CURRICULUM VITAE

Frédéric MESNARD

February 2010

Contents

1 Personal Data

Surname: MESNARD
First name: Frédéric
Date of birth: April 22, 1962
Nationality: French

1.1 Addresses

Work: Université de la Réunion
Faculté des sciences et technologies
Parc technologique universitaire - Bâtiment 2
2, rue Joseph Wetzell - 97490 Sainte-Clotilde
La Réunion - France
Tel: +262 262 48 33 95 – Fax: +262 262 48 33 91

Email: Frederic.Mesnard@univ-reunion.fr

Web: http://personnel.univ-reunion.fr/fred/

1.2 Current Professional Position

Professor of Computer Science, Department of Mathematics and Informatics of the University of Reunion, France.

1.3 Education

- Engineer, École Nationale de l'Aviation Civile, Toulouse, France, 1986.
- Ph.D. in Computer Science, University Paris VI, France, 1993.
- Habilitation, University of Reunion, France, 2001.

1.4 Foreign Language Spoken and Written

English.

1.5 Academic Employment

1986–1988: Teaching Assistant, University of Reunion.

1989–1993: Research activity as PhD student, University Paris VI and the University of Reunion.

1993–2002: Research and teaching activity as Assistant Professor, Department of Mathematics and Informatics of the University of Reunion.

2002–Present: Research and teaching activity as Professor, Department of Mathematics and Informatics of the University of Reunion.

2 Collective Activities

- Coordinator of the master Informatique et Mathématiques since 2006.
- Pedagogical supervisor of the 4th year studies in "Computer Science" since 2004, helped by Etienne Payet since 2007.
- Former team leader of the IREMIA, 2007–2009.
- Former lab leader of the Laboratoire d'Informatique et de Mathématiques, EA 2525, 2007–2009.

3 Teaching Activities

Each activity listed in this section has its own associated web page.

- "Introduction to Logic", L3 degree in "Computer Science".
- "Logic Programming and Prolog", L3 degree in "Computer Science'.
- "Computability and Complexity", master degree in "Mathematics" and "Computer Science".
- "Elements of Theoretical Computer Science", first year of the ESIROI-STIM school of IT engineering.

4 Research Activities

4.1 Research Interests

- Formal methods for the analysis and verification of computer programs
- Static analysis, abstract interpretation
- Termination and non-termination
- Logic programming and constraint logic programming

4.2 Research Period and Activity Outside France

- *Visiting researcher* at the Department of Computer Science of the University of Melbourne, Australia, for a joint research on *Termination Analysis of Typed Logic Programming Languages* with the group of Prof. P. J. Stuckey, January–June 2003, see [18].
- *Invited professor*, University of Fianarantsoa, Madagascar, May 2007, for a course on verification.
- *Invited professor*, University of Verona, Italy, May 2008, for a course on automatic termination analysis of programs, see [8].

4.3 Ph.D. and Habilitation Committees

- Alexander Serebrenik, Ph.D., K. U. Leuven, Belgium, 2003.
- Jean Diatta, habilitation, University of Reunion, 2003.
- Jean-Dany Vally, Ph.D., University of Reunion, 2003.
- Serge Burckel, habilitation, University of Reunion, 2005.
- Tristan Denmat, Ph.D., IRISA, Rennes, 2008.
- Nicolas Sébastien, Ph.D., University of Runion, 2009.
- Etienne Payet, habilitation, University of Reunion, 2009.
- Yves Dumont, habilitation, University of Reunion, 2009.

4.4 Program Committees

- Program Committee member for the international workshop *Verification of Logic Programs*, ICLP'99 (Las Cruces, New Mexico, Etats Unis).
- Program Committee member for the *journées francophones sur la programmation logique avec contraintes* in 2001, 2002 and 2003.
- **co-organizer and co-chair** for the 13th *workshop on Logic Programing Environments*, Mumbai, India, décember 2003
- **Program chair** for the 13th *journées francophones de programmation en logique et programmation par contraintes*, 2004.
- Program Committee member for the *journées nationales sur la résolution pratique de problèmes NP-complets*, 2004.
- Program Committee member for the *International Workshop on Terminaison* in 2004 (Aachen, Germany).
- Program Committee member for the *journées francophones sur la programmation par contraintes* in 2005, 2006, and 2007.
- Program Committee member for *workshop on Logic Programing Environments*, in 2004 (Saint Malo, France), 2005 (Sitges, Spain) and 2006 (Seattle, Washington, USA).
- Program Committee member for LOPSTR, *International Symposium on Logic-based Program Synthesis and Transformation* ; in 2005 (London, UK), 2006 (Venice, Italy), and 2009 (Coimbra, Portugal).
- Program Committee member for the *ACM SAC - Software Verification Track* in 2008 (Fortaleza, Brazil).
- Anonymous referee for:
  - *Journal of Logic Programming.*
  - *Journal of Functional and Logic Programming.*

– *Theory and Practice of Logic Programming.*
– *ACM Transactions on Computational Logic.*
– *ACM Transactions on Programming Languages and Systems.*

5 Publications

5.1 International Journals

[1] F. Mesnard, S. Hoarau, and A. Maillard. CLP(χ) for automatically proving program properties. *Journal of Logic Programming*, pages 77–93, 1998.

[2] F. Mesnard and S. Ruggieri. On proving left termination of constraint logic programs. *ACM Transactions on Computational Logic*, 4(2):207–259, 2003.

[3] F. Benoy, A. King, and F. Mesnard. Computing convex hulls with a linear solver. *Theory and Practice of Logic Programming*, 5(1–2):259–271, 2005.

[4] F. Mesnard and R. Bagnara. cTI: a constraint-based Termination Inference tool for ISO-Prolog. *Theory and Practice of Logic Programming*, 5(1–2):243–257, 2005.

[5] E. Payet and F. Mesnard. Non-termination inference of logic programs. *ACM Transactions on Programming Languages and Systems*, 28(2):256–289, 2006.

[6] F. Mesnard and A. Serebrenik. Recurrence with affine level mappings is P-time decidable for CLP(R). *Theory and Practice of Logic Programming*, 8(1):111–119, 2008.

[7] E. Payet and F. Mesnard. A Non-Termination Criterion for Binary Constraint Logic Programs. *Theory and Practice of Logic Programming*, 9(2): 145–164, 2009.

[8] F. Spoto, F. Mesnard, and E. Payet. A Termination Analyser for Java Bytecode Based on Path-Length. *ACM Transactions on Programming Languages and Systems*, 2010, to appear.

[9] S. Ruggieri and F. Mesnard. Typing Linear Constraints. *ACM Transactions on Programming Languages and Systems*, 2010, to appear.

5.2 International Conferences

[10] F. Mesnard and J-G. Ganascia. CLP(Q) for proving interargument relations. In A. Pettorossi, editor, *Meta-Programming in Logic*, volume 649 of *LNCS*, pages 308–320. Springer, 1992.

[11] F. Mesnard. Towards automatic control for CLP(χ) programs. In M. Proietti, editor, *Logic-Based Program Synthesis and Transformation*, volume 1048 of *LNCS*, pages 106–119. Springer, 1995.

[12] F. Mesnard. Inferring left-terminating classes of queries for constraint logic programs. In M. Maher, editor, *Joint International Conference and Symposium on Logic Programming*, pages 7–21. MIT Press, 1996.

[13] F. Mesnard, S. Hoarau, and A. Maillard. CLP(χ) for proving program properties. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems*, pages 321–338. Kluwer Academic Publishers, 1996.

[14] S. Hoarau and F. Mesnard. Inferring and compiling termination for constraint logic programs. In P. Flener, editor, *Logic-Based Program Synthesis and Transformation*, volume 1559 of *LNCS*, pages 240–254. Springer, 1998.

[15] U. Neumerkel and F. Mesnard. Localizing and explaining reasons for non-terminating logic programs with failure-slices. In G. Nadathur, editor, *Principles and Practice of Declarative Programming*, volume 1702 of *LNCS*, pages 328–341. Springer, 1999.

[16] F. Mesnard and U. Neumerkel. Applying static analysis techniques for inferring termination conditions of logic programs. In P. Cousot, editor, *Proc. of SAS'01*, pages 93–110. Springer, 2001. LNCS 2126.

[17] F. Mesnard, E. Payet, and U. Neumerkel. Optimal termination conditions of logic programs. In M. Hermenegildo and G. Puebla, editors, *Proc. of SAS'02*, pages 509–526. Springer, 2002. LNCS 2477.

[18] V. Lagoon, F. Mesnard, and P. J. Stuckey. Termination analysis with types is more accurate. In Catuscia Palamidessi, editor, *ICLP*, volume 2916 of *Lecture Notes in Computer Science*, pages 254–268. Springer, 2003.

[19] E. Payet and F. Mesnard. Non-termination inference for constraint logic programs. In R. Giacobazzi, editor, *Proceedings of SAS'04*, pages 377–392. Springer, 2004. LNCS 3148.

[20] A. Serebrenik and F. Mesnard. On termination of binary clp programs. In *proceedings of LOPSTR'04*. LNCS 3573, Springer, 2005.

[21] S. Ruggieri and F. Mesnard. Typing Linear Constraints for Moding CLP(R) Programs. In M. Alpuente and G. Vidal, editors, *Proceedings of SAS'08*, pages 128–143. Springer, 2008. LNCS 5079.

[22] S. Ruggieri and F. Mesnard. Variable Ranges in Linear Constraints. *Proceedings of ACM SAC'2010*, 25th Symposium On Applied Computing. To appear.

5.3 International Workshops

[23] F. Mesnard and M. Morillon. Linear measures for controlling clp(q). In M. Billaud, P. Castéran, M-M. Corsini, K. Musumbu, and A. Rauzy, editors, *Workshop on Static Analysis*, number 81–82 in Bigre, pages 29–34, 1992.

[24] S. Colin, F. Mesnard, and A. Rauzy. Constraint logic programming and mu-calculus. *ERCIM/COMPULOG Workshop on Constraints*, 1997.

[25] F. Mesnard. Entailment and Projection for CLP(B) and CLP(Q) in SICStus Prolog. In P. Hill, J. Gallagher, and A. King, editors, *Constraint Reasoning For Constraint Programming*, 1997. Leeds, U.K.

[26] F. Mesnard and S. Hoarau. A tabulation algorithm for CLP. In D.S. Warren et al., editor, *Tabling in Logic Programming*, pages 13–24, 1997. Leuven, Belgium.

[27] S. Burckel, S. Hoarau, F. Mesnard, and U. Neumerkel. cTI : bottom-up termination inference for logic programs. In *Workshop on Logic Programming*, 2000. Berlin, Germany.

[28] S. Hoarau, F. Mesnard, and U. Neumerkel. Implementing cTI : a constrained-based left-Termination Inference tool for LP. In I. de Costra Dutra, editor, *Parallelism and Implementation Technology for (Constraint) Logic Programming Languages*, 2000. London, U.K.

[29] E. Payet and F. Mesnard. An improved non-termination criterion for binary constraint logic programs. In Alexander Serebrenik and Susana Muñoz-Hernández, editors, *Proceedings of the 15th International Workshop on Logic Programming Environments, Sitges (Barcelona)*, pages 46–60, 2005.

[30] F. Spoto, L. Lu, and F. Mesnard. Using CLP Simplications to Improve Java Bytecode Termination Analysis. In S. Genaim, editor, *Proceedings of BYTECODE 2009*, Bytecode Semantics, Verification, Analysis and Transformation York, UK , 29th March 2009.

5.4 French Conferences

[31] F. Mesnard and J-G. Ganascia. Contrôler la résolution par sélection des littéraux et déduction partielle. In *Reconnaissance de Formes et Intelligence Artificielle*, pages 1309–1314, 1991.

[32] F. Mesnard and J-G. Ganascia. Clp(x) for proving program properties. In J-P. Delahaye, P. Devienne, P. Mathieu, and P. Yim, editors, *Journées Francophones de Programmation en Logique*, pages 328–338, 1992.

[33] F. Mesnard. Approximations entre langages de programmation logique avec contraintes. In P. Ezequel, editor, *Journées Francophones de Programmation en Logique*, pages 319–341, 1993.

[34] F. Mesnard. Etude de la terminaison des programmes logiques avec contraintes, au moyen d'approximations. In M-M. Corsini, editor, *Journées Francophones de Programmation en Logique*, pages 205–219, 1994.

[35] F. Mesnard and S. Hoarau. Contrôle dynamique de la résolution pour les programmes logiques avec contraintes. In J-J. Chabrier, editor, *Journées Francophones de Programmation en Logique*, pages 259–273, 1995.

[36] S. Hoarau and F. Mesnard. PLC(Bool) pour la détection de variables numériques bornées. In J-L. Imbert, editor, *Journées Francophones de Programmation Logique et programmation par Contraintes*, pages 169–183. Herms, 1996.

[37] S. Hoarau and F. Mesnard. Inférer and compiler la terminaison pour les programmes logiques avec contraintes. In O. Ridoux, editor, *Journées Francophones de Programmation Logique et par Contraintes*, pages 269–286. Herms, 1998.

[38] S. Colin, F. Mesnard, and A. Rauzy. Un module Prolog de mu-calcul booléen : une réalisation par BDD. In J-F. Boulicaut, editor, *Journées Francophones de Programmation Logique et programmation par Contraintes*, pages 23–38. Herms, 1999.

[39] F. Mesnard and A. Rauzy. Le *iota*-calcul : un langage de contraintes d'ordre supérieur. In Touravane, editor, *Journées Francophones de Programmation Logique avec Contraintes*. Herms, 2000.

[40] F. Mesnard, U. Neumerkel, and E. Payet. cTI : un outil pour l'inférence de conditions optimales de terminaison pour Prolog. In P. Codognet, editor, *Actes des JFPLC'01*, pages 271–286. Herms, 2001. Paris.

[41] F. Mesnard, E. Payet, and U. Neumerkel. Non-terminaison and optimalité des conditions de termination pour les programmes logiques In M. Rueher, editor, *Actes des JFPLC'02*. Herms, 2002. Paris.

[42] E. Payet and F. Mesnard. Inférence de non-terminaison pour les programmes logiques avec contraintes. In F. Mesnard, editor, *Actes des JFPLC'04*, pages 55–72. Herms, 2004. Paris.

5.5 Book

[43] F. Mesnard, editor. *Programmation en logique avec contraintes, JFPLC 2004, 21, 22 et 23 Juin 2004, Angers, France*. ISBN: 2-7462-0937-3. Hermes, 2004

5.6 Thesis

[44] F. Mesnard. *Étude de la terminaison des programmes logiques avec contraintes, au moyen d'approximations*. PhD thesis, Université Paris VI, 1993.

[45] F. Mesnard. Autour de la terminaison des programmes logiques avec contraintes. Université de La Réunion, 2001. Habilitation à diriger des recherches.

5.7 Other Published Writings

[46] F. Mesnard and J-G. Ganascia. A propos du contrôle de la résolution. In *Journées de Travail sur l'Analyse Statique de la Programmation Equationnelle, Fonctionnelle et Logique*, number 74 in Bigre, pages 125–131, 1991.

[47] F. Mesnard and U. Neumerkel. CHR for prototyping abstract interpretation. *Journal of Applied Artificial Intelligence*, 14(4), 2000. in Selected recent project descriptions involving CHR.

[48] F. Mesnard and A. Serebrenik, editors. *Proceedings of the 13th International Workshop on Logic Programming Environments, Tata Institute of Fundamental Research, Mumbai, India, December 8, 2003*, volume CW371 of *Report*. Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan 200A, B-3001 Heverlee (Belgium), 2003.

6 Software

6.1 cTI: a termination inference analyzer for ISO-Prolog

Released as a GNU/GPL-ed prototype, cTI is the first bottom-up left-termination inference tool for ISO-Prolog which answers to the question: "when does a logic program terminate?". *Termination inference*, a new kind of program analysis, answers this question with a compact formula, called a *termination condition*. All queries whose termination has been proven are described by this condition. Hence termination inference is an *annotation free* generalization of termination analysis/checking. It shifts the programmer's focus away from particular cases to the whole relation. Traditionally, a termination analyzer tries to prove that a given class of queries terminates. This class must be provided by the user, which is rather cumbersome if the programs have been written previously without any annotations. With termination inference no annotations are necessary. All provably terminating classes to all related predicates are *inferred* at once, illustrating the "multi-directionality" of predicates. Since its initial launch in April 2000, cTI has been enhanced with:

- **types:** in [18], we adapt cTI to take into account *descriptive* or *prescriptive* type information ;
- **precision:** in [5], we combine cTI with a non-termination analyzer. The resulting analyzer can sometimes conclude that with respect to the language used to describe classes of queries the analysis is *optimal*.

6.2 Julia+BinTerm: a Java Bytecode termination analyzer

Julia is a static analyzer for Java Bytecode. A YouTube video summarizes our approach, which is detailed in [8]. In 2009, Julia participated to the International Termination Competition, arrived second in the Java Bytecode category and *first* in the Java Bytecode Recursive category.