INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team LogiCal

Logic and Computation

Futurs

THEME SYM

Table of contents

1. Team **1**
2. Overall Objectives **1**
2.1. Overall Objectives 1
3. Scientific Foundations **2**
3.1. Proof assistants 2
3.2. Formalization of mathematics 2
4. Application Domains **3**
4.1. Application Domains 3
5. Software **4**
5.1. Coq 4
5.2. CiME 5
6. New Results **6**
6.1. Development of theories and tactics 6
6.1.1. The Four Color Theorem 6
6.1.2. Hales' Theorem 6
6.1.3. Prime Numbers in Type Theory 6
6.1.4. Ordinal numbers and termination 6
6.1.5. Proofs in geometry 7
6.2. Development of systems 7
6.2.1. Coq 8.0 7
6.2.2. Universes 7
6.2.3. Modules 7
6.2.4. Fellowship 7
6.2.5. Proofs in Natural Language 7
6.2.6. The Calculus of Congruent Constructions 8
6.2.7. Decision procedures 8
6.2.8. PVS 8
6.3. Studies of formalisms 8
6.3.1. Classical Logic 8
6.3.2. Foundations of computation 9
6.3.3. Deduction modulo 9
6.3.4. Lambda-calculus modulo 9
6.3.5. Types and programming languages 9
6.3.6. Director strings and proof nets 9
6.3.7. Inductive types 10
6.3.8. Chromatic polynomials 10
6.3.9. Reflective proofs of equality 10
6.3.10. Termination 10
6.3.11. Confluence 11
7. Contracts and Grants with Industry **11**
7.1. Mao 11
7.2. Averroes 11
7.3. Modulogic 11
7.4. France Telecom 11
7.5. EADS 12
8. Other Grants and Activities **12**
8.1. Collaboration with other teams 12

8.2. European actions 12
8.2.1. Working Group TYPES 12
8.2.2. Consortium MoWGLI 12
8.2.3. Alliance project 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 13
9.1.3. Visits 13
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 [Professor École polytechnique]

Vice leader
Benjamin Werner [CR INRIA]

Administrative assistant
Catherine Moreau [TR INRIA]

INRIA staff
Bruno Barras [CR]
Hugo Herbelin [CR]
Pierre Castéran [until August 31th]

Paris XI staff
Jean-Pierre Jouannaud [Professor]

CNRS staff
Évelyne Contejean [CR, part time in ProVal]
Ian Mackie [CR, part time, starting November 1st]
Jean-Marc Notin [IR, starting December 1st]

Post-doctorates
Claudio Sacerdoti-Coen [INRIA, until May 31st]
Gyesik Lee [INRIA, starting December 1st]

Ph.D. students
Olivier Hermant [DGA, until September 30th]
Dan Hernest [France-Télécom]
Florent Kirchner [École polytechnique]
Sylvain Lebresne [MENRT]
Julien Narboux [MENRT]
Clément Renard [MENRT, until September 30th]
François-Régis Sinot [École polytechnique]
Pierre-Yves Strub [EADS]
Roland Zumkeller [MENRT]

2. Overall Objectives

2.1. 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 to 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 its correctness. This participates in the quest of a new step in mathematical rigor: the point where nothing is understated, and where the reader can therefore be replaced by a program. This quest for rigor is specially important for the large proofs, either hand written or computer aided, that mathematicians have built since the middle of the XX$^{\text{th}}$ 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 focused 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. Formalization of mathematics

Keywords: *Calculus of Constructions*, *constructive proofs*, *deduction modulo*, *mathematical language*, *predicate logic*, *programming language*, *set theory*, *type 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 XX$^{\text{th}}$ century to study mathematically the properties of mathematical reasoning. For this purpose, being able to formalize mathematics “in principle” was enough. Nowadays, the problem is not to formalize mathematics “in principle” but to formalize 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 sets. It is not possible to use a formalization of mathematics that has only existence axioms, or even one having the combinator’s language 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 language as a sub-language.

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 led 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 correspondence, allows to reduce the safety critical base of the Coq system to a quite small kernel.

4. Application Domains

4.1. 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 led 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 company *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.

5. Software

5.1. Coq

Participants: Bruno Barras, Jean-Christophe Filliâtre, Hugo Herbelin, 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 name spaces, 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 preforming 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 significant achievements realized 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 proof of the Four color theorem.

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

Coq is used in hundreds of sites. We have demanding users in industry (France Télécom 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: Évelyne 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. The Four Color Theorem

Participants: Benjamin Werner, Georges Gonthier.

Benjamin Werner has collaborated with Georges Gonthier on the proof in Coq of the Four Color Theorem.

6.1.2. Hales' Theorem

Participants: Roland Zumkeller, Benjamin Werner.

Roland Zumkeller has worked on a contribution to the formalization of Thomas Hales' proof (1998) of the Kepler conjecture.

In order to prove a list of some thousand inequalities appearing in this proof he has conducted experiments with refinements of his previously developed interval arithmetic package, such as domain reduction using partial derivative information. These have extended the scope of the tactic significantly, even though in order to attain all of the targeted inequalities further work is necessary.

A novelty in this work is the usage of constructive real numbers, that are currently formalized in Coq by Bas Spitters and Russell O'Connor (Radboud University Nijmegen). Used as a basis for interval arithmetic they allow a unified treatment of precision problems arising in the optimization of transcendental functions.

As a further extension he has recently studied the technique of Taylor models, due to Kyoko Makino (Michigan State University). Instead of computing with function values, as does interval arithmetic, the arithmetic of Taylor models computes with functions. This solves the dependency problem encountered in interval arithmetic. He has developed a prototype implementation that yields promising results.

6.1.3. Prime Numbers in Type Theory

Participants: Benjamin Werner, Benjamin Grégoire, Laurent Théry.

Benjamin Werner, Benjamin Grégoire and Laurent Théry have presented a new way to treat Pocklington primality certificates in Type Theory. This was implemented in Coq [33] and allows to prove the primality of numbers of over 1000 digits.

6.1.4. Ordinal numbers and termination

Participant: Pierre Castéran.

Pierre Castéran has worked on termination proofs and representation of ordinal numbers in the system Coq. This work presents two main aspects:

- Representation of the ordinals ϵ_0 and Γ_0 through inductive data structures : Cantor and Veblen normal forms. Both developments include decision procedure for comparison, arithmetical operations (addition, product, exponentiation), and proofs of their main properties. Proofs of termination of some system can be achieved by mapping each state to an ordinal, then using computations and/or inferences to prove that the sequence of associated ordinals is strictly decreasing. This development will be generalized to a generic notion of *ordinal system* (according to Setzer).
- In contrast with the preceding use of ordinal notations, Pierre Castéran implements the axiomatic definition of (denumerable) ordinals by Schütte. This development has two main interests:
  - It aims to be a common measure for comparing the strength of various notation systems, and "validate" these systems,
  - Schütte's presentation of ordinals uses a "classical" mathematical reasoning. Its adaptation to Coq made necessary the development of tactics for dealing with Hilbert's operator, and use of partial functions.

  At present, this development includes the proof of existence and unicity of Cantor normal form.

6.1.5. Proofs in geometry

Participants: Julien Narboux, Hugo Herbelin.

Julien Narboux has worked toward the use of the Coq proof assistant to teach mathematics. For that purpose he has designed the prototype of a graphical user interface to deal with proofs in geometry called GeoProof (http://home.gna.org/geoproof/). The software developed combines three tools : a dynamic geometry software (similar to "Cabri Geometer") to explore, measure and invent conjectures, an automatic theorem prover to check facts and the Coq proof assistant to mechanically check proofs built by the user.
He has also worked on expanding the capabilities of this prototype to deal with diagrammatic proofs in the field of abstract term rewriting.

6.2. Development of systems

6.2.1. Coq 8.0

Participants: Hugo Herbelin, Bruno Barras.

Hugo Herbelin provided support for the version 8 of Coq that was released in April 2004. Bug-fix versions were released in January 2005 and January 2006. See the coq web site for further details.

Bruno Barras coordinated the release of Coq 8.0pl3, the 3rd patch-level release of Coq 8.0. Its goal is to fix bugs of the previous release. This concerns mainly CoqIde and the extraction of programs from proofs. See ftp://ftp.inria.fr/INRIA/coq/V8.0pl3/doc/Changes.html for details.

6.2.2. Universes

Participant: Hugo Herbelin.

Hugo Herbelin wrote a note on the optimized universe stratification algorithm that he implemented in 2001 in the Coq kernel [53].

6.2.3. Modules

Participant: Claudio Sacerdoti Coen.

Claudio Sacerdoti Coen has also continued his previous work on the improvement of the handling of modules and functors by the Coq theorem prover. He has also maintained the new version of the "setoid rewrite" tactic for the Coq theorem prover he implemented in 2004.

6.2.4. Fellowship

Participants: Florent Kirchner, Claudio Sacerdoti Coen, César Muñoz, Gilles Dowek.

Florent Kirchner has implemented a proof management software, named Fellowship, to factorize the proofs of several theorem provers. This platform, based on first-order sequent calculus, also experiments with new tactic languages. Florent Kirchner has been working on the creation of a library of theorems for Fellowship, in the field of real numbers. This formalization has entailed the study of first-order set theory, and a paper on this subject is being worked upon. Florent Kirchner and Claudio Sacerdoti Coen have added lambda-bar-mu-mu-tilde proof terms to Fellowship.

6.2.5. Proofs in Natural Language

Participant: Claudio Sacerdoti Coen.

Claudio Sacerdoti Coen has studied how to provide a natural language rendering of the lambda-bar-mu-mu-tilde proof terms. The rendering rules (given as a sort of semantics) as shown very remarkable properties of the calculus when used to represent proof terms. In particular the translation rendering rules have been modeled on the corresponding translation for the lambda-calculus by Yann Coscoy. Contrary to the latter translation, that was quite involved, the rules obtained for lambda-bar-mu-tilde-mu are much easier and more structural. As a matter of fact, they are so simple that it is possible for a human to make the translation on-the-fly while look directly at the term. Moreover the proof terms are more expressive since they allow to record naturally

both bottom-up and top-down proofs. They are also richer since no implicit information must be made explicit to produce the natural language a phenomenon that happens with the lambda-calculus.

6.2.6. The Calculus of Congruent Constructions

Participants: Pierre-Yves Strub, Jean-Pierre Jouannaud.

Pierre-Yves Strub has developed, in the OCaml language, a proof checker for the *Calculus of Congruent Constructions*; and next, a proof editor (which includes a tiny subset of the tactics available in the Coq system) allowing users to develop proofs in the new calculus. In order to allow the use of the Maude2 system in other programming languages, Pierre-Yves Strub has implemented a module which drives the Maude2 system by the way of a XML protocol. An OCaml module which implements this protocol has been developed too.

6.2.7. Decision procedures

Participants: Pierre-Yves Strub, Jean-Pierre Jouannaud.

Pierre-Yves Strub has implemented in the Maude2 system (Maude is a reflective language, influenced by the OBJ3 language, supporting both equational and rewriting logic specification and programming for a wide range of applications. See http://maude.cs.uiuc.edu/) a module which implements the Shostak algorithm for combining decision procedures, for the equality in first order logics, into a general one. He also has implemented, in the same system, decision procedures for linear arithmetic and algebraic data types.

6.2.8. PVS

Participants: Florent Kirchner, Gilles Dowek, César Muñoz.

Florent Kirchner and César Muñoz, have reworked and dramatically improved the PVS library ProofLite and in particular the script enabling batch proving in PVS, proveit. He has also developed a library of proof tactics, Practicals, based on the monadic formalization of PVS's proof state, which aim at semantical clarity. A paper on this subject is being submitted to the STRATEGIES 2006 workshop.

6.3. Studies of formalisms

6.3.1. Classical Logic

Participants: Hugo Herbelin, Sylvain Lebresne, Zena Ariola, Amr Sabry, Mircea-Dan Hernest.

Hugo Herbelin continued his collaboration 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 extended a conference paper that shows that `reset` and `prompt` can be interpreted as dynamic binders of continuations. They made the connection with the original by Danvy and Filinski explicit and, incidentally, gave a normalization proof for a call-by-value variant of subtractive logic. The extended paper is submitted to the journal Higher-Order and Symbolic Computation [49].

Zena Ariola, Hugo Herbelin, and Amr Sabry also published a slight extension of an initial work by the first two authors that motivates the explicit introduction of a "toplevel continuation" in calculi with control operators (Felleisen's λ_c-calculus, Parigot's $\lambda\mu$-calculus, ...), and connects it to the logical axiom "$\bot \to A$" [14].

Hugo Herbelin also collaborated with Zena Ariola and Matthias Felleisen (Northeastern University, Massachusetts, USA) on a note showing that Felleisen's and Parigot's calculi of control are not simulable in call-by-value setting though they observationally behave the same with respect to evaluation of closed terms [48].

Hugo Herbelin has shown an inconsistency result for 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). Typically, this implies that the Set-predicative Calculus of Inductive Constructions (the type theory of Coq version 8) would become inconsistent if extended with computational classical logic. The result has been presented to the conference TLCA '05 [34].

Sylvain Lebresne has continued to work on the topic of control operator (mainly `callcc`) in the presence of dependent types and has particularly been interested in studying Σ-types in this context.

Mircea-Dan Hernest has continued his work on the complexity of programs extracted by means of Gödel's functional interpretation and its monotone variant due to Kohlenbach. His research materialized into the paper [35] which gives a new adaptation of Gödel's technique to the extraction of more efficient programs from Natural Deduction arithmetical proofs. He also finalized his joint paper with Kohlenbach on the computational complexity of the monotone Dialectica extraction algorithm, see [20]. In the end, he started to design a framework for the extraction of poly-time computable bounds by the monotone Dialectica interpretation.

6.3.2. Foundations of computation

Participants: Hugo Herbelin, Pierre-Louis Curien.

Hugo Herbelin defended an "Habilitation à diriger les recherches" diploma at the University Paris 11 in December 2005. At this occasion, he wrote a dissertation [12] on the "dualities of computation" that extended an initial work realized with Pierre-Louis Curien in 2000.

6.3.3. Deduction modulo

Participants: Olivier Hermant, Gilles Dowek, Benjamin Werner.

Olivier Hermant has studied the cut elimination property in the frame of the intuitionnistic 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. For this, he defines semantical methods base on the strengthening of Gödel's completeness theorem. Part of these results have been presented at the TLCA'05 conference. He also has proved Skolem theorem in intuitionistic logic with the help of such methods. This method can also apply to many other logical frames, such as deduction modulo. He defended and obtained his PhD degree on the 6th December.

Gilles Dowek and Benjamin Werner have proposed a formalization of arithmetic in deduction modulo and proved the cut elimination property for this presentation. A variant of this theorem also permits to prove the termination of Gödel's system T [32].

Gilles Dowek has defined a semantical strengthening of consistency, called *super-consistency* and proved that all super-consistent theories in deduction modulo had the normalization property [52].

6.3.4. Lambda-calculus modulo

Participants: Pierre-Yves Strub, Jean-Pierre Jouannaud, Fréderic Blanqui, Gilles Dowek.

Pierre-Yves Strub, Jean-Pierre Jouannaud and Frédéric Blanqui have worked on the Calculus of Congruent Constructions which replaces the conversion rule of the traditional Calculus of Constructions by a much stronger version checking whether the equality of two formulae is implied by some information collected from the context of the proof. This mechanism is indeed inspired from Shostak's combination of decision procedures, which has been proved very useful in PVS. This work is now ready for submission. Apart from a new implementation of Coq, a further step of the work includes replacing Shostak's method by a more recent one called DPLL in which the implication used by Shostak is replaced by an arbitrary propositional formula.

Gilles Dowek has proved that all functional Pure Type Systems could be embedded in the lambda-Pi-calculus modulo [51].

6.3.5. Types and programming languages

Participants: Benjamin Werner, Martin Abadi, Georges Gonthier.

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 [27].

6.3.6. Director strings and proof nets

Participants: François-Régis Sinot, Jean-Pierre Jouannaud, Ian Mackie, Maribel Fernandez.

The previous work (journal versions published this year) of François-Régis Sinot on closed reduction [17] and director strings [19], [25] has led him to work on the relationships between traditional abstract machines based on environments and interaction nets. Both are used in implementations of functional languages, but their philosophies are radically different and seem irreconcilable. François-Régis Sinot has nonetheless isolated a particular class of interaction nets, called token-passing nets, that correspond closely to environment machines. He has exhibited a correspondence for call-by-name and call-by-value in [46] and has extended the approach further in order to exhibit a correspondence with call-by-need in [26]. There is some hope that this work can be applied to improve the implementation techniques for efficient strategies like closed reduction for functional languages.

François-Régis Sinot has also worked with Maribel Fernández and Ian Mackie (King's College London) on some problems linked with the encoding of the rewriting calculus in interaction nets. This has in particular led them to introduce the framework of bigraphical nets, inspired by Milner's bigraphs, in [18]. They also have worked on efficient strategies for reduction in the λ-calculus, in particular on notions of closed reduction and implementations derived though director strings [17], [19]. Many of these ideas have either helped, or been helped by, investigations into using interaction nets for the implementation of various reduction strategies.

Interaction nets were introduced over 15 years ago. Since then they have been put forward as both a graphical programming paradigm and as an intermediate language into which we can compile other languages. Whichever way we use interaction nets, a problem has remained in that the language is very primitive. Drawing an analogy with functional programming, there is the λ-calculus but we are missing the functional programming language: syntactic sugar, language constructs, data-structures, etc. Ian Mackie has worked with François-Régis Sinot to build and extend a language for interaction nets [23], [24]. As part of this more general picture of using interaction nets as an intermediate programming language, some results have also been obtained for the encoding of data-types into interaction nets [22].

6.3.7. Inductive types

Participants: Sylvain Lebresne, Hugo Herbelin.

Sylvain Lebresne has proposed a coding of the dependent elimination schemes of inductive data types using Σ-types.

6.3.8. Chromatic polynomials

Participants: Sylvain Lebresne, Hugo Herbelin, P. Berthomé, K. Nguyên.

Sylvain Lebresne has finished a joint with P. Berthomé and K. Nguyên on the computation of chromatic polynomials. Work which has ultimately lead to [29].

6.3.9. Reflective proofs of equality

Participants: Évelyne Contejean, Pierre Corbineau.

Évelyne Contejean and Pierre Corbineau have studied reflective proofs of equality for first order terms are produced from a trace of ordered completion by CiME, and are checked by Coq. This approach has been validated on the relevant part of the TPTP library (http://www.cs.miami.edu/~tptp/). This work has been presented at the international "Conference on Automated Deduction (CADE-20)" [30].

6.3.10. Termination

Participants: Jean-Pierre Jouannaud, Albert Rubio, Évelyne Contejean, Claude Marché, A.-P. Tomas, Xavier Urbain.

Jean-Pierre Jouannaud and Albert Rubio have continued their work on the higher-order recursive path ordering for proving termination of higher-order rules that use plain pattern matching in a setting with weak higher-order polymorphic rules. This ordering includes beta- and eta-reductions, is well-founded, and polymorphic in the sense that a single comparison allows to prove termination of all monomorphic instances of the rule. The paper submitted to JACM is far more advanced that their previous version published at LICS'99 and allows to deal with most examples found in the literature. As the next step, now they need to generalize

this new version of the ordering to the type discipline of the calculus of constructions, therefore generalizing as well the work of Daria Walukiewicz.

Jean-Pierre Jouannaud and Albert Rubio have also been interested in proving termination of higher-order rules that use higher-order pattern matching in the same type setting as above. They have described a general powerful mechanism to transform an order for proving termination of higher-order rules that use plain pattern matching in an order for proving termination of higher-order rules that use higher-order pattern matching. Again, most examples found in the literature can be processed automatically. This work has not been submitted yet.

Évelyne Contejean is also interested in automatic proof of termination: a full paper written with C. Marché, A.-P. Tomas and X. Urbain has been accepted for publication in JAR [16].

6.3.11. Confluence

Participants: Jean-Pierre Jouannaud, Albert Rubio, Femke Van Raamsdonk.

Jean-Pierre Jouannaud, Albert Rubio and Femke Van Raamsdonk have worked on the problem of proving confluence of terminating higher-order rules in the same type setting again. They have described a general abstract framework called *normal rewriting* which can then be instantiated in various ways, depending on the pattern matching algorithm in use, in order to compute the appropriate critical pairs for each case. This work is currently being rewritten after an unsuccessful submission.

Jean-Pierre Jouannaud has also worked on Toyama's theorem for modular confluence, whose proof has remained complex until now. He found a new proof based on a modularity property of ordered completion which is easy to grasp and teach. He also has generalized the result to rewriting modulo an arbitrary equational theory, for all known (and yet unknown!) variations of rewriting modulo thanks to a simple remark made by Toyama. This work is submitted to the next RTA conference.

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 Marelle) 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. Labelized by National Network of Software Technologies (Réseau National des Technologies Logicielles, RNTL), it follows project Calife and have the same partners: CRIL, France Télécom 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/.

7.4. France Telecom

The project has a a three year contract with France Télécom.

7.5. EADS

The project has a a three year contract with EADS.

8. Other Grants and Activities

8.1. Collaboration with other teams

Pierre Castéran collaborated actively with Yves Bertot (project Marelle). It mainly concerns the book on Coq (maintenance of the site, adaptation to the future evolution of the system). Two new topics 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). François-Régis Sinot and Ian Mackie collaborate actively with the Theory of Computing group at King's College (London). LogiCal has also active collaborations with other INRIA projects: Marelle, 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 Télécom, Nokia, Dassault-Aviation, Trusted Logic and Xerox companies.

8.2.2. Consortium MoWGLI

Consortium "MoWGLI" (Mathematics on the Web, Get it by Logic and Interface) is about developing an hypertext library of mathematical theories, organized 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, Bologna, Nijmegen, Saarbrücken, Sophia-Antipolis, and Trusted Logic company.

The Project MoWGLI has been terminated in February.

8.2.3. Alliance project

François-Régis Sinot is involved in an Alliance project on Implementation Techniques for the Rewriting Calculus, including several people from the LogiCal INRIA Futurs project, the Protheo INRIA Loria project and the Theory of Computing group at King's College London.

8.3. Other cooperations

8.3.1. Maud

Jean-Pierre Jouannaud and Évelyne 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

Gilles Dowek has been a member of the program committee of the Workshop *Structure and Deduction* 2005. Pierre Castéran has been member of the program committee of the *Journées Francophones des Langages*

Applicatifs 2006. Hugo Herbelin has been a member of the program committee of the workshop *Classical Logic and Computing* 2005. Hugo Herbelin has been a member of the program committee of the workshop *Strategies in Automated Deduction* 2006. Hugo Herbelin has been a member of the program committee of the workshop *Programming Languages meets Program Verification* 2006. Benjamin Werner co-edited the post-proceedings of the TYPES 2004 conference. François-Régis Sinot has been the local organizer of the Second Workshop on the Rho-Calculus, at LIX, Ecole polytechnique. Jean-Pierre Jouannaud has been co-chair of the second French-Taiwanese conference in Taiwan. Jean-Pierre Jouannaud has been a co-organizer of the conference in honor of Joseph Goguen for his 65th birthday. Ian Mackie is the series editor for "Texts in Computing", College Publications; editor of the "Programming Languages Column", EATCS Bulletin. He has also acted as co-editor, Proceedings of the First Workshop on Developments in Computational Models, Electronic Notes In Theoretical Computer Science. Ian Mackie has acted as Programme co-chair for DCM: The first international workshop on Developments in Computational Models, a satellite event of ICALP, Lisbon, 2005. He as also acted as a member of the programme committee for GT-VC 2005: Graph Transformation for Verification and Concurrency, satellite workshop of CONCUR, San Francisco, 2005. He is also a member of the Steering Committee for TERMGRAPH.

9.1.2. Committees

Hugo Herbelin has been a member of the thesis committee of Silvia Likavec. Benjamin Werner has been a member of the thesis committee of Fabrice Barbier. Gilles Dowek has been a member of the thesis committee of Jesper Carlström, Benjamin Wack, Olivier Hermant, Mnacho Echenim, Fabrice Barbier and Steve Oudot.

9.1.3. Visits

Gilles Dowek has visited Ying Jiang in Beijing in April. Hugo Herbelin has visited Silvia Ghilezan in Novi Sad and Kosta Došen in Belgrad in October. Florent Kirchner has visited César Muñoz at the National Institute for Aerospace, Hampton, in September. Julien Narboux has visited Jacques Fleuriot at the University of Edinburgh. Benjamin Werner and Roland Zumkeller have visited Paul Zimmermann in Nancy. Roland Zumkeller has visited Herman Geuvers, Bas Spitters, Milad Niqui and Russell O'Connor in Nijmegen.

9.1.4. Conferences

Bruno Barras and Mircea-Dan Hernest have attended CSL 05 (Oxford), where they have both given a talk. Hugo Herbelin, Benjamin Werner, Gilles Dowek, François Régis Sinot and Olivier Hermant have attended TLCA 2005 (Nara). Hugo Herbelin and Olivier Hermant have both given a talk. Jean-Pierre Jouannaud, Hugo Herbelin, Benjamin Werner, Gilles Dowek and Olivier Hermant have attended RTA 2005 (Nara) where Benjamin Werner have given a talk and Jean-Pierre Jouannaud an invited talk for the 20 years anniversary of the conference. Julien Narboux has participated to the *7th International Conference on Technology in Mathematics teaching*, where he has given a talk and animated a workshop. Gilles Dowek has attended CADE 2005 where he has given an invited talk. Bruno Barras has participated to TPHOLs 2005 (Oxford). Florent Kirchner has attended the *Seizièmes Journées Francophones des Langages Applicatifs*, where he has given a talk. François-Régis Sinot has attended the *Seventh International Conference on Typed Lambda Calculi and Applications*, where he has given a talk. Jean-Pierre Jouannaud has attended the conference in honor of Harald Ganzinger for the first anniversary of his death, where he has given a talk. Jean-Pierre Jouannaud has attended the LPAR conference and to a collocated workshop on practical theorem provers.

Florent Kirchner, Julien Narboux and Mircea-Dan Hernest have participated to the TYPES workshop *High Level Languages for Proofs* in Chambéry, where each of them has given a talk. Julien Narboux has participated to Georgia meeting in Lyon, where he has given a talk. Roland Zumkeller has attended the seminar *Mathematics, Algorithms, Proofs*, held at Dagstuhl, where he has given a talk. Bruno Barras has participated to a working group about the convergence of tools for defining recursive functions in Coq (Sophia-Antipolis). Roland Zumkeller has participated at a meeting of the *Graduiertenkolleg Logik in der Informatik* in Venice, organized by the two universities of Munich. Roland Zumkeller has attended the TYPES workshop *Constructive analysis, types and exact real numbers* (Nijmegen), where he has given a talk. Gilles Dowek has attended the Oberwolfach meeting on Proof theory, where he has given a talk. Gilles Dowek has attended

the security workshop in Tokyo, where he has given a talk. Gilles Dowek has attended the Alliance meeting in London, where he has given a talk. Gilles Dowek has attended the meeting *Géométrie et complexité: la logique et ses images* in Paris, where he has given a talk. Gilles Dowek has attended the Q-days in Paris. Gilles Dowek has attend the meeting on Quantum computating in Lyon. François-Régis Sinot has attended the *First International Workshop on Developments in Computational Models*, where he has given a talk. François-Régis Sinot has attended the *Twelfth International Workshop on Expressiveness in Concurrency*, where he has given a talk. François-Régis Sinot has participated to the *Second Workshop on the Rho-Calculus*, where he has given a talk. Ian Mackie has attended the 17th International Workshop on Implementation and Application of Functional Languages in Dublin, where he has given a talk.

Pierre-Yves Strub, Sylvain Lebresne and Julien Narboux have attended to the Marktoberdorf'06 Summer School on *Logical Aspects of Secure Computer Systems*. Olivier Hermant and Sylvain Lebresne have attended the Types 2005 Summer School (Göteborg). Mircea-Dan Hernest has attended the EST Training Workshop 2005, where he has given a talk.

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 is consultant in formal methods at Trusted Labs, located in Versailles.

Pierre Castéran reviewed the book "Adapting Proofs as Programs: The curry-Howard Protocol" by Poernomo, Crossley and Wirsing for The Computer Journal.

Benjamin Werner has been invited to give a scientific talk to the scientific counsel of INRIA.

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

9.2. Teaching

Gilles Dowek is the thesis advisor of Olivier Hermant and Florent Kirchner. Benjamin Werner is the thesis advisor of Rolland Zumkeller. Hugo Herbelin is the thesis advisor of Julien Narboux and co-advisor of Sylvain Lebresne. Jean-Pierre Jouannaud is co-advisor of François-Régis Sinot and Pierre-Yves Strub. Pierre Castéran is co-tutoring Houda Anoun's thesis with Alain Lecomte.

Gilles Dowek has given a course at the Markoberdorf Summer School and at the TYPES Summer School.

Gilles Dowek, Bruno Barras and Évelyne Contejean have given courses in the *Master Parisien de Recherche en Informatique*.

Olivier Hermant has been a teaching assistant at École polytechnique, in May-July. Julien Narboux has been teaching assistant at the University Paris XI.

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. CASTÉRAN. *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.

Books and Monographs

[11] J.-C. FILIÂTRE, C. PAULIN-MOHRING, B. WERNER (editors). *TYPES 2004*, Lecture Notes in Computer Science, vol. 3839, Springer, 2005.

Doctoral dissertations and Habilitation theses

[12] H. HERBELIN. *C'est maintenant qu'on calcule, au cœur de la dualité*, Habilitation dissertation, Université Paris 11, 2005.

[13] O. HERMANT. *Méthodes sémantiques en Déduction Modulo*, Ph. D. Thesis, Université Paris 7, 2005.

Articles in refereed journals and book chapters

[14] Z. M. ARIOLA, H. HERBELIN, A. SABRY. *A Proof-Theoretic Foundation of Abortive Continuations*, in "Higher Order and Symbolic Computation", To appear, 2005.

[15] P. ARRIGHI, G. DOWEK. *A computational definition of the notion of vectorial space*, in "Electronic Notes in Theoretical Computer Science", vol. 117, 2005, p. 249-261.

[16] E. CONTEJEAN, C. MARCHÉ, A. P. TOMÁS, X. URBAIN. *Mechanically proving termination using polynomial interpretations*, in "Journal of Automated Reasoning", 2005, http://www.lri.fr/~marche/contejean05jar.ps.

[17] M. FERNÁNDEZ, I. MACKIE, F.-R. SINOT. *Closed reduction: explicit substitutions without alpha-conversion*, in "Mathematical Structures in Computer Science", vol. 15, n° 2, 2005, p. 343–381.

[18] M. FERNÁNDEZ, I. MACKIE, F.-R. SINOT. *Interaction Nets vs. the rho-calculus: Introducing Bigraphical Nets*, in "Electronic Notes in Theoretical Computer Science", to appear, 2005.

[19] M. FERNÁNDEZ, I. MACKIE, F.-R. SINOT. *Lambda-Calculus with Director Strings*, in "Journal of Applicable Algebra in Engineering, Communication and Computing", vol. 15, nº 6, April 2005, p. 393–437.

[20] M.–D. HERNEST, U. KOHLENBACH. *A complexity analysis of functional interpretations*, in "Theoretical Computer Science", vol. 338, nº 1–3, 2005, p. 200-246.

[21] M.-D. HERNEST, U. KOHLENBACH. *A complexity analysis of functional interpretations*, in "Theoretical Computer Science", Shortened and slightly revised version of the BRICS Technical report RS-03-12, University of Aarhus, Denmark, Feb 2003, vol. 338, nº 1-3, June 2005, p. 200–246.

[22] I. MACKIE. *An Interaction Net Implementation of Additive and Multiplicative Structures*, in "Journal of Logic and Computation", vol. 15, nº 2, April 2005, p. 219–237.

[23] I. MACKIE. *Towards a Programming Language for Interaction Nets*, in "Electronic Journal in Theoretical Computer Science", vol. 127, nº 5, May 2005, p. 133–151.

[24] F.-R. SINOT, I. MACKIE. *Macros for Interaction Nets: A Conservative Extension of Interaction Nets*, in "Electronic Journal in Theoretical Computer Science", vol. 127, nº 5, May 2005, p. 153–169.

[25] F.-R. SINOT. *Director Strings Revisited: A Generic Approach to the Efficient Representation of Free Variables in Higher-order Rewriting*, in "Journal of Logic and Computation", vol. 15, nº 2, 2005, p. 201–218.

[26] F.-R. SINOT. *Token-Passing Nets: Call-by-Need for Free*, in "Electronic Notes in Theoretical Computer Science", to appear, 2005.

Publications in Conferences and Workshops

[27] M. ABADI, G. GONTHIER, B. WERNER. *Choice in dynamic linking*, in "FoSSaCS", I. WALUKIEWICZ (editor). , Lecture Notes in Computer Science, vol. 2987, Springer, 2004.

[28] B. BARRAS, B. GRÉGOIRE. *On the role of type decorations in the calculus of inductive constructions*, in "19th International Workshop, CSL 2005, 14th Annual Conference of the EACSL, Oxford, UK", L. ONG (editor). , vol. LNCS 3634, Springer-Verlag, 2005.

[29] P. BERTHOMÉ, S. LEBRESNE, K. NGUYÊN. *Computation of Chromatic Polynomials Using Triangulations and Clique Trees*, in "Graph-Theoretic Concepts in Computer Science : 31st International Workshop (WG 2005), Metz (France)", Lecture Notes in Computer Science, vol. 3787, Springer Verlag, June 2005, p. 362-373.

[30] E. CONTEJEAN, P. CORBINEAU. *Reflecting Proofs in First-Order Logic with Equality*, in "Conference on Automated Deduction", R. NIEUWENHUIS (editor). , Lecture Notes in Artificial Intelligence, vol. 3632, Springer, 2005, p. 7–22.

[31] G. DOWEK. *What do we know when we know that a theory is consistent ?*, in "Conference on Automated Deduction", R. NIEUWENHUIS (editor). , Lecture Notes in Artificial Intelligence, vol. 3632, Springer, 2005, p. 1-6.

[32] G. DOWEK, B. WERNER. *Arithmetic as a theory modulo*, in "Term rewriting and applications", J. GIESEL (editor). , Lecture Notes in Computer Science, vol. 3467, Springer, 2005, p. 423-437.

[33] B. GRÉGOIRE, L. THÉRY, B. WERNER. *A computational approach to Pocklington certificates in Type Theory*, in "FLOPS", M. HAGIYA, P. WADLER (editors). , Lecture Notes in Computer Science, Springer, 2006.

[34] H. HERBELIN. *On the Degeneracy of Sigma-Types in Presence of Computational Classical Logic*, in "Seventh International Conference, TLCA '05, Nara, Japan. April 2005, Proceedings", P. URZYCZYN (editor). , Lecture Notes in Computer Science, vol. 3461, Springer, 2005, p. 209–220.

[35] O. HERMANT. *Semantic cut elimination in the Intuitionistic Sequent Calculus*, P. URZYCZYN (editor). , Springer-Verlag, Nara, Japan, 2005, p. 221–233.

[36] M.-D. HERNEST. *Light Functional Interpretation*, Computer Science Logic: 19th International Workshop, CSL 2005, vol. 3634, July 2005, p. 477 - 492.

[37] J.-P. JOUANNAUD. *Higher-Order rewriting: Framework, Confluence and termination*, Essays Dedicated to Jan Willem Klop on the Occasion of his 60th Birthday, Springer Verlag, 2005.

[38] J.-P. JOUANNAUD. *Twenty years later*, in "16th International Conference on Rewriting Techniques and Applications, Nara, Japan", J. GIESL (editor). , Lecture Notes in Computer Science, vol. 3467, Springer Verlag, April 2005.

[39] J.-P. JOUANNAUD, W. XU. *Automatic Complexity Analysis for Programs extracted from Coq Proofs*, in "Proc. Workshop Constructive Logic for Automated Software Engineering, Edinburgh, Great Britain", I. POERMONO (editor). , Satellite event of ETAPS 2005, April 2005.

[40] F. KIRCHNER. *Store-based Operational Semantics*, in "Seizièmes Journées Francophones des Langages Applicatifs", INRIA, 2005.

[41] C. MUÑOZ, G. DOWEK, V. CARRENO. *Provably Safe Coordinated Strategy for Distributed Conflict Resolution*, in "AIAA Guidance Navigation, and Control Conference and Exhibit", 2005.

[42] C. MUÑOZ, G. DOWEK. *Hybrid Verification of an Air Traffic Operational Concept*, in "IEEE ISoLA Workshop on Leveraging Applications of Formal Methods, Verification, and Validation", 2005.

[43] J. NARBOUX. *Toward the Use of a Proof Assistant to Teach Mathematics*, in "Proceedings of ICTMT7", 2005.

[44] C. SACERDOTI COEN. *A semi-reflexive tactic for (sub-)equational reasoning*, in "Types2004", Lecture Notes in Computer Science, vol. 3539, Springer, 2005, p. 99-115.

[45] C. SACERDOTI COEN. *Explanation in Natural language of lambda-bar-mu-mu-tilde-terms*, in "MKM2005", Lecture Notes in Artificial Intelligence, vol. 3863, Springer, 2005, p. 234-249.

[46] F.-R. SINOT. *Call-by-Name and Call-by-Value as Token-Passing Interaction Nets*, in "Proceedings of Typed Lambda Calculi and Applications (TLCA'05)", Lecture Notes in Computer Science, vol. 3461, 2005, p. 386–400.

[47] B. WERNER. *La Vérité et la Machine*, in "Image des Maths 2005", E. GHYS, J. ISTAS (editors). , Société Mathématique de France.

Miscellaneous

[48] Z. M. ARIOLA, M. FELLEISEN, H. HERBELIN. *Control Reduction Theories*, Submitted, 2005.

[49] Z. M. ARIOLA, H. HERBELIN, A. SABRY. *A Type-Theoretic Foundation of Delimited Continuations*, Submitted, 2005.

[50] P. CASTÉRAN. *Ordinal in Cantor Normal Form*, 2005, Contributions to the system Coq.

[51] G. DOWEK. *Embedding Pure Type Systems in lambda-Pi-calculus modulo*, 2005, manuscript.

[52] G. DOWEK. *Truth values algebras and normalization*, 2005, manuscript.

[53] H. HERBELIN. *Type inference with algebraic universes in the Calculus of Inductive Constructions*, Manuscript, 2005.

[54] J.-P. JOUANNAUD, J. GOUBAULT-LARRECQ. *Finite Semantic Trees Suffice for Ordered Resolution and ParamodulationProceedings of the Workshop on Programming Logics in memory of Harald Ganzinger*, 2005.

[55] J.-P. JOUANNAUD, A. RUBIO. *Polymorphic Higher-Order Recursive Path Ordering*, 2005.

[56] C. MUÑOZ, R. SIMINICEANU, V. CARREÑO, G. DOWEK. *KB3D Reference Manual - Version 1.a*, NASA/TM-2005-213769, 2005.