November 3, 2011

Report on the PhD thesis “Generic Proof Tools and Finite Group Theory” by François Garillot

Moving the mathematical knowledge to certified computer libraries is an exciting challenge for the forthcoming decades and the current thesis exactly comes within the scope of this endeavour, in this case in relation with the Mathematical Components project started by Georges Gonthier and aiming at fully formalising the Feit-Thompson theorem (classification of finite groups of odd order) in the Coq proof assistant and seizing this opportunity to develop scalable formalisation techniques applicable to the design of reusable mathematical libraries.

Several of these formalisation techniques rely on a feature of Coq called *Canonical Structures* and the first chapter of the thesis provides with a comprehensive and thorough description of many (if not all) development techniques and tricks related to canonical structures that are used in the Mathematical Components libraries[1]. Starting with a summary of the main features of the type theory Coq is based on, the chapter goes on with a comparison of the most standard ways to formalise mathematical structures on computer (modules, but above all so-called pebble-style records, i.e. type classes, and telescopic-style records) before diving into the substance of the Mathematical Components techniques. The primary components here are “packed classes”, a concept introduced by François Garillot as part of a collaboration with other contributors to the Mathematical Components library and which highlights in a given mathematical structure (e.g. lattices) the other structures it inherits (here partial order and underlying equality) while still putting a special emphasis on the parameters of the structure (here the domain, but virtually also the operations of the structure) so as to optimally factorise the dependencies in these parameters. Then can the play with canonical structures begin and François Garillot gives numerous (fascinating for the novice I am) generic applications, covering e.g. multiple inheritance, overlapping instances, extensible types with extensible operations on them (the so-called expression problem), including also ad hoc (pragmatic but ugly[2]) tricks aiming at bypassing specific idiosyncrasies of Coq. A noticeable point of all the story is that the techniques presented

[1] These libraries are also called Ssreflect libraries from the name of the package bundling these libraries with a Coq extension named Ssreflect. This latter name itself comes from the name of the primary formalisation technique used to develop the libraries: small scale reflection, especially reflection of decidable propositions into two-valued computable functions.

[2] I'm mainly thinking here to the mirrored duplication of fields in records so as to speed

allow at the end to support a kind of formalisation very close to the one found in mathematical writings. Also, the study is such extensive that I recommend the reading of the thesis to anyone interested in a comprehensive presentation of what can be done with canonical structures, if ever a table of contents and an index to the main notions introduced could be added so that the minimal context needed to read specific sections of the chapter in isolation could be easily accessed.

While the first chapter was about generic formalisation techniques based on canonical structures, the second chapter investigates specific implementations of these techniques achieved by François Garillot. This covers issues with the representation of isomorphisms, partial functions, cyclic groups, automorphisms groups, with a new and short proof of correctness of the RSA encryption algorithm as an application.

The third chapter is more about pure mathematics and focuses on a study of the mathematical meta-properties of some classes of groups and of some classes of functions for defining subgroups. With this respect, it connects to a long history of works about classifying mathematical properties (e.g. based on category theory), suggesting in passing that the meta-study of properties about group theory is actually not as developed as it could be. Nevertheless, the contribution also takes place at the level of the formalisation techniques since being able to extract the true formal substance of a notion as François Garillot does allows to mechanically factorise the proofs involving the notion, as the striking example of the reduction of the "charactericity" proofs for the sixteen kinds of subgroups defined in the Mathematical Components library to a couple of meta-properties about "subgroup-defining functions" shows.

All in all, I found this thesis to be a remarkable piece of work first narrating the most up-to-date developments in formalising mathematics and secondly improving, scaling and exploiting these techniques into highly technical applications at the frontier of computer science and mathematics. I strongly recommend it to be defended.

Hugo Herbelin
INRIA Rocquencourt - Paris

up the syntactic comparison of fields whatever the comparison is performed left-to-right or right-to-left. I'm less sceptical about another trick used to artificially equip functions with extra information relevant for type inference, namely phantom types.