University Paris Sud (Orsay) site and ENS Lyon subsite

1/08/2002-31/07/2003

1 Participants

The following people participated to the TYPES project in association with Laboratoire de Recherche en Informatique, UMR 8623, CNRS-Université Paris Sud, Orsay.

Permanent researchers

- Judicaël Courant
- Jean Duprat (ENS Lyon)
- Jean-Christophe Filliâtre
- Claude Marché
- Christine Paulin (Site leader)

Post-Doc researchers

- Alexandre Miquel
- Benjamin Monate (École Polytechnique)
- Xavier Urbain

PhD students

- Pierre Corbineau
- Pierre Letouzey
- Nicolas Oury
- Julien Signoles

2 Results

2.1 The Coq proof assistant

The Coq Proof assistant [1, 2] is conjointly developed by the Paris Sud University and the INRIA Rocquencourt site. A new version V7.4 integrating the module system developed by Jacek Chrząszcz during his PhD at both LRI and Warsaw University was released in February 2003.

CoqIde a Gtk graphical interface for Coq B. Monate developed a new interface coqide for the Coq system. This interface is developed in Ocaml using the lablgtk library. It is integrated with the Coq system and provides in particular a good localization of errors and integration of notations using unicode characters.

Program extraction P. Letouzey [7] worked on program extraction in Coq. He designed a method to assign simple types which are approximations of Coq types to extracted program. He extended the extraction mechanism to Coq modules and functors which are extracted towards Ocaml modules or functors.

At the TLCA'03 international conference, N. Oury [12] presented his work on a generic scheme to substitute efficient data-structures within programs extracted from Coq proofs, while preserving both the correctness of the extraction process and the computational behavior of data structures on the Coq side.

Automation of proofs J. Courant and P. Corbineau worked on proof automation in Coq. P. Corbineau designed the new tactic `Ground` which extends the decision procedure for intuitionistic propositional calculus to solve some first-order problems. He also integrated his tactic `CC` for congruence closure to the Coq system.

2.2 Developments

Finite sets library P. Letouzey and J.-C. Filliâtre formalized in the Coq proof assistant a finite sets library[1]. It actually contains three different implementations of finite sets: using ordered lists, red-black trees and AVLs. This formalization makes a heavy use of modules and functors, a novelty in Coq 7.4, thus demonstrating their power in the context of proof, as well as in the context of programming. These developments are available as Coq contributions.

Vectors J. Duprat developed a theory of bit-vectors with their interpretation as numeric values in order to modelize types of bounded integers in programming languages. It is integrated to the standard Coq theories.

Timed automata A new version of a theory of timed automata using the functor system of Coq V7.4 has been designed by C. Paulin and is used in the Calife tool[2], an environment for formal proofs and tests of algorithms used in telecommunication which uses both proof assistants and model-checkers. These developments are available as Coq contributions.

2.3 Proof of programs

Why Jean-Christophe Filliâtre continued the development of the Why tool [5] to certify imperative programs using proof assistants Coq, PVS or HOL Light. He extended the tool with an output for the haRVey[3] decision procedure which was able to discharge automatically proof obligations for the heapsort algorithm.

[1] see `http://www.lri.fr/~filliatr/fsets/`
[2] `http://www.loria.fr/projets/calife/`
[3] `http://www.loria.fr/~ranise/haRVey/`

Another extension is the possibility to use a large subset of C programs as input for the Why tool. The Why tool has been presented at the TYPES'2003 meeting. The theoretical background appeared in [4].

Krakatoa E. Contejean, J. Duprat, J.-C. Filliâtre, C. Marché, X. Urbain and C. Paulin are working on the Krakatoa project for modelization, specification and proof of Java(Card) programs in Coq.

A tool Krakatoa [8] has been developed. It takes Java(Card) programs, annotated with JML specifications as input and produces Coq theories and Why programs from which proof obligations are automatically generated. This work is described in [9]. It has been developed as part of the VERIFICARD[4] project.

2.4 Meta-theory

The work of A. Miquel together with Benjamin Werner from INRIA site on proof irrelevance model of the Calculus of Constructions has been published in [11].

A. Miquel proposed a method for realising the proofs of Intuitionistic Zermelo-Fraenkel set theory (IZF) by strongly normalizing lambda-terms. This method relies on the introduction of a Curry-style type theory extended with specific subtyping principles, which is then used as a low-level language to interpret IZF via a representation of sets as pointed graphs inspired by Aczel's hyperset theory. As a consequence, this work refines a classical result of Myhill and Friedman by showing how a strongly normalizing lambda-term that computes a function of type $N \to N$ can be extracted from the proof of its existence in IZF. This work is published in [10].

2.5 Modules

J. Signoles applied general results on modules in Pure Type Systems to design a tool, ocamldefun[5], which unfolds functor applications in Objective Caml programs. This work is described in [13].

3 Collaborations

Math in Coq Orsay collaborates with the Nijmegen site in the domain of development of mathematical theories in Coq. Orsay organized a first meeting on december 18th and a two days workshop on April 23rd and 24th. This workshop[6] included nine presentations from researchers of Bologna, INRIA Sophia-Antipolis, Nijmegen, Orsay, Trusted Logic...
A mailing list[7] has been created for this specialized subject. P. Letouzey from Orsay collaborates with L. Cruz-Filipe, B. Spitters and F. Wiedijk from Nijmegen in order to improve efficiency of programs extracted from the proof of the Fundamental Theorem of Algebra (FTA).

[4] `http://www.verificard.org`
[5] `http://www.lri.fr/~signoles/ocamldefun/`
[6] `http://www.lri.fr/~paulin/MathinCoq/`
[7] `http://coq.inria.fr/mailman/listinfo/mathincoq`

J.-C. Filliâtre modified the `coqdoc` documentation tool[8] in order to fulfill the requirements of presentation of the FTA development.

Geometry J. Duprat organized at ENS Lyon a one-day workshop (June 11th) on Geometry and Proof on Computer.

Orsay and Lyon sites have strong interactions with the INRIA sites in Rocquencourt (now at INRIA Futurs) and Sophia-Antipolis. They participate together to common seminar and one-day workshops.

Orsay collaborates with the industrial site France Telecom R&D on proof environments for the development of critical algorithms in Telecommunication applications.

C. Paulin visited University of Warsaw for Daria Walukiewicz's PhD defense in April 2003.

A. Miquel joined the Orsay group for one year (2002-2003) after his thesis at INRIA Rocquencourt and a one year post-doc at Chalmers University.

References

[1] The Coq Proof Assistant. `http://coq.inria.fr/`.

[2] Coq development team. *The Coq Proof Assistant Reference Manual – Version V7.4*, February 2003. `http://coq.inria.fr/doc/main.html`.

[3] J.-C. Filliâtre. *Preuve de programmes impératifs en théorie des types*. Thèse de doctorat, Université Paris-Sud, July 1999.

[4] J.-C. Filliâtre. Verification of Non-Functional Programs using Interpretations in Type Theory. *Journal of Functional Programming*, 13(4):709–745, July 2003. [English translation of [3]].

[5] Jean-Christophe Filliâtre. The Why certification tool. `http://why.lri.fr/`.

[6] Herman Geuvers and Freek Wiedijk, editors. *Types for Proofs and Programs Second International Workshop, TYPES 2002, Berg en Dal, The Netherlands, April 24-28, 2002, Selected Papers*, volume 2646 of *Lecture Notes in Computer Science*. Springer, 2003.

[7] Pierre Letouzey. A new extraction for Coq. In Geuvers and Wiedijk [6].

[8] Claude Marché, Christine Paulin, and Xavier Urbain. The KRAKATOA proof tool, 2002. `http://krakatoa.lri.fr/`.

[9] Claude Marché, Christine Paulin-Mohring, and Xavier Urbain. The KRAKATOA tool for certification of JAVA/JAVACARD programs annotated in JML. *Journal of Logic and Algebraic Programming*, 2003. To appear.

[10] Alexandre Miquel. A strongly normalising Curry-Howard correspondence for IZF set theory. In *Computer Science Logic, CSL'03*, Lecture Notes in Computer Science. Springer-Verlag, 2003.

[8] `http://www.lri.fr/~filliatr/coqdoc`

[11] Alexandre Miquel and Benjamin Werner. The not so simple proof-irrelevant model of CC. In Geuvers and Wiedijk [6], pages 240–258.

[12] Nicolas Oury. Observational equivalence and program extraction in the Coq Proof Assistant. In Martin Hofmann, editor, *TLCA*, volume 2701 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2003.

[13] Julien Signoles. Calcul statique des applications de modules paramétrés. In *Journées Francophones des Langages Applicatifs*, 2003.