and Pierre-Louis Curien. Journal of Functional Programming, Vol. 1(4), pp. 375-416, 1991. [Pdf]. Also presented at the Seventeenth Annual ACM Symposium on Principles of Programming Languages (POPL) 1990, pp.31-46 San Francisco, USA. [Pdf].
- A Confluent Calculus of Substitutions, with Thérèse Hardin, France-Japan Artificial Intelligence and Computer Science Symposium, Izu, 1989.
- Confluence properties of Weak and Strong Calculi of Explicit Substitutions, with Thérèse Hardin and Pierre-Louis Curien, Journal of the ACM, 43(2):362–397. 1996. [Pdf].
- The Geometry of Optimal Lambda Reduction, with Martin Abadi and Georges Gonthier Proceedings of the Nineteenth Annual ACM Symposium on Principles of Programming Languages (January 1992), pp.15-26, Albuquerque, USA [Pdf].
- Linear Logic Without Boxes, with Martin Abadi and Georges Gonthier, Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science (June 1992), pp.223-234, Santa Cruz, USA [Pdf].
- An Abstract Standardisation Theorem, with Georges Gonthier and Paul-André Melliès, Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science (June 1992), pp.72-81, Santa Cruz, USA [Pdf].
- Analysis and Caching of Dependencies, with Martin Abadi and Butler Lampson, Proceedings of the 1996 ACM International Conference on Functional Programming (May 1996), pp.83-91, Philadelphia, USA [Pdf].

- A Calculus of Mobile Agents, with Cédric Fournet, Georges Gonthier, Luc Maranget and Didier Rémy, Proceedings of the 7th International Conference on Concurrency Theory (CONCUR), 1996, Pisa, Italy [Pdf].
- Some results in the Join-Calculus, Proceedings of the Third International Symposium on the Theoretical Aspects of Computer Software (TACS), Sendai, Japan. Lecture Notes in Computer Science, 1281, Springer-Verlag (1997). [Pdf].
- Explicit Substitutions and Programming Languages, with Luc Maranget, FSTTCS '99: foundations of software technology and theoretical computer science, edited by C.Pandu Rangan, V.Raman, R.Ramanujam. LNCS, 1738, Springer, 1999, pp.181-200, Chennai, India [Pdf].
- An asynchronous distributed implementation of Ambients, with Cédric Fournet and Alan Schmitt, Proceedings of the second IFIP Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics, edited by J.van Leeuwen, O.Watanabe, M.Hagiya, P.D.Mosses, T.Ito, LNCS, 1872, Springer, pp.348-364, 2000, Sendai, Japan. [Pdf].
- Sharing in the weak lambda-calculus, with Tomasz Blanc and Luc Maranget. Processes, Terms and Cycles: Steps on the Road to Infinity. Essays dedicated to Jan Willem Klop. LNCS 3838, Springer, 2005. [Pdf].
- Sharing in the weak lambda-calculus revisited, with Tomasz Blanc and Luc Maranget. In Reflections on Type Theory, Lambda Calculus, and the Mind, Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday, Erik Barendsen, Herman Geuvers, Venanzio Capretta, Milad Niqui (Eds.) 2007. [Pdf].
- Structures de données. In Encyclopédie de l'informatique et des systèmes d'information. Editor Jean-Eric Pin. Vuibert (Ed.), pp. 919-928, 2008. ISBN : 978-2-7117-4846-4. (In French) [Pdf]
- Generalized Finite developments (2007). In Essays in Honour of Gilles Kahn, Cambridge University Press 2009. ISBN-13: 9780521518253 [Pdf].
- Les Dominos de Wang, Revue Quadrature, "Magazine de mathématiques pures et épicées", 82, Octobre-novembre-décembre 2011 (5 pages). [Pdf]. (In French)
- Redexes are stable in the lambda-calculus. A special issue dedicated to Corrado Bŏhm for his 90th birthday. Submitted 2012. Published in Mathematical Structures in Computer Science, 27(5), pp. 738-750, 2017. [Pdf].
- The cost of usage in the lambda-calculus, with Andrea Asperti. Twenty-Eighth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2013), June 25-28, 2013, New Orleans, USA [Pdf].
- Simple proofs of simple programs in Why3. Essays for the Luca Cardelli Fest, ed. Martin Abadi and Philippa Gardner and Andrew D. Gordon and Radu Mardare, MSR-TR-2014-104, September 2014, Microsoft Research Cambridge [Pdf].
- Readable semi-automatic formal proofs of Depth-First Search on graphs using Why3, with Chen Ran. Inria Tech. Report, October 2015. [Pdf].
- Une preuve formelle de l'algorithme de Tarjan-1972 pour trouver les composantes fortement connexes dans un graphe, with Ran Chen. Journées Francophones des Langages Applicatifs (JFLA), Gourette, January 2017. (In French) [Pdf].
- A Semi-automatic Proof of Strong Connectivity, with Ran Chen. 9th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), Heidelberg, July 2017. [Pdf].
- Formal Proofs of Tarjan's Strongly Connected Components Algorithm in Why3, Coq and Isabelle, with Ran Chen, Cyril Cohen, Stephan Merz, Laurent Théry. 10th conference on Interactive Theorem Proving (ITP), Portland, USA, September 2019. [Pdf].

Volumes

- Algorithmes et programmation, with Robert Cori. (290 pages) Les Éditions de l'École polytechnique 1992. ISBN 2-7302-0619-1.
- Algorithms, Concurrency and Knowledge, Co-edited with Kanchana Kanchanasut. Proceedings of the First Asian Computing Science Conference (ASIAN), Pathumthani, Thailand, December 11-13, 1995. Lecture Notes in Computer Science 1023, (410 pages), Springer. ISBN: 3540606882.

- Informatique fondamentale (271 pages) Les Éditions de l'École polytechnique 2001. ISBN 2-7302-1004-0.
- Exploring New Frontiers of Theoretical Informatics, Co-edited with Ernst W.Mayr and John C.Mitchell. Proceedings of the IFIP 18th World Computer Congress, TC1 3rd International Conference on Theoretical Computer Science (TCS2004), 22-27 August 2004, Toulouse, France, (674 pages). Kluwer 2004. ISBN-13: 978-1441954862.
- Introduction à la théorie des langages de programmation, with Gilles Dowek. (110 pages) Les Éditions de l'École polytechnique 2006. ISBN10 : 2-7302-1333-3.,
- Introduction to the Theory of Programming Languages, with Gilles Dowek. (Undergraduate Topics in Computer Science). (108 pages) Springer. ISBN-13: 978-0857290755.
- From Semantics to Computer Science, Essays in Honour of Gilles Kahn. Co-edited with Yves Bertot, Gérard Huet and Gordon Plotkin. (594 pages). Cambridge University Press 2009. ISBN-13: 9780521518253.

SOFTWARE

- LP70 compiler [1970] used as a key building block of the Esope time-sharing system.
- Luciole [1983], a programmable layout editor for VLSI circuits, connected to LeLisp.
- Efficient raster-op package [1984] on top of Colorix and LeLisp.
- An event driver [1986] for the keyboard & mouse of the SM90 workstation running Unix (with F. Ledru).
- A library of formal proofs [2018] for graph algorithms on top of Why3 and Coq) at `http://pauillac.inria.fr/~levy/why3` (with Chen Ran) presented at ITP'2019.