Using AI to aid automation of proof search in Formal Methods
Cliff B. Jones, Alan Bundy, Andrew Ireland, Gudmund Grov, Leo Freitas, Andrius Velykis & Michael Butler

Overview & motivation
Most formal methods give rise to proof obligations (POs).

Even though two POs have a similar proof pattern or are of “similar form”, it does not necessarily mean that they have the same proof, e.g.: different intermediate lemmas are required; different instantiations of quantifiers; different data structures are used. Thus, a high-level representation is required in order to capture such variations, and AI4FM’s research hypothesis is:

Research hypothesis: we believe that it is possible (to devise a high-level strategy language for proofs and) to extract strategies from successful proofs that will facilitate automatic proofs of similar POs.

The key to achieving this is classification of the POs – and in AI4FM we envisage several dimensions: the data structures; the shape of the PO; and the information from the PO generator. These will be analysed separately. Then using the information from the PO shape and PO generator (POG), the proof obligations can be classified into “families”.

Strategy language
A key challenge is to construct a strategy language. The process of designing it will be both empirical and iterative. We have identified some requirements, including (see [2, 3] for details):

- LCF-style tactics are overly operational and sequential. We believe a non-sequential and more declarative language will be more robust to changes, and will capture larger families of “similar” POs.
- We see many dimensions of informations. The strategy on the left abstracts over domain and data-type information, and we expect to discover additional dimensions.
- Rippling [1] provides evidence for a high-level strategy language, and we hope to adapt many rippling items, such as lemma discovery, induction revision and generalisation.
- The language should be able to describe the search space, such that “dead-ends” of the exemplar proof can be captured in (and ruled out by) the high-level strategy.
- An approach, used in ACL2, is to reduce a real problem into a toy problem, and use the toy problem to find a good strategy for the real problem. We would like to support this.

Other key challenges are automatic abstractions of an exemplar proof into a strategy – and to find out which strategy would solve a given problem. We will target model-based top-down formal methods (e.g. VDM, Z, B and Event-B). Here, specifications often change during development, and we believe high-level strategies will be more robust to changes than low-level proofs. We envisage a flexible style of interaction as in “mural” [4].

References

Resources
This work has been supported by EPSRC grants EP/H024050/1, EP/H024204/1, and EP/H023852/1. Please see [2, 3] for more information on AI4FM. Internet resources:

- Webpage: www.ai4fm.org
- Mailing list: ai4fm-info@jiscmail.ac.uk