It is our thesis that formal elegance is a prerequisite to efficient implementation.
– Gérard Huet [4]

We, the communities behind the RTA [1] and TLCA [2] conferences, believe that our field has evolved and developed richer connections with many both practical and theoretical aspects of computer science and logic research since the inception of RTA in 1983 and TLCA in 1993. In particular, the scope of the two original conferences widened to include a significant overlap, and in fact the conferences have already collaborated by having most of our meetings since 2003 as the joint RDP [3] conference.

We have therefore decided to propose a new conference,

Formal Structures
for
Computation and Deduction
(FSCD)

which not only combines our scope but further extends it to cover all the inter-related formal areas that researchers in formal structures for computation and deduction engage in.

The name of the new conference comes from an unpublished but important book by Gérard Huet [4] that was a strong influence on many researchers in our area. We are grateful to Gérard for allowing us to reuse the name.

The extended scope of the conference will include all research related to formal structures for computation and deduction, in particular all areas/categories included in the attached non-exhaustive list of topics.

We look very much forward to serve the scientific community with this new conference, which inherits as well as updates and modernizes the scope of the conferences it replaces.

References.

[1] `http://rewriting.loria.fr/rta/`

[2] `http://www.mimuw.edu.pl/tlca/`

[3] `http://users.dsic.upv.es/~rdp03/`

[4] `http://pauillac.inria.fr/~huet/PUBLIC/Formal_Structures.ps.gz`

Topics. FSCD initial non-exhaustive list of topics (intended to extend the current RTA and TLCA scope, and expected to evolve over time):

1. Calculi
   (a) Lambda-calculus
   (b) Rewriting formats (string, term, higher-order, graph, conditional, ...)
   (c) Proof theory (natural deduction, sequent calculi, proof nets, ...)
   (d) Strategies in computation and deduction
2. Type Theory and Logical Frameworks
   (a) Type systems (recursive, intersection types, polymorphism, ...)
   (b) Dependent types and homotopy type theory
   (c) Linear logic and other constructive logics
   (d) Implicit complexity
3. Fundamentals of Functional and Declarative Programming
   (a) Unification and narrowing
   (b) Tree automata
   (c) Continuations and control operators
   (d) Coinduction and infinitary systems
4. Semantics
   (a) Abstract machines
   (b) Categorical semantics
   (c) Denotational and game semantics
   (d) Quantitative models (timing, probabilities)
5. Algorithmic Analysis of Formal Systems
   (a) Type inference and type checking
   (b) Complexity analysis
   (c) Checking termination, confluence, and related properties
   (d) Formalisation and certification
6. Tools and Applications
   (a) Proof assistants and interactive theorem proving
   (b) Automated deduction (completion, constraints, equational logic...)
   (c) Symbolic computation
   (d) Implementation techniques for formal systems
   (e) Case studies and applications based on formal systems