François Garillot
Generic Proof Tools and Finite Group Theory

a report by Conor McBride

October 31, 2011

1 Overview

Garillot's thesis represents a substantial contribution to formalized mathematics in its development of finte group theory, but is at least as strong a contribution to the study of tools and techniques for large scale proof engineering. The development is mediated via the Coq system, and while the author has clearly made a considerable study of Coq's quirks and subtleties, the better to play his games of automation, he does not lose sight of the point: he argues cogently for the engineering principles he uses in his development, whatever the tricks he must play to deliver them. He addresses key isses of structure management which any designer of proof systems and libraries must face, working at a scale where the cost of exponentially bad representation choices does become prohibitive and only the scalable can serve at all.

The thesis begins with a compact and helpful introduction to the Calculus of Inductive Constructions by incremental augmentation of Pure Type Systems. From there, we see a careful analysis of the representation of mathematical structures as records carrying sets, their operations, and proof of relevant structural properties. The rest of the first chapter deals extensively with mechanisms outside the Coq kernel which support abbreviation, and particularly with 'canonical structures'. Garillot develops his 'packed classes' methodology for modelling concept hierarchies and demonstrates the principles of its usage on small examples.

The second chapter introduces the techniques Garillot exploits to manage developments in finite group theory, dividing labour efficiently between human and machine, favouring reflection (using the computational power of the kernel) over tactics (computing outside the kernel). He takes considerable advantage of the domain's finiteness to work with certified Boolean decision procedures wherever possible, exploiting the proof irrelevance which naturally accompanies propositions reflected as Boolean values. He demonstrates the effectiveness of these techniques by giving a pleasingly algebraic development of the Chinese Remainder Theorem, leading to a delightfully compact verification of RSA public key encryption.

The third chapter takes the algebraic approach still further, developing a toolkit for subgroup computations which are characteristic *by construction*. This compositional method hinges on recognizing that characteristicity can be seen as functoriality, baking preservation of structure into the operations on sets. This is Mathematics conducted with the engineering discipline of Computer Science, leading to an adventurous formalization of torsion theories for groups. This investigation of "by construction" reasoning finishes with a speculative section on the possibility of exploiting parametricity in a reflective way, indicating a strong and potentially valuable strand of future work.

2 Detailed Remarks

Which design choices result from Coq pragmatics; which are fundamental? I certainly make no dispute with the choice of Coq for this project nor with Garillot's methods of proof engineering in Coq. However, to maximize his contribution to proof developers whatever the system, and to designers of systems including future versions of Coq, he should try to clarify to what extent his adopted techniques are Coq-specific.

For example, the trade-off between Pebble-style and telescope-style records is clearly influenced by the way Coq proof terms require repetition of record parameters in both introduction and elimination forms: a 'bidirectional' type discipline, checking rather than synthesizing types for introduction forms, might alleviate some of this pain. If Coq were to adopt such an approach (as Agda does) what would change? It might be the case that the 'packed classes' compromise would continue to be an excellent pragmatic choice.

Similarly, Garillot's machinery currently rests on aspects of Coq's current unification algorithm that might seem unfortunate more broadly. Firstly, the canonical structures mechanism relies on selecting unifiers which are not most general, making unforced 'default' choices. Secondly, the strictly depth-first strategy makes unification extremely sensitive to the order in which constraints arise: the disagreement set $\{\text{carrier}\ ?R = \mathbf{bool},\ ?R = \mathit{MyGroup}\}$ might fail if tackled left-to-right and the canonical structure with carrier **bool** is other than $\mathit{MyGroup}$. Right-to-left, it is but a matter of checking that $\text{carrier}\ \mathit{MyGroup} = \mathbf{bool}$. Thirdly, whilst avoiding δ-steps in unification is prudent, crucial reliance is placed on the prioritization of canonical structure inference over δ-reduction. The treatment in section 1.2.4, while noting the efficiency value of δ-delaying, gives no hint that a crucial technique—prioritised search for canonical structures—will be made to work by regarding equal things as different. I am pleased to find Garillot's case for inference of relevant structure by programmable search compelling, which is why I should like to see a clearer separation of the required functionality from its local implementation by quirk of unification.

Can the design methodology be made systematic? Garillot has demonstrated the clear success of his methodology for managing hierarchies of mathematical structures, so much so that this enquiring reader wonders if that methodology could benefit from notational support. At the very least, I should like to know which parts of developments like the lattice hierarchy in figure 1.31 represent the user's design choices and which are purely mechanical consequences of following Garillot's method. A better notation would focus on the former and leave the latter to an elaboration mechanism.

What is art, and what is engineering? The setup of group theory in section 2.2 is a little delicate. In particular, Garillot takes a lot of care with the choice of how tightly to circumscribe the underlying carrier type of groups, rather than using the set-characterizing predicates to cut down larger carriers. Clearly, more uniformity in carriers eases the interchangeability of data, but junk in carrier types makes it harder to define functions meaningfully. We see domain noise being mapped in one case to the unit element of a group, and in another case to itself. It seems clear that the development is rather sensitive to these choices and that they have been rather artfully conceived. Whilst the pragmatics of this particular problem instance are very well treated, I am left wondering what the transferrable lessons might be. I should like to see some broader reflection on this sort of situation.

A related situation is that the group morphisms are defined as Coq functions, despite the fact their finite domain lends themselves to the 'lookup table' approach developed earler. It seems to me that Coq function composition satisfies identity and associativity abstractly at the definitional level, which somehow is more valu-

able than the decidability of extensional equality for concrete functions. What is the engineering lesson that we learn from this choice?

The reader deserves more pity, at times. Chapter three left me struggling to recall lost undergraduate memories. There are perhaps a few incidents where an extra clarifying remark would have helped me out of my perplexity. A case in point is in section 3.1.2, where $H^\phi = H$ is 'trivially' replaced by $H^\phi \subseteq H$. It took me a while to see why it is trivial: crucially ϕ is injective and H is finite. Further along, I found it a struggle to visualize what the 'upper product' actually means. I drew myself a diagram, factoring G into $F_1(G)$ and its cosets. Perhaps a little presentational effort here would help the reader through this crucial part of the setup. What follows is a masterstroke of compositionality, if only the reader can reach the point of comprehending it.

Minor remarks. As with many draft theses, Garillot's needs a little more editorial care and attention than it has yet received. Some concepts, notations and methods are used long before their introduction, e.g. the use of "CoInductive" to avoid the generation of trivial recursors, and the `{morph ... / ...}` construct. I also noted 'list' in some places and 'seq' in others. For the most part, citation style is agreeable, but the preamble to chapter 2 has several instances of author-as-thing ('operations already touched upon in Bertot'), and even parenthetic-citation-as-noun ('finite structures described in (...)'): these can and should be finessed.

3 Verdict

I am highly impressed by Garillot's thesis, and while I do have a number of comments, questions, additions and minor repairs to suggest, I find nothing seriously objectionable or controversial that would prevent him from advancing to its defence. I shall separately post my marked up copy to Garillot to assist his final adjustments, and I am confident that the document will be in satisfactory condition on the appointed date of 5 December. I am happy to give my approval to this doctoral work.

DR CONOR T McBRIDE
UNIVERSITY OF STRATHCLYDE
OCTOBER 31 2016