



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team LogiCal

Logic and Calculus

Futurs

THEME SYM

Activity
R *eport*

2004

Table of contents

1. Team	1
2. Overall Objectives	1
3. Scientific Foundations	2
3.1. Proof assistants	2
3.2. Formalisation of mathematics	2
4. Application Domains	3
5. Software	3
5.1. Coq	3
5.2. CiME	5
6. New Results	5
6.1. Development of theories and tactics	5
6.1.1. Four Color Theorem	5
6.1.2. Kepler's Conjecture	5
6.1.3. Formalization of ordinal numbers	6
6.1.4. Formalization of data structures	6
6.1.5. Air traffic control	6
6.1.6. Geometry	6
6.1.7. Proof languages	6
6.1.8. First order decision procedure with constructors	6
6.2. Development of systems	7
6.2.1. Development of Coq	7
6.2.2. Development of CiME	7
6.2.3. Integration of rewriting to Coq	7
6.2.4. Inlining of modules	7
6.2.5. Generalization of rewriting tactics	8
6.3. Studies of formalisms	8
6.3.1. Types and programming languages	8
6.3.2. Type theory	8
6.3.3. Computational interpretation of classical logic	8
6.3.4. Program extraction in classical logic	8
6.3.5. Deduction modulo	9
6.3.6. Semantics of rewriting	9
6.3.7. Rewriting modulo	9
6.3.8. Higher-order rewriting	9
6.3.9. Modularity properties of rewrite systems	9
6.3.10. Linguistics	10
6.3.11. Quantum computation	10
6.3.12. Higher-order unification and type inference	10
6.3.13. Closed reduction	10
6.3.14. Interaction nets	10
7. Contracts and Grants with Industry	11
7.1. Mao	11
7.2. Averroes	11
7.3. Modulogic	11
8. Other Grants and Activities	11
8.1. Collaboration with other teams	11
8.2. European actions	11

8.2.1.	Working Group TYPES	11
8.2.2.	Concortium MoWGLI	12
8.2.3.	Alliance project “Compiler Technology for Parallel Graph Rewriting”	12
8.3.	Other cooperations	12
8.3.1.	Maud	12
9.	Dissemination	12
9.1.	Animation of the scientific community	12
9.1.1.	Editorial charges	12
9.1.2.	Committees	12
9.1.3.	Visits	12
9.1.4.	Conferences	13
9.1.5.	Other charges	14
9.2.	Teaching	14
10.	Bibliography	14

1. Team

The LogiCal project is a common project gathering researchers from INRIA-Futurs at LIX and Laboratoire de Recherche en Informatique of University Paris XI.

Scientific leader

Gilles Dowek [École polytechnique]

Vice leader

Benjamin Werner [CR INRIA]

Administrative assistant

Catherine Moreau [TR INRIA]

INRIA staff

Bruno Barras [CR]

Hugo Herbelin [CR]

Pierre Castéran [Bordeaux]

Paris XI staff

Jean-Pierre Jouannaud [Professor at University of Paris XI]

CNRS staff

Évelyne Contejean [CR]

Post-doctorates

Claudio Sacerdoti-Coen [École polytechnique]

Ph.D. students

Pierre Corbineau [École normale supérieure]

Olivier Hermant [DGA subsidée]

Dan Hernest [LIX]

Florent Kirchner [ENS Cachan]

Sylvain Lebesne [MENRT]

Julien Narboux [ENS Cachan]

Nicolas Oury [ENS Lyon]

Clément Renard [MENRT]

François-Régis Sinot [École polytechnique]

Roland Zumkeller [École polytechnique]

Student intern

Nikhil Barthwal [INRIA]

2. Overall Objectives

Many human activities have been transformed by the invention of the computer and its broad diffusion in the second half of the XXth century. In particular, the mathematicians could have a tool allowing to carry out operations that were too long or too tedious to be executed by hand. Like the use of the telescope in astronomy, the use of the computer opened many new prospects in mathematics. One of these prospects is the use of *proof assistants*, *i.e.* computer programs which perform some operations on mathematical proofs. The goal of the research developed in the LogiCal project-team is to develop such *proof assistants*.

The main effort the project-team is the development of the **Coq** system, which has an important community of users in industry and in academia. However, we believe that the development of a proof assistant cannot be accomplished without a joint reflection about the structure of mathematical proofs and about the use of proof assistants in various applicative domains. Thus, the questions addressed in the team range from questions related to the Coq system, such as “What will be the features of the next version of Coq?”, to more theoretical

questions of logic, such as “What is a proof?” and more applied ones, such as “How can we use a proof assistant to check a protocol if free of deadlocks?”.

3. Scientific Foundations

3.1. Proof assistants

Keywords: *correctness, proof assistant, tactic language.*

The first operation that a proof assistant can perform on a proof is to check that it is correct. This participates in the quest of a new step in mathematical rigour: the point where nothing is understated, and where the reader can therefore be replaced by a program. This quest for rigour is specially important for the large proofs, either hand written or computer aided, that mathematicians have built since the middle of the XXth century. For instance, without using a proof assistant, it is quite difficult to establish the correctness of a proofs using symbolic computations on polynomials formed with hundreds of monomials, or a case analysis requiring the inspection of several hundreds of cases, or establishing that a complex object such as a long program or a complex digital circuit has some property. This quest for correctness is especially important in application domains where a malfunction may jeopardize human life, health or environment, such as transportations or computer aided surgery.

Besides this correctness check, proof assistants can help the users to build proofs interactively. The “tactic language” allowing the user to control the system in this proof construction process has always been the object of intensive studies. The ML language, for instance, was originally the tactic language of the LCF proof assistant. More recent questions about this language are focussed on the formal expression of its operational semantic, in particular the handling of exceptions.

Proof assistants may also prove some easy lemmas automatically, transform mathematical proofs into other formal objects such as programs.

A more recent kind of applications is the construction of large libraries of mathematical results on the net.

3.2. Formalisation of mathematics

Keywords: *Calculus of Constructions, constructive proofs, deduction modulo, mathematical language, predicate logic, programming language, set theory.*

A proof assistant implements a particular formalism allowing to express mathematics. A traditional formalism allowing to express mathematics is set theory, built on top of first-order predicate logic. Unfortunately, this formalism does not address exactly the needs of a proof assistant. Set theory has been elaborated at the beginning of the XXth century to study mathematically the properties of mathematical reasoning. For this purpose, being able to formalise mathematics “in principle” was enough. Nowadays, the problem is not to formalise mathematics “in principle” but to formalise them “in facts”. Thus, the design of proof assistants has led to ask new questions in logic and, in particular, in proof theory.

Several variants or alternative to set theory have been designed to express mathematics in practice. The system Coq is based on a formalism called *The Calculus of Inductive Constructions*.

An important feature for such a formalism is the language allowing to express mathematical objects such as functions and set. It is not possible to use a formalisation of mathematics that has only existence axioms, or even one having the combinator’s langage obtained by skolemizing these axioms in predicate logic. It is important to have a rich and compact language, in particular a language with binders such as the λ -calculus.

Another important feature is the ability to integrate deduction and computation. It is not possible, when we use a proof assistant to consider that the proposition $2 + 2 = 4$ requires a proof, even a proof simple enough to be found by a automated theorem proving system. Several formalisms such as Martin-Löf’s type theory, Boyer-Moore logic, the Calculus of Constructions and the Calculus of Inductive Constructions, include such a possibility to compute inside a proof. Thus, these formalisms designed to express mathematics contain a programming langage as a sublanguage.

More recently the research in this area has taken several different directions: first the study of *deduction modulo* that is the simplest extension of predicate logic allowing to mix deduction and computation. Deduction modulo has applications both in automated theorem proving and in proof theory, where it paves the way to a unified theory of cut elimination. Another direction is the design of extensions of the Calculus of Constructions with arbitrary computation rules, while the original calculus had a fixed set of rules. This extension called the *Calculus of Algebraic Constructions* may be the future formalism used in the Coq system. Finally, the need to improve the efficiency of computations in the system Coq, has lead to the use of compilation techniques issued from the theory of programming language. This has brought logical languages and programming languages closer, allowing for instance to use the language of Coq as a general purpose programming language. This perspective of unifying languages and programming languages is a real challenge for future proof assistants.

Another property of the Calculus of Inductive Constructions is important for its use as the language of a proof assistant. The first is the possibility to write both constructive and classical proofs. When a proof of existence is constructive, the user can request the computation of a witness, but, of course, not when it is classical.

By insisting on this idea that constructive proofs must be distinguished from classical proofs, the project-team LogiCal participates to rise of a new form a constructivism, not trying to restrict mathematics to constructive mathematics, but trying to identify the part of mathematics that can be done constructively and the part that cannot.

A last property of the Calculus of Inductive Constructions is that proofs are objects of the formalism, exactly as numbers, functions and sets are. This property, based on the celebrated Curry-De Bruijn-Howard correspondance, allows to reduce the safety critical base of the Coq system to a quite small kernel.

4. Application Domains

Keywords: *algorithms, mathematics, programs.*

The applications of the research of the LogiCal project-team take several directions.

The first is the applications to pure mathematics. The use of proof assistants for proving genuine mathematical theorems has been considered as utopic for long. But several recent developments have changed the situation. First of all, the development of libraries of both constructive and classical analysis has lead the possibility to use Coq, not only in remote areas of discrete mathematics, but also to prove mainstream mathematical theorem as taught in an undergrad textbook for instance. This direction culminated with the proof in Coq of the Fundamental Theorem of Algebra, a few years ago, by a group of researchers in Nijmegen. More recent work include a proof of the Four color theorem in Coq. Proofs of lemma's on polynomials used in the proof of Hale's Sphere packing theorem (Kepler's conjecture) and proofs in algebraic geometry by a group of mathematicians in Nice.

Another direction is the proof of algorithms. In proofs of algorithms (as opposed to proofs of programs) a property is proved on an algorithms formalized in the language of Coq. An example is the recent proof of algorithms used in floating point arithmetic or the older proof carried out by the compagny *Trusted Logic* of the correctness that has reached, for the first time, the EAL7 level in common criteria.

But, our main application domain is the proof of programs where an actual program written in the syntax of a general purpose programming language (such as Caml, Java or C). The system Coq is used by the ProVal project-team, that has strong historical connections to LogiCal, as a back-end of their systems Why, Krakatoa and Caduceus.

Finally, the members of the LogiCal project-team have a general culture in formal method that is not restricted to the Coq system and that they can use as consultants.

5. Software

5.1. Coq

Participants: Bruno Barras, Jean-Christophe Filliâtre, Benjamin Grégoire, Hugo Herbelin, Pierre Letouzey, Christine Paulin.

The *Coq* system, developed in the project, is a processor of mathematical proofs allowing an interactive development of specifications and proofs. The main original aspect of the *Coq* system is its formalism that includes:

- a primitive notion of mutual inductive definitions allowing high level specification either in a functional style by declaring concrete datatypes and defining functions by equations representing computations, or in a declarative style by specifying relations thanks to clauses;
- an interpretation of proofs as certified programs, implemented by the compilation of proofs as ML programs but also tools to associate a program to a specification and automatically generate proof obligations to assert its correctness;
- a primitive notion of co-inductive definitions allowing a direct representation of infinite rational data structures and build proofs upon such objects without resorting to the classical notion of bisimulation.

At the architectural level, the main features are:

- an interactive loop that allows to define mathematical and computational objects and to state lemmas,
- the interactive development of proofs thanks to a large and extendable set of tactics that decompose into elementary tactics (giving a precise control over the proof structure and thus over the underlying program) and decision or semi-decision procedures.
- a modular standard library and retrieving tools,
- a mechanism to perform partial or total evaluation of programs written within the language of *Coq*,
- a module system to manage namespaces, and featuring functors to develop parameterized development and making easier the instantiation of such functors,
- the possibility to develop evolved tactics written in the implementation language of *Coq* (namely Objective Caml), and that can be dynamically loaded and used from the toplevel,
- the isolation of the critical code performing the proof checking in a kernel small enough to reach higher levels of reliability of the whole system (with the current goal of achieving the self-validation), and the production of an abstract interface of that kernel granting that theories can only be built using the features of the kernel.

Among the most significative achievements realised using *Coq*, it worths mentioning:

- the model of authentication protocol CSET used in electronic shopping and the proof of properties of this protocol,
- the correctness proof of a compiler of the reactive language Lustre, used in the industrial setting of Scade,
- a proof of the critical kernel of the *Coq* environment,
- several models of the properties of the π -calculus,
- the development of libraries about algebra, analysis and geometry,
- a certified version of Buchberger's algorithm used in computer algebra,
- the proof of FTA theorem,
- the proof of Taylor's approximation theorem.

The *Coq* system is available from URL <http://coq.inria.fr/>. Written in Objective Caml and Camlp4, it is ported to mosts Unix architectures, but also to Windows and MacOS.

Coq is used in hundreds of sites. We have demanding users in industry (France Telecom R & D, Dassault-Aviation, Trusted Logic, Gemplus, Schlumberger-Sema, ...) in the academic world in Europe (Scotland, Netherlands, Spain, Italy, Portugal) and in France (Bordeaux, Lyon, Marseille, Nancy, Nantes, Nice, Paris, Strasbourg).

An electronic mailing list (<mailto:coq-club@pauillac.inria.fr>) fosters exchange between persons interested by the system.

5.2. CiME

Participants: Evelyne Contejean, Claude Marché, Benjamin Monate, Xavier Urbain.

CiME is a rewrite tool that allows the definition of rewrite systems, and provides tools for checking their termination. CiME is available on the web (<http://cime.lri.fr>).

The main features are the following:

- an interactive toplevel to allow naming of objects and call to various functions,
- solving Diophantine constraints over finite intervals,
- solving Presburger constraints,
- string Rewriting Systems, KB completion,
- Term Rewrite Systems, possibly with commutative or associative-commutative symbols,
- termination of TRSs using standard or dependency pairs criteria, automatic generation of termination orderings based on polynomial interpretations, including weak orderings for dependency pairs criteria,
- parameterized String Rewriting Systems confluence checking,
- TRS confluence checking, KB completion, modulo AC when needed.

6. New Results

6.1. Development of theories and tactics

6.1.1. Four Color Theorem

Participant: Benjamin Werner.

Benjamin Werner collaborated with Georges Gonthier on the proof in Coq of the Four Color Theorem. The proof of the combinatorial version of the theorem was completed in september 2004 in Coq V 7.3 and requires several hundred hours of computation for total checking.

In the process of constructing this proof, Georges Gonthier developed a promising new style of proof scripts using a new set of very compact tactics. Hugo Herbelin and Benjamin Werner are working on porting these tactics to Coq V 8 which should allow to make them widely available together with porting the four color proof to Coq V 8. In V8, using the technology of compilation, the checking time for the whole developpement is expected to drop to a few hours.

6.1.2. Kepler's Conjecture

Participants: Benjamin Werner, Roland Zumkeller.

Roland Zumkeller has started his PhD under the supervision of Benjamin Werner, investigating the possibility to formalize in Coq parts of Thomas Hales' proof (1998) of the Kepler Conjecture. This is part of the global Flyspeck project started by Hales.

They have particularly looked at how to prove inequalities over real numbers in Coq. Zumkeller has developed a library of interval arithmetic in Coq. In order to improve the performances of the package, they have starting promising discussions with computer algebra people like Eric Schost (LIX), Mohab Safey El Din (LIP6) and Jean-Pierre Merlet (INRIA Sophia-Antipolis).

The most promising of these are based on interval arithmetic with further refinements such as branch-and-bound methods and monotonicity checks done by evaluating partial derivatives. He provided an implementation of a reflectional Coq tactic with a (partial) correctness proof. As a result, in some cases the tactic is already sufficient to verify inequalities occurring in Hales' proof, in others further work needs to be done.

6.1.3. Formalization of ordinal numbers

Participant: Pierre Castéran.

Since October 2004, Pierre Castéran works on proofs of termination of complex problems. He starts developing a library on ordinals for that purpose. At present, ordinals less than ϵ_0 are represented in Cantor normal form. The main parts of the present development are a proof of well foundedness of ϵ_0 , as well as proofs that any Goodstein sequence eventually hits zero, and that every strategy is a winning strategy (in the game of Hercules against the Hydra). The last two proofs are adapted from the work of Kirby and Paris, who show they cannot be done in Peano Arithmetic. This work will continue with a development of the library on ordinals, in order to make easier proofs of termination of processes. Investigation on the representation and use of larger ordinals is planned.

6.1.4. Formalization of data structures

Participant: Pierre Corbineau.

In a paper submitted to publication, Pierre Corbineau showed an isomorphism between a functional version of *skip-lists* and a certain class of randomized binary search trees.

6.1.5. Air traffic control

Participants: Gilles Dowek, Nikhil Barthwal.

Gilles Dowek, César Muñoz and Víctor Carreño have studied an hybrid model of the air traffic concept SATS. This model permits to give a geometrical information on the spacing of aircraft. Their previous work on a discrete model of the same concept of operation has been published [26].

Nikhil Barthwal proved in Coq the correction of a synchronization algorithm of messages exchanged by aircrafts in the same airspace.

6.1.6. Geometry

Participant: Julien Narboux.

Julien Narboux has implemented in Coq a decision method for Euclidean geometry using the Ltac language. This work has been published in [27].

Julien Narboux is working on diagrammatic reasoning for geometry and more precisely on the notion of "generic sketches of a geometric configuration". This is intended to be used to reason using sketches without losing soundness. A set of sketches is said to be generic relatively to some property when if the property holds for some points on each of the generic sketches then it holds in any case.

6.1.7. Proof languages

Participant: Florent Kirchner.

Florent Kirchner has formalized a semantic framework specially adapted to the features of imperative languages, in particular proof languages. This work has been submitted to the JFLA 2005 conference.

In conjunction with César Muñoz, he has proposed a monadic representation of a proof state, and is implementing it as a library for PVS. A NASA technical report is being written that sums up this work.

Florent Kirchner has prototyped and is now implementing a meta-prover to factorize the proofs of several theorem provers.

6.1.8. First order decision procedure with constructors

Participant: Pierre Corbineau.

Pierre Corbineau worked on extending his congruence-closure tactic with the theory of free constructors, which corresponds to the semantics of Coq's inductive datatypes.

To improve the performance of the `firstorder` tactic implemented in the latest distributed version of Coq, he is currently working on a backend of his procedure based on reflection. This approach already gave encouraging results in the propositionnal case.

6.2. Development of systems

6.2.1. Development of Coq

Participants: Bruno Barras, Hugo Herbelin, Clément Renard.

Hugo Herbelin worked with Bruno Barras on the Coq version 8 project whose main feature is an entirely new more expressive and more extensible syntax. Coq version 8.0 has been released in April 2004. It has been downloaded at least more than 700 times in 6 months from the INRIA ftp server. It is also part of the Debian distribution which means that we don't have full control on its spreading. Clément Renard also took part to the development and maintenance of the Coq proof assistant.

Bruno Barras simplified unification in Coq by making code of formula unification and type inference converge. This is a preliminary work before implementing better unification algorithms such as Clément Renard's algorithm which main interest is to provide a uniform framework for the two tasks mentioned above.

Bruno Barras and Benjamin Grégoire have developed a new implementation of the `ring` tactic (which solves equations upon elements of a ring). The new implementation is more efficient thanks to a better representation of polynomials. It also extends the older tactic by only requiring the carrier set of the ring to be a setoid, instead of enforcing Leibniz equality. Another original feature is to let the user distinguish a sub-ring upon which ring operations are computable, even if the ring itself is not computable. For instance, the user can declare the integers as a sub-ring of real numbers, which allows to compute with reals that happen to be integers.

6.2.2. Development of CiME

Participant: Evelyne Contejean.

Evelyne Contejean (co-)develops the CiME rewrite tool, in which she is in charge of the matching, the unification and the completion parts (standard but also with associative-commutative symbols).

6.2.3. Integration of rewriting to Coq

Participant: Olivier Hermant.

Olivier Hermant began a work on how to integrate rewrite rules in the Coq proof assistant. Namely, he tries to understand how to compile rewrite rules toward a low-level machine language, to take advantage of the recent integration of an abstract machine in Coq, that could allow fast rewriting steps.

6.2.4. Inlining of modules

Participant: Claudio Sacerdoti-Coen.

Since a few years the state of the art of proof assistants and mathematical reasoning tools in general has made it possible to produce large libraries of formalized mathematics and complex program certifications. However, it is still unclear how to integrate in these tools mechanisms to structure and organize large developments, and to parameterize the developments over abstract theories that can be instantiated later on. Such high level structures are usually called modules.

Several authors have remarked that, since Coq V8.0, the theory of the Coq proof assistant is rich enough to define first order modules simply as depend records. This proposal targets modules made of definitions and axioms only. However, there is a clear need to study module systems that allow to declare and store also other extra-logical informations, such as parsing and pretty printing rules, tactics, realizers for axioms used for code extraction, etc. Thus a new system of modules inspired from that of the Ocaml language has been recently introduced in Coq. However, only a few developments have used it so far, and the whole standard library of the Coq proof assistant have had no benefit yet from it.

The main activity of Claudio Sacerdoti Coen consists in performing a non trivial development using the new module system: the part of the standard library of Coq that deals with arithmetic notions is being reimplemented using modules to minimize code and proof duplication and to allow to easily change the representation of the usual arithmetic data types to improve the efficiency of the operations on them.

He immediately noticed that the behavior of the main operations over modules (i.e. abstraction and instantiation) with respect to both logical and extra-logical objects was not the one expected from the user. Hence his main activity during the last six months has consisted in improving such behavior.

6.2.5. Generalization of rewriting tactics

Participant: Claudio Sacerdoti-Coen.

In parallel, Claudio Sacerdoti Coen has also studied a generalization of the rewrite tactic of the Coq proof assistants that allows the user to perform one step rewriting in non standard rewriting systems. These systems are obtained by relaxing the properties of the congruence relation that is assumed to be the underlying equality of the term rewriting system. As a consequence the notion of compatibility of an operations with the relation have been generalized to monotonicity/anti-monotonicity. The new generalized tactic has been implemented in the Coq proof assistant, and the study has been presented at the workshop “Types for Mathematics / Libraries of Formal Mathematics” held at Nijmegen the 1st and 2nd of November.

6.3. Studies of formalisms

6.3.1. Types and programming languages

Participant: Benjamin Werner.

Martin Abadi, Georges Gonthier and Benjamin Werner have given a new typed interpretation of dynamic linking through a Curry-Howard interpretation of Hilbert’s epsilon operator [13].

6.3.2. Type theory

Participant: Gilles Dowek.

Gilles Dowek has published a paper [11] analyzing the reasons why type theory is more appropriate than set theory for proof processing systems.

6.3.3. Computational interpretation of classical logic

Participants: Hugo Herbelin, Sylvain Lebesne.

Hugo Herbelin worked with Zena Ariola (University of Oregon, USA) and Amr Sabry (Indiana University, USA) on the logical foundations of computational classical logic extended with a control delimiter such as Danvy-Filinski’s *reset* or Felleisen’s *prompt* operators. They show that *reset* and *prompt* were nothing else than a regular binder of continuations (like Felleisen’s \mathcal{C} or Parigot’s μ), except that the binding name is predefined (and actually interpretable as the name of the toplevel continuation) and dynamically bound. This work has been presented at the conference ICFP 2004 [] and a journal version has been submitted to the special issue on continuation of the journal Higher-Order and Symbolic Computation.

Hugo Herbelin submitted his proof of the inconsistency of type theories which contain both sigma-types and computational classical logic (i.e. Felleisen’s \mathcal{C} or Parigot’s μ and the corresponding reduction rules).

Sylvain Lebesne started to work on subject around classical logic and type theories. In particular, he had a first look on control operator and cps translations.

6.3.4. Program extraction in classical logic

Participant: Dan Hernest.

Dan Hernest works on Program Extraction from Classical Proofs which is part of the more general project of "Proof Mining", leaded by Prof. Ulrich Kohlenbach from the Technical University of Darmstadt, Germany - a continuation of Kreisel’s “Unwinding of proofs” program. This year Dan Hernest collaborated with Ulrich Kohlenbach on our joint paper “A complexity analysis of functional interpretations”. he also maintained the Program Extraction module based on Gödel’s functional *Dialectica* interpretation in the Proof-and-Program-Extraction system MINLOG of Prof. Helmut Schwichtenberg from the University of Munich, Germany. he also wrote a draft which describes the theory behind this implementation of Gödel’s *Dialectica* in MINLOG

6.3.5. *Deduction modulo*

Participants: Gilles Dowek, Olivier Hermant, Benjamin Werner.

Gilles Dowek and Benjamin Werner [32] have given a presentation of Heyting arithmetic in Deduction modulo as a theory formed with computation rules only.

Gilles Dowek and Alexandre Miquel [31] have given a presentation of Set theory in Deduction modulo as a theory formed with computation rules only.

Olivier Hermant has studied the cut elimination property in the frame of the intuitionistic deduction modulo. He proved cut elimination for a wide range of rewrite systems, including the quantifier-free ones, the positive one, and a mix of the two previous conditions. Regarding the links between the cut elimination property and the proof normalization method, he found a rewrite system that possesses the former property, but that doesn't normalize with any method based on proof reduction.

6.3.6. *Semantics of rewriting*

Participant: Nicolas Oury.

Nicolas Oury is working on semantic of rewriting in the Calculus of Inductive Constructions. He has proved the conservativity of extensionality over a slight extension of this calculus. A report on this fact is being written. He is studying some heuristics to analyse the coverage of pattern matching, an undecidable problem in presence of dependent types.

6.3.7. *Rewriting modulo*

Participant: Evelyne Contejean.

Evelyne Contejean's main research interest is rewriting modulo (both first-order and high-order), and especially unification and matching. She modeled and certified in Coq the AC-matching algorithm of CIME.

She has collaborated with the team of José Meseguer in order to use the unification algorithm (modulo) of CIME inside the Maud system developed at Urbana-Champaign. She is currently modelling (commutative) unification for function symbols which admit a unit. The goal is to integrate the corresponding algorithm in CIME since this is needed by Maud.

6.3.8. *Higher-order rewriting*

Participant: Jean-Pierre Jouannaud.

In collaboration with Femke Van Raamsdonk and Albert Rubio, Jean-Pierre Jouannaud introduced a new comprehensive framework for higher-order rewriting, which combines aspects of earlier defined frameworks with novel ones: function symbols and variables have both a type and an arity, and rules of functional type are admitted without compromising the relationship between equational reasoning and confluence. The study of this rewrite relation is carried out first at an abstract level called normal rewriting. It is shown that the Church-Rosser properties of normal rewriting follow from its termination assumption, the joinability of irreducible higher-order critical pairs, and the presence of appropriate extended rules. These abstract results allow to capture and generalize Nipkow's approach to higher-order rewriting as well as our approach with types and arities.

In collaboration with Albert Rubio, Jean-Pierre Jouannaud studied how to extend the termination proof methods based on higher-order reduction orderings the previous framework generalized so as to admit polymorphic rules. We successively define: polymorphic higher-order algebras and rewrite rules; normal higher-order reduction orderings, which are proved adequate for proving termination of polymorphic higher-order rewriting based on higher-order matching; a variant of the higher-order recursive path ordering which we prove is a normal higher-order reduction ordering.

6.3.9. *Modularity properties of rewrite systems*

Participant: Jean-Pierre Jouannaud.

Toyama proved that the union of two confluent term-rewriting systems that share absolutely no function symbols or constants is likewise confluent, a property called modularity. The proof of this beautiful modularity result, technically based on slicing terms into an homogeneous cap and a so called alien, possibly heterogeneous substitution, was later substantially simplified by Klop, Middledorp, Toyama and de Vrijer.

In this work Jean-Pierre Jouannaud looked at a further simplification of the proof of Toyama's result for confluence, which shows that the crux of the problem lies in two different properties: a cleaning lemma, whose goal is to anticipate the application of collapsing reductions; a modularity property of ordered completion, that allows to pairwise match the caps and alien substitutions of two equivalent terms. He showed that both properties hold when considering associative-commutative rewriting in place of plain rewriting, yielding a new generalization of Toyama's theorem.

6.3.10. Linguistics

Participant: Pierre Castéran.

Pierre Castéran is working on the possible use of the Calculus of Constructions and the Coq system in the framework of computational linguistics. This work is the thema of Houda Anoun's thesis, started in Oct 2003. The first experimentations concern Michael Mortgaat's multimodal categorial grammars. These grammars are parameterized with many modes, structural rules and interaction principles. Coq is used to study the influence of these parameters and their mutual interaction on syntax analysis and computation of the semantics of a phrase. This approach is presented in the notes for ESSLLI'2004 (written with Houda Anoun and Richard Moot).

6.3.11. Quantum computation

Participant: Gilles Dowek.

Gilles Dowek and Pablo Arrighi [16] have defined a rewrite system for an algorithm transforming any term expressing a vector as a linear combination of base vectors. They have shown that this system is confluent provided the rewrite system for scalars verifies some basic properties. They have shown that the models of this rewrite system are exactly the vectorial spaces.

Building of this, they have shown [17] how this rewrite system could be the base of the seamntic of a programming language for quantum computation. In particular, they have shown that the linearity constraint of quantum physic could be expressed by restricting the beta-rule to the case where the argument is a base vector.

6.3.12. Higher-order unification and type inference

Participant: Clément Renard.

Clément Renard continued his work on higher-order unification in the Calculus of Constructions. He developed an type inference algorithm based an unification. A prototype implementation of an unification algorithm allowed to study the shape of unification problems coming from type inference and to test unifications rules specifically adapted to such equations in order to obtain ML-style type inference.

6.3.13. Closed reduction

Participant: François-Régis Sinot.

In collaboration with Maribel Fernández and Ian Mackie (King's College London), Franccois-Régis Sinot continued to develop the related notions of Closed Reduction (an interesting reduction strategy in the λ -calculus) and of Director Strings (a representation of terms with explicit substitutions and explicit annotations of variable paths). He also extended the notion to higher-order rewriting. Three journal papers are accepted for publication.

6.3.14. Interaction nets

Participant: François-Régis Sinot.

Francois-Régis Sinot developed a conservative extension of interaction nets in collaboration with Ian Mackie. This extension is of practical interest, since it allows to "program" with interaction nets in a much more natural way, and conservativity ensures that no property is lost. This work appeared as [28].

Francois-Régis Sinot continued his reflexion on strategies in the λ -calculus and interaction nets, resulting in a draft called "Call-by-name and call-by-value as token-passing interaction nets", currently submitted.

7. Contracts and Grants with Industry

7.1. Mao

MAO is an ACI (ministry grant) about developing an interface and libraries on top of Coq in order to provide support for "professional mathematicians". It gathers both computer scientists (projects Logical and Lemme) and mathematicians (Lab. Dieudonné, University of Nice). The project's homepage URL is <http://math1.unice.fr/~jpg/aci/index.htm>

7.2. Averroes

We are part of project AVERROES which started in october 2002. Labelised by National Network of Software Technologies (Réseau National des Technologies Logicielles, RNTL), it follows project Calife and have the same partners: CRIL, France Telecom R & D, INRIA, LaBRI (Bordeaux), LORIA, LRI (Orsay) and LSV (ENS Cachan). The goal of the project is to develop formal methods able to reliably check properties raising in industrial problems. It extends project Calife in not limiting to functional properties. It also studies stochastic properties and resources consumption of protocols.

7.3. Modulogic

ModuLogic is an ACI (ministry grant) about security. Its goal is to build a laboratory for the construction of certified software. Our partners are: group FOC (LIP6, CEDRIC, INRIA-Rocquencourt), project PROTHEO (LORIA) and action MIRO. It is described at URL <http://modulogic.inria.fr/>.

8. Other Grants and Activities

8.1. Collaboration with other teams

Pierre Castéran collaborated actively with Yves Bertot (project Lemme). It mainly concerns the book on Coq (maintenance of the site, adaptation to the future evolution of the system). Two new themas should reinforce this collaboration : use of Pcoq in Houda Anoun's toolkit for multimodal grammars (with Laurence Rideau and Yves Bertot), and adaptation of Baala and Bertot's approach for building recursive functions (using ordinal numbers).

Project Logical has active collaborations with other INRIA projects: Cristal, Protheo, Proval.

8.2. European actions

8.2.1. Working Group TYPES

Working Group << TYPES >> is about computer aided development of proofs and programs.

It is composed of teams from Helsinki, Chambéry, Paris, Lyon, Rocquencourt, Sophia Antipolis, Orsay, Darmstadt, Freiburg, München, Birmingham, Cambridge, Durham, Edinburgh, Manchester, London, Sheffield, Padova, Torino, Udine, Nijmegen, Utrecht, Bialystok, Warsaw, Minho, Chalmers, and also from Prover Technology, France Telecom, Nokia, Dassault-Aviation, Trusted Logic and Xerox companies.

8.2.2. *Concortium MoWGLI*

Concortium << MoWGLI >> (Mathematics on the Web, Get it by Logic and Interface) is about developing an hypertext library of mathematical theories, organised around a notation for document and mathematical formulas in XML format (OnDoc and MathML), the design of search analysis tools and the design of interfaces capable of handling theories.

It is composed of teams from Berlin, Bologne, Nijmegen, Saarbrücken, Sophia-Antipolis, and Trusted Logic company.

8.2.3. *Alliance project “Compiler Technology for Parallel Graph Rewriting”*

Francois-Régis Sinot is a member of an Alliance project (France-UK) named “Compiler Technology for Parallel Graph Rewriting” with Maribel Fernández, Ian Mackie, Jean-Pierre Jouannaud and Detlef Plump and attended three meetings of this working group (in York, Paris and London). The aim of this project is to characterise more precisely the relationship between interaction nets and graph rewriting (e.g. in the sense of the double-pushout school). A draft is currently being written.

8.3. Other cooperations

8.3.1. *Maud*

Jean-Pierre Jouannaud and Evelyne Contejean have a collaboration with José Meseguer and Mark-Olliver Stehr (University of Illinois at Urbana-Champaign), on the topic of Maud (fast prototyping type-theoretic calculi), through a contract between CNRS and Urbana-Champaign.

9. Dissemination

9.1. Animation of the scientific community

9.1.1. *Editorial charges*

Hugo Herbelin and Benjamin Werner were members of the organizing committee of the European TYPES 2004 workshop held 15-18 December in Jouy-en-Josas, France (<http://types2004.lri.fr/>). The workshop gathered about one hundred participants.

Hugo Herbelin co-organised the MoWGLI meeting with Eduardo Gimenez from the Trusted Logic company. The meeting held in Palaiseau on 14 December 2004 and gathered 10 participants.

Gilles Dowek organized a seminar on Air Traffic Control in November in École polytechnique.

Gilles Dowek organized a series of conference on mathematical games in la Cité des Sciences et de l'Industrie in Paris and gave two conferences in this serie.

Gilles Dowek has been a member of the program committee of the LICS conference.

9.1.2. *Committees*

Jean-Pierre Jouannaud has been a member of the habilitation committee of Gilles Barthe and referee in the habilitation committee of Francois Pottier.

Gilles Dowek has been a member of the committee of the Prix d'Alembert and of the Prix Anatole Decerf.

Gilles Dowek has been a member of the defence committee of the thesis of Georgi Jojgov (Eindhoven), Charles Hymanns (École polytechnique) Kuntal das Barman (Nice) and Sylvie Boldo (Lyon).

Hugo Herbelin was a referee of Silvia Likavec's PhD thesis.

9.1.3. *Visits*

Gilles Dowek have spent two month at the National Institute for Aerospace in Hampton (United-States) in April and June.

Florent Kirchner also visited the National Institute for Aerospace during the summer.

Julien Narboux visited Frédérique Guilhot and Loïc Pottier at INRIA Sophia-Antipolis.

Gilles Dowek has spent one week at the Institute of Mathematics of Beijing (China) in May.

Evelyne Contejean visited José Meseguer and his team for a week at Urbana-Champaign during the summer within the program “Rewriting calculi, logic and behavior” between the CNTS-DSTIC and the University of Illinois at Urbana-Champaign.

9.1.4. Conferences

The logical team participated to the TYPES 2004 meeting held in Jouy-en-Josas, France. Gilles Dowek gave a talk and Julien Narboux presented a poster.

Evelyne Contejean, Olivier Hermant, Florent Kirchner, Clément Renard and Franccois-Régis Sinot attended the 2nd Workshop on Coq and Rewriting at LIX (Palaiseau). Franccois-Régis Sinot presented his work on Closed Reduction and Director Strings, especially in the context of Coq (work in collaboration with Maribel Fernández and Ian Mackie) (<http://protheo.loria.fr/workshops/coq+rec04/>). Evelyne Contejean and Olivier Hermant also gave a talk.

Claudio Sacerdoti Coen and Roland Zumkeller attended the small TYPES workshop "Types for Mathematics / Libraries of Formal Mathematics" held at the Radboud University in Nijmegen, Netherlands on November 1 and 2. Roland Zumkeller gave a talk entitled “Towards a formal proof of the Kepler conjecture”

Jean-Pierre Jouannaud and Gilles Dowek participated to the conference LICS in Turku (Finlande) in July.

Gilles Dowek Jean-Pierre Jouannaud and Benjamin Werner participated to the conference ETAPS in March in Barcelona.

Evelyne Contejean and Jean-Pierre Jouannaud participated to the 15th International Conference on Rewriting Techniques and Applications (RTA'04), in Aachen, Germany, June 2004. Evelyne Contejean presented her work on formalizing an AC-matching algorithm.

Hugo Herbelin and Julien Narboux attended conferences ICFP 2004 in Snowbird, Utah, USA and TPHOLs 2004 in Park City, Utah, USA. Julien Narboux gave a presentation at TPHOLs.

Jean-Pierre Jouannaud attended ICALP 2004.

Claudio Sacerdoti Coen attended the Third International Conference on Mathematical Knowledge Management MKM2004, Bialowieza - Poland, 19-21 September.

Franccois-Régis Sinot attended the Second International Conference on Graph Transformation in Rome (<http://icgt2004.dsi.uniroma1.it/>).

Franccois-Régis Sinot attended the Second International Workshop on Term Graph Rewriting in Rome, and presented his joint work with Ian Mackie on Macros for Interaction Nets (<http://www.dcs.kcl.ac.uk/staff/maribel/TERMGRAPH.html>).

Pierre Castéran attended the workshop "La construction du savoir scientifique dans la langue" at University Pierre Mendès France in Grenoble, October 2004

Gilles Dowek participated to the workshop on rho-calculus in March in Nancy where he gave a talk.

Gilles Dowek participated to the workshop on rewriting logic and applications, in March, in Barcelona, where he gave an invited talk.

Jean-Pierre Jouannaud was invited by “alliance française” to give several seminars in India: Delhi, Bhopal, Jandigarh and Ahmedabad about Formal mathematics and application to software safety and Internet security.

Gilles Dowek has participated to the workshop on formal methods and security in Nanjing (China) in May.

Benjamin Werner was invited to give a scientific introductory talk at the european programming contest SWERC'2004.

Gilles Dowek, Sylvain Lebesne and Benjamin Werner participated to the ACI Geocal “Constructivism and program extraction” in Marseilles in November.

Gilles Dowek participated to the workshop Modulogic in Val d’Ajol, in September.

Olivier Hermant attended to an ACM meeting at Loria, Nancy in May.

Franccois-Régis Sinot discussed his work in progress aiming at generalising the notion of director strings to higher-order rewriting at a working group at PPS, Paris VII.

Olivier Hermant gave a talk at the PPS seminar, Paris in January.

Pierre Castéran gave a talk at the seminar of cryptography (University of Rennes) presenting Coq's formalism and system.

Olivier Hermant presented his work in a meeting at DGA, Paris in March.

Florent Kirchner attended a seminar on air traffic control (November 18-19th, 2004).

Gilles Dowek gave a course at the ESSLLI summer school in Nancy in July. Pierre Castéran also gave a course "Proof automation for type logical grammars", given with Richard Moot (INRIA Futurs, project team Signes) at ESSLLI 2004. The notes are available at <http://esslli2004.loria.fr>.

Julien Narboux and Olivier Hermant participated to the ICCL 2004 summer school (Proof Theory and Automated Theorem Proving) in Dresden, Germany.

Nicolas Oury and Franccois-Régis Sinot attended the Fifth International Summer School on Advanced Functional Programming in Tartu (Estonia) (<http://www.cs.ut.ee/afp04/>).

9.1.5. Other charges

Jean-Pierre Jouannaud is the leader of the LIX laboratory. He is president of AFIT, and member of "council of ETACS".

Bruno Barras has been a consultant in formal methods at Trusted Logic.

Gilles Dowek has been a consultant for the *National Institute of Aerospace* supporting the *NASA Langley's Formal Methods Group*.

Hugo Herbelin is the correspondent of the LogiCal team within the European MoWGLI project. He attended the MoWGLI meetings in Sarrbrücken, Germany and Palaiseau, France.

Florent Kirchner et Julien Narboux are the web-masters of the Coq and Logical web sites.

9.2. Teaching

Gilles Dowek is supervisor of Oliver Hermant's and Florent Kirchner's PhD theses.

Pierre Castéran participated to the direction of Houda Anoun's PhD thesis (Bordeaux) on the representation of categorial grammars in the calculus of inductive constructions (with Christian Retoré and Paul Gloess).

Pierre Castéran started in October 2004 a participation in the direction of Tran Van Truc's thesis on partial knowledge in linguistics (direction: Alain Lecomte, University Pierre Mendès France, Grenoble, and Areski Nait Abdallah University of Brest).

Bruno Barras supervised Clément Renard's PhD thesis.

Hugo Herbelin is supervisor of Julien Narboux's PhD thesis and co-supervisor with Alexandre Miquel from University Paris 7 of Sylvain Lebesne PhD's thesis that started in October 2004.

Benjamin Werner supervised Roland Zumkeller's PhD thesis.

Benjamin Werner and Bruno Barras taught a master's degree course (MPRI) "Constructive Proofs" at University Paris 7.

Benjamin Werner taught Coq and programming language semantics at the Ecole Nationale Supérieure des Techniques Avancées (Paris).

Julien Narboux taught compilation methods to fourth year student and principles of interpretation of programs to second year students at University Paris XI. He also gave practical courses of introduction to computer science to first year students at Ecole Polytechnique.

Clément Renard taught mathematics for computer science and logic to third year students at Orsay university.

10. Bibliography

Major publications by the team in recent years

- [1] B. BARRAS. *Auto-validation d'un système de preuves avec familles inductives*, Thèse de Doctorat, Université Paris 7, 1999.

- [2] Y. BERTOT, P. CASTÉLAN. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*, Texts in Theoretical Computer Science. An EATCS series, Springer Verlag, 2004.
- [3] T. COQUAND, G. HUET. *The Calculus of Constructions*, in "Information and Computation", vol. 76, 1988, p. 95-120.
- [4] J. COURANT. *A Module Calculus for Pure Type Systems*, in "TLCA'97", LNCS, Springer-Verlag, 1997, p. 112 - 128.
- [5] P.-L. CURIEN, H. HERBELIN. *The duality of computation*, in "Proceedings of the International Conference of Functional Programming 2000", LNCS, vol. 1210, Springer-Verlag, April 2000.
- [6] G. DOWEK, T. HARDIN, C. KIRCHNER. *Theorem proving modulo*, in "Journal of Automated Reasoning", vol. 31, 2003, p. 33–72.
- [7] J.-C. FILLIÂTRE. *Preuve de programmes impératifs en théorie des types*, Thèse de Doctorat, Université Paris-Sud, July 1999.
- [8] C. PAULIN-MOHRING. *Inductive Definitions in the System Coq - Rules and Properties*, in "Proceedings of the conference Typed Lambda Calculi and Applications", M. BEZEM, J.-F. GROOTE (editors), Lecture Notes in Computer Science, LIP research report 92-49, n° 664, 1993.
- [9] THE COQ DEVELOPMENT TEAM. *The Coq Proof Assistant, Reference Manual*, <http://coq.inria.fr/doc/main.html>.
- [10] B. WERNER. *Une théorie des constructions inductives*, Thèse de Doctorat, Université Paris 7, 1994.

Articles in referred journals and book chapters

- [11] G. DOWEK. *La Théorie des types et les systèmes informatiques de traitement de démonstrations mathématiques*, in "Mathématiques et Sciences Humaines", 2004.
- [12] M.-D. HERNEST, U. KOHLENBACH. *A complexity analysis of functional interpretations*, in "Theoretical Computer Science – B", to appear, 2004.

Publications in Conferences and Workshops

- [13] M. ABADI, G. GONTHIER, B. WERNER. *Choice in Dynamic Linking*, in "Proceedings of the 7th International Conference FOSSACS 2004 (ETAPS 2004)", Lecture Notes in Computer Science, vol. 2987, Springer Verlag, march 2004, p. 12-â26.
- [14] H. ANOUN, P. CASTÉLAN, R. MOOT. *Proof Automation for Type Logical Grammars*, in "ESSLLI'2004", 2004, <http://esslli2004.loria.fr>.
- [15] Z. M. ARIOLA, H. HERBELIN, A. SABRY. *A Type-Theoretic Foundation of Continuations and Prompts*, in "Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming (ICFP '04), Snowbird, Utah, September 19-21, 2004", ACM, 2004, p. 40–53.

- [16] P. ARRIGHI, G. DOWEK. *A computational definition of the notion of vectorial space*, in "Workshop on rewriting logic and applications", 2004.
- [17] P. ARRIGHI, G. DOWEK. *Operational semantics for a formal tensorial calculus*, in "International workshop on quantum programming languages", P. SELINGER (editor), Turku Centre for Computer Science General Publication, vol. 33, 2004, p. 21–38.
- [18] E. CONTEJEAN. *A certified AC matching algorithm*, in "15th International Conference on Rewriting Techniques and Applications", 2004, p. 70–84.
- [19] P. CORBINEAU. *First-order reasoning in the calculus of inductive constructions*, in "TYPES 2003 : Types for Proofs and Programs", S. BERARDI, M. COPPO, F. DAMIANI (editors), Lecture Notes in Computer Science, vol. 3085, Springer Verlag, 2004, p. 162–177.
- [20] M.-D. HERNEST. *A comparison between two techniques of program extraction from classical proofs*, in "LPAR 2002: Short Contributions and CSL 2003: Extended Posters", M. BAAZ, J. MAKOVSKY, A. VORONKOV (editors), Kurt Gödel Society's Collegium Logicum, vol. VIII, Springer Verlag, 2004, p. 99–102.
- [21] J.-P. JOUANNAUD. *Formal Mathematics: Application to software safety and Internet Security*, in "9th Artificial Intelligence Conference", November 2004.
- [22] J.-P. JOUANNAUD. *Modular Confluence modulo Associativity and Commutativity*, submitted to publication, 2004.
- [23] J.-P. JOUANNAUD. *Theorem Proving languages for verification*, in "2nd International Symposium on Automated Technology for Verification and Analysis", November 2004.
- [24] J.-P. JOUANNAUD, A. RUBIO. *Higher-order orderings for normal rewriting*, submitted to publication, 2004.
- [25] J.-P. JOUANNAUD, F. VAN RAAMSDONK, A. RUBIO. *Higher-order rewriting with types and arities*, submitted to publication, 2004.
- [26] C. MUÑOZ, G. DOWEK, V. CARREÑO. *Modeling and verification of an air traffic concept of operations*, in "ISSTA 2004", 2004, p. 175–182.
- [27] J. NARBOUX. *A Decision Procedure for Geometry in Coq*, in "Proceedings of TPHOLS'2004", S. KONRAD, B. ANNETT, G. GANESH (editors), Lecture Notes in Computer Science, vol. 3223, Springer-Verlag, 2004, <http://www.lix.polytechnique.fr/Labo/Julien.Narboux/papers/GeometryInCoqTphol04.ps.gz>.
- [28] F.-R. SINOT, I. MACKIE. *Macros for Interaction Nets: A Conservative Extension of Interaction Nets*, in "Proceedings of the 2nd Int. Workshop on Term Graph Rewriting, TERMGRAPH'04, Roma", M. FERNÁNDEZ (editor), Electronic Notes in Theoretical Computer Science, to appear, 2004.

Miscellaneous

- [29] H. ANOUN, P. CASTÉLAN. *Lambek Calculus*, contribution to the Coq System, 2004.

-
- [30] B. BARRAS, B. GRÉGOIRE. *Preservation of Typing for the Domain-Free Calculus of Inductive Constructions with Implicit Parameters*, submitted to publication, 2004.
- [31] G. DOWEK, A. MIQUEL. *A cut elimination result for set theory*, submitted to publication, 2004.
- [32] G. DOWEK, B. WERNER. *Arithmetic as a theory modulo*, submitted to publication, 2004.
- [33] H. HERBELIN, F. KIRCHNER, B. MONATE, J. NARBOUX. *Coq Version 8.0 for the Clueless*.
- [34] M.-D. HERNEST. *A Natural Deduction formulation of functional interpretations*, Draft, available in the author's home page, 2004.
- [35] J.-P. JOUANNAUD. *Extension orderings revisited*, draft, 2004.