New Directions in Automated Conjecture Making

Alan Bundy*, Simon Colton**, Sophie Huczynska* and Roy McCasland*
*Division of Informatics, University of Edinburgh, UK
**Department of Computing, Imperial College London, UK

The HR system has been developed to form scientific theories automatically, with special application to mathematical conjecture making [1]. Two projects with HR are currently coming to an end and within both, there have been recent studies of the potential application of HR to research mathematics. These studies have led to some criticisms about the current limitations of HR and suggestions for improvements. Furthermore, a new project with a goal of extending HR in order to test its application to an area of research mathematics is about to start. Hence, now is a ideal time to look at these criticisms and recommendations and suggest directions for the future development of the system, which we do here. To appreciate the recommendations, we first give a brief overview of the theory formation HR undertakes.

1 Automated Conjecture Making

HR is provided with a background theory of concepts, each given with a definition and set of examples. Alternatively, it can be supplied with a set of axioms (for instance, from group theory), from which it (a) extracts the background concepts itself and (b) uses the MACE model generator [6] to generate the initial examples. HR builds a theory by inventing new concepts using a set of 10 production rules. After the introduction of a new concept, an empirical check is undertaken to see if it can be related to concepts already in the theory, which may lead to a non-existence, implication or equivalence conjecture [2]. HR uses the Otter theorem prover [5] to attempt to prove these conjectures and MACE to find counterexamples to false conjectures. HR orders its concepts and conjectures in terms of measures of interestingness in order to fuel a heuristic search [4]. Recent improvements to HR in order to make it more accessible to mathematicians include (i) the use of Otter as a *filter*, so that any results proved by Otter are withheld from the user [3], which can improve the quality of the theorems presented (which has been especially useful in number theory) (ii) an ability for HR to read Maple code, so that background concepts can be input in Maple format, a computer algebra package (CAS) used increasingly by mathematicians and (iii) allowing the user to step in as a theorem prover/counterexample generator in order to supply more information about HR's conjectures at run-time.

2 Limitations and Suggestions for Improvement

Two separate explorations by research mathematicians (McCasland and Huczynska) were undertaken into the potential for HR in research mathematics. Both concluded that, while HR has some unique qualities which may have potential, the system as it stands is not ready for use in research mathematics (although Huczynska concluded that the conjectures produced by the system were of a standard which may apply to the setting of number theory tutorial questions and to recreational mathematics). The following suggestions for improvement were outlined as part of these explorations:

- Iterative deepening theory formation using sophisticated background knowledge

HR grew from an AI investigation into how to build a mathematical theory from the bare minimum of information – they axioms of a domain. Furthermore, we had assumed that we could build up domains using HR to slowly but surely re-invent and re-discover the important concepts and conjectures in successively more difficult domains using a pyramidal approach. We have realised that this approach may be limited in its application to research mathematics. This is partly because of limitations of HR and the mathematical systems it relies upon, and partly because (as with many machine learning applications to scientific discovery), it would be better to enable the user to specify higher level mathematical concepts – which they are interested in – and to get HR to make discoveries about those concepts, than to build theories from first principles.

As mentioned above, the user is now able to supply Maple code to HR, which greatly enhances the range of background information which can be supplied. We have similarly implemented an interface to Gap

CAS and are doing likewise for the Macauley2 CAS. In the end, we hope to link HR to all of these systems via the MathWeb system. Moreover, in the domain of application looked at by McCasland – Zariski spaces – we are building up a knowledge base of higher level concepts such as ideals, varieties and topologies, in order to increase the sophistication of the background information with which HR is supplied. In addition, in the new project, we will look at getting HR to work with infinite examples to further extend the range of theories that HR can work with.

The studies into HR's application to research maths suggest that the manner in which HR is given information and the way it forms theories should be much more controlled, designed to follow the 'little theories' approach advocated by Guttman and Farmer. This in turn suggests we drop another assumption: that HR should be run until it finds concepts of a sufficient complexity. In the past, we have run HR until the concepts it forms are fairly complicated with respect to the concepts it was supplied with. Because the background concepts were the simplest in the domain, HR's constructions were still fairly easy to understand. However, we have found that similar convoluted constructions when the background concepts are already fairly sophisticated concepts, have limited appeal, because they may be over-specialised. In overview, the two investigations have led us to adopt a new mode of operation for HR, which we can possibly characterise as an iterative deepening search. In this mode, the user supplies many fairly sophisticated concepts as background for HR. HR then performs successive short theory formation involving increasingly larger numbers of these concepts, aiming to find simple relationships between well known concepts, rather than attempting to invent complicated concepts itself.

- Proof planning for theory formation

HR's theory formation could be characterised as explorative (inventive) and inductive. The deduction carried out by the system is limited to theorem proving when a conjecture has been made inductively. A particular limitation of HR is that it barely refers to the axioms of the domain other than to pass them to Otter (note the exception: HR's 'embed algebra' production rule uses the axioms to identify algebras embedded within other concepts). We intend to enable HR to use deduction to form theories by deducing facts which must follow from the theorems discovered. This process will be controlled by the λClam proof planner, with the proof plans collected from the mathematical literature.

- User-defined production rules

A final upgrade proposed to HR is to enable the user to define production rules for the invention of new concepts (from which conjectures would follow). In particular, in number theory, it has been suggested that a production rule which enables Dirichlet convolution of two arithmetic functions would make HR's theory formation richer and possibly more appealing to number theorists. It appears, therefore, that the HR project will follow two distinct paths. Firstly, we will carry on generalising the way in which it forms theories to cover more sciences (indeed, there are currently some bioinformatics projects under way with HR). Secondly, we will add more domain-specific functionality to HR such as the Dirichlet convolution production rule mentioned above.

References

[1] S. Colton. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag, 2002.

[2] S. Colton. The HR program for theorem generation. In *Proceedings of CADE-18*, 2002.

[3] S. Colton. Making conjectures about maple functions. In *Proceedings of the Tenth Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, LNAI 2385*. Springer, 2002.

[4] S Colton, A Bundy, and T Walsh. On the notion of interestingness in automated mathematical discovery. *International Journal of Human Computer Studies*, 53(3):351–375, 2000.

[5] W McCune. The OTTER user's guide. Technical Report ANL/90/9, ANL, 1990.

[6] W McCune. A Davis-Putnam program and its application to finite first-order model search. Technical Report ANL/MCS-TM-194, ANL, 1994.