

CURRICULUM VITÆ

DAVID PICHARDIE

PERSONAL DETAILS

Nationality	French
Date of Birth	30 April 1977
Familial Position	Married
Home Address	48 chemin des Groulles 06600 Antibes France
Office Address	Everest Research Group, INRIA Sophia-Antipolis 2004 Route des Lucioles BP 93, 06902 Sophia Antipolis Cedex France
Office Tel	+33 4 92 38 75 59
E-mail	David.Pichardie@sophia.inria.fr
Homepage	http://www-sop.inria.fr/everest/personnel/David.Pichardie/

EDUCATION

2002-2005	Phd Thesis in Computer Science at University Rennes 1 and Irisa. Title : <i>Abstract interpretation in intuitionist logic: extraction of certified Java analysers</i> . Defended in December 2005.
1998-2002	Student at ENS Cachan, brittany site Agrégation de mathématiques ¹ . Rank : 18 th amongst 350 successful candidats Master of Mathematics and Computer Science at ENS Cachan and University Rennes 1, including : <ul style="list-style-type: none">Pre-doctoral studies in Computer Science, first class honours. Rank : 1stMSc in Computer Science, first class honours. Rank : 1stMSc in Mathematics, first class honours. Rank : 5thBSc in Computer Science, first class honours. Rank : 4thBSc in Mathematics, first class honours. Rank : 1st

RESEARCH INTERESTS

My research interests currently deal with non-interference type systems. We are currently implementing and proving correct in Coq a modular analysis for Java bytecode programs. Our aim is to illustrate the precision of the analysis on real case studies.

During the first months of my postdoc in INRIA Sophia Antipolis I have been working on an extension of Coq to facilitate the definition and the reasoning of recursive general functions. This work has lead to a paper in FLOPS'06.

¹A high-level national competitive examination for recruiting teachers.

My phd's works dealt with the development of certified static analyses using the Coq proof assistant and the abstract interpretation theory. During this work, I have developed static analyses for a programming language similar to Java bytecode. The extraction features of Coq have enabled me to extract an analyser written in OCaml, from the correctness proof of the analysis. The correctness proof follows the abstract interpretation methodology. The main theorem establishes that the analyser computes correct informations with respect to the operational semantics of a program. I propose a generic framework to formalise static analysers in Coq. This work is mainly explained in the article "Extracting A Data Flow Analyser in Constructive Logic", TCS'05 (extended version of ESOP' 03).

We have recently applied these work for proof carrying code. We propose an abstract interpretation framework to build PCC architecture with a downloadable Coq soundness proof of the certificate checker. The architecture has been evaluated experimentally on a byte code language for which we have designed an interval analysis that allows to generate certificates ascertaining that no array-out-of-bounds accesses will occur. This work will be presented at EAI'06.

My previous research activities were related to formalisation in Coq of geometrical algorithms like computation of convex hulls (see "Formalizing Convex Hulls Algorithms", TPHOLs' 01), proof of programs describing hardware implementations using systems of affine recurrence equations (see "Embedding of Systems of Affine Recurrence Equations in Coq", TPHOLs' 03) and formalisation, in PVS, of an algorithm of optimization used for the compilation of JavaCard programs (see "A Java Card CAP converter in PVS", COCV' 03).

PUBLICATIONS

- [1] Frédéric Besson, Thomas Jensen and David Pichardie. *A PCC Architecture based on Certified Abstract Interpretation*. Proc. of the first International Workshop on Emerging Applications of Abstract Interpretation (EAAI'06), Electronic Notes in Theoretical Computer Science, 2006. To appear.
- [2] Gilles Barthe, Julien Forest, David Pichardie and Vlad Rusu. Defining and reasoning about recursive functions: a practical tool for the Coq proof assistant. In *Proc. of 8th International Symposium on Functional and Logic Programming (FLOPS'06)*, Lecture Notes in Computer Science. Springer-Verlag, 2006. To appear.
- [3] David Pichardie. Modular proof principles for parameterized concretizations. In *Proc. of 2th International Workshop on Construction and Analysis of Safe, Secure and Interoperable Smart Devices (CASSIS 2005)*, Lecture Notes in Computer Science. Springer-Verlag, 2005. To appear.
- [4] David Cachera, Thomas Jensen, David Pichardie, and Vlad Rusu. Extracting a Data Flow Analyser in Constructive Logic. *Theoretical Computer Science*, 342(1):56–78, September 2005. Extended version of [6].
- [5] David Cachera, Thomas Jensen, David Pichardie, and Gerardo Schneider. Certified memory usage analysis. In *Proc. of 13th International Symposium on Formal Methods (FM'05)*, number 3582 in Lecture Notes in Computer Science, pages 91–106. Springer-Verlag, 2005.
- [6] David Cachera, Thomas Jensen, David Pichardie, and Vlad Rusu. Extracting a Data Flow Analyser in Constructive Logic. In *Proc. of 13th European Symposium on Programming (ESOP'04)*, number 2986 in Lecture Notes in Computer Science, pages 385–400. Springer-Verlag, 2004.
- [7] Thomas Genet, Thomas Jensen, Vikash Kodati, and David Pichardie. A Java Card CAP converter in PVS. In Jens Knoop and Wolf Zimmermann, editors, *Proc. of 2nd International Workshop on Compiler Optimization Meets Compiler Verification (COCV 2003)*, *Electronic Notes in Theoretical Computer Science*, volume 82. Elsevier, 2004.
- [8] David Cachera and David Pichardie. Embedding of Systems of Affine Recurrence Equations in Coq. In *Proc. of 16th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'03)*, number 2758 in Lecture Notes in Computer Science, pages 155–170. Springer-Verlag, 2003.
- [9] David Pichardie and Yves Bertot. Formalizing Convex Hulls Algorithms. In *Proc. of 14th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'01)*, number 2152 in Lecture Notes in Computer Science, pages 346–361. Springer-Verlag, 2001.

TEACHING

2005 Lecture (3 hours) on Abstract Interpretation for pre-doctoral students in the Nice University (France).

2002-2005 Teaching Assistant at ENS Cachan, brittany site and University Rennes 1. Class work and practical work in undergraduate modules:

- Logic and Calculability (propositional and first-order logic, proof systems, temporal logic, logic programming, turing machines). For the practical work, I have developed a small proof assistant to manage proof in the natural deduction proof system.
- Algorithmic Methods (discrete mathematics, dynamic programming, greedy algorithms, backtracking, P&NP)
- Compilers (lexical and syntax analysis, code generation). For the practical work, I have developed a small Logo compiler and an associated Logo Virtual Machine.
- Functional Approach to Programming (Scheme programming).
- Mathematics (linear algebra, real analysis).

2001-2002 Practical work in Mathematics and Computer science in *classes préparatoires* at Lycée Chateaubriand, Rennes (preparatory courses to French high schools): functional programming (Caml langage) and applied mathematics (with Maple).

Many notes about these teaching can be found on my web page (in French):

<http://www.irisa.fr/lande/pichardie/enseignement.html>

I have already some experience in giving talk in english, as my list of talk can demonstrate:

<http://www.irisa.fr/lande/pichardie/publications/recherche.en.html#talk>

VISITING POSITIONS

January-February 2005 Collaboration with David Monniaux at École Normale Supérieure (Paris), in the Semantics and Abstraction Interpretation Team (team directed by Patrick Cousot).
Subject : *Formal proofs on symbolic manipulations of floating-point numbers*.

Summer 2000 and 2001 Collaboration with Yves Bertot, INRIA Sophia-Antipolis, Lemme team.
Subject : *Formalizing Convex Hulls Algorithms*