*Invited Talk*

# Automating the Verification of Floating-point Algorithms

Guillaume Melquiond

Inria Saclay – Île-de-France & LRI, CNRS UMR 8623, Université Paris-Sud
Centre Universitaire d'Orsay, Bâtiment 650 (PCRI), Orsay, F-91405

Floating-point numbers are limited both in range and in precision, yet they are widely used as a way to implement computations on real numbers. Thus arithmetic operations introduce small errors which might be amplified during subsequent computations and cause inaccuracies. As such, proving the correctness of a floating-point algorithm usually entails verifying that the computed results are still close enough to some ideal values, despite the method error and the round-off errors. The traditional way to tackle such a verification is to perform an error analysis, possibly using automated tools.

Unfortunately, when it comes to the low-level functions found in mathematical libraries, the floating-point code is usually so contrived that this approach falls short. Indeed, just knowing the code is no longer sufficient to verify it, one also has to know the mathematical reasons that led to choosing this code in the first place. This excludes any hope of full automation, yet automated tools are sorely needed, if only because performing a pen-and-paper proof of such functions is long, tedious, and error-prone.

This talk will show some issues specific to the verification of the floating-point functions of a mathematical library, and some methods for solving them automatically. These methods will be exemplified using Gappa, a tool dedicated to proving the logical formulas that arise during the verification of small yet complicated floating-point algorithms. This tool is based on interval arithmetic, expression rewriting, and theorem saturation. For increased confidence, the tool also generates formal proofs which can be verified by the Coq proof assistant.