Bringing Theorem Proving to the (sonic) Masses

[Joint work with P. Jouvelot, B. Pin]

Congrès "Des outils et des méthodes innovantes pour l'enseignement de la musique et du traitement du signal"
St-Etienne (France)
November 2-3, 2015

Abstract

We explore the intersection of interactive theorem proving and digital signal processing through the use of web-based, rich interfaces. Traditionally, the barrier to entry to interactive theorem proving has been high. Provers are complex systems using obscure programming languages, and libraries may be under-documented and use formalisms and notations far from the standard domain-specific practice. Thus, it doesn't come at a surprise that interactive theorem proving has seldom been explored in the the digital signal processing community.

In previous work [1,2], we formalized several DSP tools and concepts using the Coq theorem prover [3]. [1] presents a simplified model Faust [4], a programming language tailored to audio DSP. The formalization allows to reason about Faust programs and prove typical properties like filter stability. In [2], we mechanized some theory of the Discrete Fourier Transform, following [5], and proving the main theorems. In our opinion, both developments are suitable as teaching material, either as an introduction to theorem proving or to students interested in DSP theory.

However, in their current form, the student or reader must work with two versions of the document. The paper-and-ink version: which misses the interactive component of the theorem proving process, and the Coq proof scripts, that being plain text files usually miss convenient formatting and structure. This problem is also common in other teaching material (see for instance [6]), and while efforts to improve this situation have been made ([7] is one of the latest), the situation is still far from optimal, as most initiatives have been limited by implementation and platform choices.

Our take on the problem --- inspired by the long tradition of interactive documents found in the computer music domain --- is to enrich the Coq document model to the full power of HTML5. Our goal is to have an integral solution, free of external components, so we have ported Coq to javascript, and we are writing a custom IDE for enriched proof documents.

Thus, the once painful Coq installation and setup has become now a page load in the browser, document navigation and interaction can take advantage of browser support, and our documents can take advantage of most available JS libraries for extensibility. For instance, our model would allow to incorporate a javascript-based score like [8] and relate its output to a particular Coq datatype in a quite straightforward way.

The primary design goal of the web-based user interface is supporting education and interactive papers, focusing on supporting excellent user feedback, comprehensive documentation search, and good notational/navigational facilities.

The current prototype can be accessed at [9]. It supports full Coq

functionality, but the user interface is still in heavy development. While support for the enriched document model won't likely be available at the time of this submission, the main technical difficulties of the development have been solved, thus we are confident on getting usable versions in the workshop timeframe.

References:

[1] A Taste of Sound Reasoning, Emilio J. Gallego Arias, Olivier Hermant, Pierre Jouvelot. 13th Linux Audio Conference. Mainz, April 9-12th, 2015

[2] Adventures in the not-so-complex Space, Emilio J. Gallego Arias, Jouvelot. 7th Coq Workshop, Nice, June 26th, 2015

[3] The Coq Theorem Prover (http://coq.inria.fr)

[4] The Faust Programming Language (http://faust.grame.fr)

[5] The Mathematics Of The Discrete Fourier Transform (DFT) With Audio Applications, 2nd Edition, Julius O. Smith III, Stanford University

[6] B.C. Pierce et al. Software Foundations (http://www.cis.upenn.edu/~bcpierce/sf/)

[7] Edukera (http://www.edukera.com/)

[8] http://www.vexflow.com/

[9] The latest (experimental) release can be found at https://x80.org/rhino-coq/

Bio:

Emilio J. Gallego Arias is a long-time functional programmer and amateur musician. He has been wondering for a while how audio programming should be.

He currently works on the formal semantics of the Faust audio programming language at MINES ParisTech, PSL Research University.

In the past, he has worked on privacy, security, software verification and semantics, received a PhD from the Technical University of Madrid and was a postdoctoral researcher at the University of Pennsylvania.