Cross Domain Mathematical Concept Formation

Cross-domain Mathematical Concept Formation

Graham Steel, Simon Colton, Alan Bundy and Toby Walsh*

University of Edinburgh, 80 South Bridge, Edinburgh, Scotland, EH1 1HN.
*University of York, Heslington, York, England, Y010 5DD.

{grahams,simonco,bundy}@dai.ed.ac.uk, toby.walsh@cs.york.ac.uk

Abstract

Many interesting concepts in mathematics are essentially 'cross-domain' in nature, relating objects from more than one area of mathematics, e.g. prime order groups. These concepts are often vital to the formation of a mathematical theory. Often, the introduction of cross-domain concepts to an investigation seems to exercise a mathematician's creative ability. The HR program, (Colton et al., 1999), proposes new concepts in mathematics. Its original implementation was limited to working in one mathematical domain at a time, so it was unable to create cross-domain concepts. Here, we describe an extension of HR to multiple domains. Cross-domain concept formation is facilitated by generalisation of the data structures and heuristic measures employed by the program, and the implementation of a new production rule. Results achieved include generation of the concepts of prime order groups, graph nodes of maximal degree and an interesting class of graph.

1 Introduction

In previous work on automated mathematical discovery, (Lenat, 1976), (Colton et al., 1999), and in this paper, a mathematical concept is taken to mean a class of mathematical objects, such as prime numbers, square numbers, Abelian groups, complete graphs etc.[1] Mathematical concept formation is the process of identifying new classes of mathematical objects with interesting and/or desirable properties. In human mathematics, this is typically pursued in one of two ways: it can be a free exercise, in which a mathematician is looking for new things to investigate, or a more directed process, in which a mathematician is looking for a concept satisfying certain requirements as part of an investigation or a proof, see Colton (2000), chapter 3.

A mathematical domain is an area of mathematical study. Examples include number theory, the study of questions about numbers (usually meaning whole numbers), and graph theory, the study of sets of vertices, V, and edges, E, consisting of pairs of elements from E (or more informally, the study of diagrams consisting of nodes joined together with lines, see Figure 1). A *cross-domain concept* is a set of objects in one domain that are identified as a distinct class using information from at least one other domain. Some examples illustrate this idea.

The concept of *even order nodes* is cross-domain. It is the set of nodes in a graph which are joined to an even number of edges (see Figure 1). The order of a node is a graph theory concept, and the concept of even numbers is from number theory. Euler's solution to the Königsberg bridge problem, (Euler, 1736), required the use of even order nodes, and launched the field of graph theory.

The city of Königsberg in East Prussia was divided by a river containing two islands. Seven bridges connected the islands to each other and to the banks of the river, and the citizens of the city wondered if there was a way to tour the city crossing every bridge exactly once. Euler proved that this was in general possible if and only if every land mass was connected to an even number of bridges, which was not the case in Königsberg.[2]

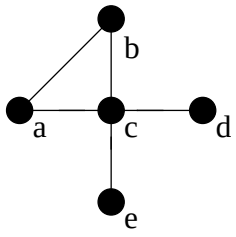

Figure 1: Illustration of even order nodes. Nodes a, b and c are even-order nodes, but nodes d and e are not

[1] A prime number is a natural number with exactly two factors, e.g. 2, 3, 5, 7. A square number is one that is equal to an integer times itself, e.g. 1,4,9,16. An Abelian group is a group in which, for all elements a, b in the group, $ab = ba$. A complete graph is one in which every node is joined to every other node.

[2] See http://www.cut-the-knot.com/do_you_know/graphs.html for more details.

The concept of *prime order groups* is another example of a cross-domain concept. A prime order group is one containing a prime number of elements. Sylow's theorem, (Sylow, 1872), required the concept of prime order groups. While not as easy to state and understand as Euler's result, it too was a breakthrough and forms perhaps the most profound result in finite group theory.[3]

Some of the most common examples of cross-domain concepts relate numbers to other domains, like the two examples above. There are also others, such as sets of matrices forming a group. A recent Fields medal winner was awarded the prize for a cross-domain investigation: Richard Borcherds proved the 'Moonshine Conjecture', originally proposed by John Conway and others, (Conway and Norton, 1979). This theorem links elliptic modular functions with concepts from string theory and group theory.[4]

Cross-domain reasoning also arises in the natural sciences, most commonly when some mathematics is applied to experimental results. For example, Mendel founded the field of genetics when he applied some elementary combinatorics to the results of his famous pea experiments.

In summary, we've seen that cross-domain concepts, while not dense in the mathematical literature, often provide the inspirational step leading to results of real importance. These ideas represent what we think of as creative stages in the development of a theory. An effective automated mathematical discovery package should have the ability to form cross-domain concepts, and therefore be able to provide the inspiration for such creative steps.

1.1 Background

The first automated mathematical concept formation program was Lenat's *AM*, as described in Lenat (1976). *AM* stored concepts as definitions in LISP, and modified old definitions to create new concepts. It also made conjectures about the concepts it had invented. Lenat provided *AM* with 76 initial concepts, mostly referring to bags, sets and set operations. When running, *AM* developed set-theory based definitions of numbers. It then quickly moved into number theory, and re-invented some famous concepts such as highly composite numbers. Lenat followed up *AM* with *EURISKO*, Lenat (1983), which was given some number-theory and set-theory concepts to start with. However, neither program had facilities for explicitly reasoning about the two different domains in order to propose cross-domain concepts.

[3]For those interested, Sylow's three-part theorem built on a result of Cauchy, which stated that a group whose order is divisible by a prime p has an element of order p. Sylow extended this to the following: If p^n is the largest power of the prime p to divide the order of a group G then (i) G has subgroups of order p^n (ii) G has $1 + kp$ such subgroups for some k, (iii) any two such subgroups are conjugate. Almost all work on finite groups uses Sylow's theorems. (O'Connor and Robertson, 1999)

[4]There is a good, short summary of the story of the moonshine conjecture at http://www.sciam.com/1998/1198issue/1198profile.html

GT[5], Epstein (1988), was written by Epstein to perform concept formation, conjecture making and theorem proving in graph theory. In GT, a graph type is represented by a seed, S, consisting of a set of base cases for the type, a constructor, f, and a set of constraints for the constructor, σ. These last two together describe a correct and complete construction of all graphs in the class. GT formed new concepts by generalising, specialising and merging old ones, performing heuristic search on a best-first agenda basis. At least one of the conjectures proposed and proved by GT involved cross-domain concepts:

> There are no odd-regular graphs on an odd number of vertices.

This theorem states that, given an odd number of points, there is no way to join them up such that every point is connected to the same *odd* number of other points. However, no other cross-domain conjectures are reported. GT eventually succumbed to the size of the search space and stopped producing concepts and conjectures that the program's author considered interesting.

The HR[6] program, Colton et al. (1999), is an automated concept formation which also makes conjectures about its concepts and calls on theorem proving and model generation tools to settle them. At the start of an HR session, a domain is chosen, some initial objects in that domain are provided, and axioms for that domain are specified by the user. All model generation and theorem proving tasks are then carried out with respect to these axioms. HR will then proceed to form concepts within that domain, using general production rules such as 'conjunct' (pick out objects satisfying the definition of two previous concepts) and 'common' (pick out pairs of objects sharing a particular property) to make new concepts. A weighted sum is taken of a variety of heuristic measures to evaluate the interestingness of a concept. HR will then apply more production rules to the most interesting concepts. However, the original version of HR could not store objects from more than one domain at a time, so could not reason about multiple domains in order to form cross-domain concepts. The HR project is ongoing, and a new version is being developed in Java, (Colton et al., 2000).

1.2 Paper outline

In the rest of this paper, we describe the generalisation and extension of the HR program to perform cross-domain concept formation. In section 2, we describe the operation and knowledge structures of the HR program, and how they were altered to allow cross-domain concept formation. In section 3 we discuss the results achieved by the new version of the program looking for cross-domain

[5]GT stands for Graph Theorist

[6]HR is named after Hardy and Ramanujam, two famous mathematicians who worked together for a period early in the twentieth century.

Table 1: Concept table for 'group operation' (rows for 2 groups shown)

Group	Element	Element	Element
c2	e	e	e
c2	e	a	a
c2	a	e	a
c2	a	a	e
c3	e	e	e
c3	e	a	a
c3	e	b	b
c3	a	e	a
c3	a	a	b
c3	a	b	e
c3	b	e	b
c3	b	a	e
c3	b	b	a

concepts in graph theory and number theory. These included rediscovery of nodes of maximal degree, discovery of an interesting type of graph and a generally high yield of good quality concepts. In section 4, we draw some conclusions and outline some suggestions for further work.

2 Concept formation in HR

There follows a brief description of the HR program. Further details can be found in (Bundy et al., 1998) or on the HR project webpage[7].

2.1 Representation

Designed initially to work with finite algebras, HR can work in any finite mathematical domain. The user must supply a set of axioms for the domain, and a set of initial concepts to work with. Typically, these will consist of a set of entities from the domain, together with a way of breaking them down. So, for example, in group theory the user might supply the Cayley table[8] for some small groups, or in number theory, the breakdown of the first 10 integers into their divisors.

In HR, each concept is represented as a data table, and each entity to which that concept is applicable will have one or more rows in the table. The first column in the table identifies the entity, and the other columns refer to properties of the entity or its components. For an example, see table 1, the Cayley table for two cyclic groups.

2.2 Operation

HR functions on a best-first heuristic search basis. New concepts are generated by HR's 9 production rules, and evaluated using 7 heuristics. A weighted sum of the seven scores is calculated for each concept, and the concepts are sorted into order, the highest scoring concepts coming first. The weights for each heuristic are set by the user.

2.3 HR's heuristics and production rules

HR uses the following heuristics to measure the interestingness of a concept:

Parsimony Concepts with small data tables tend to describe objects more parsimoniously, so these are scored positively.

Discrimination and Invariance The user can supply a 'gold standard' categorisation, and these two heuristics will measure how close a concept comes to achieving that categorisation.

Complexity This is a measure of how many production rule steps were applied to build the concept. Simpler concepts are preferred.

Applicability The proportion of entities known to HR which are referred to by the concept is the applicability.

Novelty If the categorisation produced by a concept hasn't been seen before, then it is a novel concept, and so scores highly. Categorisations that have been seen many times before yield lower novelty scores.

Provable facts Concepts relating to proved theorems are scored highly.

The production rules used to form new concepts are:

Size Measures the size of a set, e.g. given a table of integers and their divisors, it will produce a table showing how many divisors each number has.

Split Picks out an integer value - usually 1 or 2. Given a table showing how many divisors a number has, the split rule can pick out integers with exactly 2 divisors (prime numbers).

Match Produces a table containing rows with matching columns. So, given a table of integers decomposed into their factors, it can pick out numbers which decompose into two identical factors, i.e. square numbers.

Forall Finds a set of entities, all of which have sub-objects of a certain type. Given the concept of central elements in a group, it will produce Abelian groups (i.e. groups in which all elements are central).

Conjunct Forms the conjunction of two previous concepts, e.g. from square numbers and prime divisors, it can form squares of primes.

[7] http://www.dai.ed.ac.uk/šimonco/research/hr

[8] A Cayley table for a group gives the result of applying the group's multiplication operation to any pair of elements.

Exists Removes a column from a data table. So, given a table consisting of groups and their elements of order 3, it will produce a table of all groups which contain an element of order 3.

Compose Given two concepts representing a function, this production rule will produce a data table representing their composition. For example, given a data table containing groups and their sizes, and a data table containing integers and their prime factors, the compose rule would produce a table containing groups and the prime factors of their size.

Common Picks out entities with rows in common, e.g. integers sharing the same prime divisors.

Negate Returns the entities and corresponding rows *not* appearing in a previous concept's table with respect to the complete list of entities provided by the user. For example, given the concept of even numbers, this production rule will produce the concept of odd numbers.

Note that none of the production rules are domain specific, and all perform only very basic operations.

2.4 Adapting HR for Cross-domain Work

We now look at the modifications made to HR in order to allow it to form cross-domain concepts. As we mentioned above, Colton's HR program performs not only concept formation but also conjecture making, theorem proving and counterexample finding. In the work reported here, however, only the concept formation capabilities of the program were used. This was because HR currently lacks the ability to interface with inductive theorem provers, and so would be unlikely to be able to prove any conjectures involving numbers or graphs. It was anticipated that all the testing would involve number theory, so the theorem proving mechanism of the program was switched off for this work.

The first stage of the development work involved making some changes to the way HR stores concepts in order to be able to distinguish between concepts from different domains. HR stores a list of 'entities', e.g. a list of groups or a list of numbers depending on the working domain. This was changed to allow entities to be identified with their domains.

Some small changes were required to the way HR builds new concepts. For example, `negate`, returns a table containing entities not satisfying a previous concept definition. This was changed to test the domain of the previous concept, and return a table containing only those entities *from the same domain* which do not satisfy the previous concept definition.

Certain aspects of the way HR measures the interestingness of its concepts also required modification. One heuristic HR employs measures the 'applicability' of a concept, i.e. the proportion of entities to which it applies. This had to be changed so that HR only measured how many entities in the concept's own domain were referred to by the concept. The 'complexity' measure was also changed so as to be more lenient towards cross-domain concepts. The measures concerned with classifications (discrimination, invariance and novelty) required changes to allow meaningful measurements to be made with respect to the applicable domain. These changes consisted of directing HR to only refer to categorisations in the same domain when making the measurements.

2.4.1 User settings to control cross-domain work

Some new settings were added to allow the user to control the multi-domain aspects of an investigation. The first alteration was to set up a system whereby the user can prescribe the amount of investigation to take place in each domain presented. This was achieved by adding user settable options `domain_list` and `domain_multiplier`. The user sets `domain_list` to be a list of domains, say `[group,integer]`, and `domain_multiplier` to be a number, say 50, and then HR will produce 50 concepts in graph theory, 50 concepts in number theory, and then carry on pursuing whatever it finds most interesting. This setting has no effect on the cross-domain aspect of the investigation, it simply specifies the domain that a concept should refer to.

The second change was to add two more user options to control the cross-domain behaviour of the program, `no_cross_before` and `encourage_cross_after`. The user sets `no_cross_before` to a value, say 150, and then no cross-domain concepts will be formed until 150 single-domain concepts have been formed. `encourage_cross_after` is set to a value, say 200, and then after 200 concepts have been formed, cross-domain pairs will be chosen first by the two-table production rules.

A further change was to relax the complexity limits for cross domain concepts. HR has a user settable option called `complex_max` which sets a depth limit for the search. This was modified so that a concept relating objects from n domains could be built on up to a depth of n times the complexity limit set.

2.4.2 A new production rule: `extreme`

Although HR's existing multi-concept production rules could be made to function across multiple domains (with minor modifications), it was important to establish the efficacy or otherwise of production rules designed explicitly to facilitate cross-domain reasoning. We decided to develop a rule to introduce extremes of orderings into HR. Several previously inaccessible concepts require the use of an ordering. For example, in graph theory we may be interested in the node of maximum degree, the largest clique, or the longest path. In group theory, we might be interested in the elements of maximal order or the largest proper subgroup. In number theory, we might be interested in the largest prime divisor or largest common

Table 2: `extreme` with parameters (1,3). Bold rows are extracted

Group	Element	Number
G0	a	1
G0	b	2
G0	**c**	**3**
G1	a	1
G1	**b**	**2**

Table 3: `extreme` with parameters (1,3). Bold rows are extracted

Graph	Node	Number
G0	a	4
G0	a	5
G0	a	7
G0	b	1
G0	**b**	**9**

factor. We needed to keep the rule general, to allow HR to use any ordering it has available, and to allow it to orient an ordering in either direction, in order to be able to extract both maximal and minimal values.

This new production rule was called `extreme`. It takes in two pre-existing concepts, and treats one of them as an ordering. It then takes the other concept and extracts only those rows whose entry in a specified column (first parameter) are the 'largest' for a specified entity (second parameter: graph, node, group member, etc.) with respect to the ordering chosen. If several rows share the same extremal value, they are all extracted. Tables 2, 3 and 4 show examples, using the ordering in table 5.

Note that no ordering properties such as asymmetry or transitivity are assumed or checked for when choosing a concept to use as an ordering for the `extreme` rule. This was a deliberate decision, taken for several reasons: firstly, checking that a table has the necessary properties to qualify as a partial or total ordering would be computationally expensive. Secondly, it was anticipated that completely inappropriate ordering tables would tend to give empty tables, i.e. would not have any extremal values, and so HR would not form a new concept (it would instead conjecture that such a value did not exist). This was borne out by the results achieved. Thirdly by keeping the rule as general as possible we were giving HR a chance to come up with something truly novel using a table as an 'ordering' that a human mathematician had never previously considered using. The original HR project had followed a pro-generality methodology, and we wanted to preserve this in our extensions.

Table 4: `extreme` with parameters (2,3). Bold rows are extracted

Graph	Node	Number
G0	a	4
G0	a	5
G0	**a**	**7**
G0	b	1
G0	**b**	**9**

Table 5: Ordering - this concept is given to HR when working in number theory

Number	Number
1	0
2	0
2	1
3	0
3	1
3	2
⋮	⋮

3 Results

We evaluated the final product with respect to several precise criteria, in the hope that all these measures together would give an accurate account of the degree of success of the project. The two main areas of testing were: an evaluation of HR's ability to spot classically interesting cross-domain concepts and evaluation of the quality of the new concepts output by HR. There follows a description of the testing criteria, the methods used for testing and the results. In this paper, for considerations of space and intelligibility to non-mathematicians, we report only the results obtained in graph theory and number theory. Details of group theory and number theory testing, and further details of the graph theory testing including the complete output from a run, can be found in (Steel, 1999).

3.1 Generating standard interesting concepts

To measure the ability to re-invent standard cross-domain concepts, we compiled a list of target concepts for which the original HR implementation was capable of finding the individual single domain concepts required. Then we ran HR in the two domains, and examined the output to see how many of these concepts had been rediscovered. The methodology used was the following: first set HR up with the correct production rules to build the single domain concepts that are required to form the cross-domain target concepts. Then, set the weights for the concept measuring heuristics, and set HR off to form 500 concepts.

Several 500 concept batches were run with different

Table 6: Graph theory and number theory target concepts. The double line separates 'core' concepts from 'peripheral' concepts

Concept	Reason for Inclusion
Eulerian graphs	A result in the first graph theory paper, Euler (1736).
Maximal order nodes	Used in many inequalities of invariants.
Minimal order nodes	Used in the greedy graph factorization algorithm and several inequalities.
Odd order nodes No. of odd order nodes	A graph contains an Eulerian path if it has 0 or 2 odd order nodes.
Even order nodes	Needed for Eulerian graphs.
Number of nodes of max/min order	Give an indication of the structure of the graph.
Order of a star graph centre	Identifies star graph uniquely.
Star graph with an even order node	Characterises symmetric star graphs.

weights assigned to the concept measures. After each run, weights were altered to try to favour the root single domain concepts required to build the missing targets. The idea behind this was that if the target concepts really were representative of the kind of concepts we want HR to find, then by adjusting the weights to perform well on these concepts, we would not be 'over-tweaking' but rather determining a set of weights well-suited to concept formation in that particular pair of domains. In the event, after four 500 concept runs, we had found a set of weights that gave our best performance on the target set.

3.1.1 What is a 'classically interesting' concept?

Cross-domain concepts, whilst vital to mathematics, are not particularly dense in the literature. This is inconvenient, as we need a significant number of target concepts to get representative test results. Consequentially, a small number of vital concepts were picked out and identified as 'core' target concepts. Then, some slightly more obscure concepts that could still be considered interesting were identified, and labelled as 'peripheral' target concepts. Table 6, gives the test concepts for the runs in graphs and numbers, and the reasons for their inclusion.

3.1.2 Standard concepts reinvented

HR was able to reinvent 2 out of 3 of the core targets and 3 out of 5 of the peripheral targets in the testing undertaken. The missed targets were traced to a particular step in the investigation where HR applied its `exists` production rule in such a way as to miss abstracting away the actual order of the nodes. This can be easily fixed - a version of HR is under development that will insist on applying a production rule with all possible parameters before attention is switched to another concept, or another rule.

3.2 Inventing new concepts

We measured the interestingness level of all the concepts produced in a 500 concept run on the following scale:

Type 1 - Concepts in the classical target set (core or peripheral).

Type 2 - Concepts of a similar level of interest to those in the peripheral target set, of interest but only in specialised areas of the theory. For example, non-regular graphs, graphs with all nodes of order greater than one and nodes of prime order were all found in one test run and classified as type 2.

Type 3 - Concepts which may be of interest, but only in a specialised situation. For example, graphs with more than two nodes of order 1 and graphs where all nodes are of order either a or b, and $a + 1 = b$.

Type 4 - Anything not falling into the above three types

To analyse thoughtfully and accurately the quality of cross-domain concepts in a 500 concept run with respect to our four-point scale was a time consuming process, and only one run was completely analysed in this way. The run chosen was the run in graph theory and number theory that produced the most classical target concepts. A complete list of the concepts evaluated, and their classifications, is given in appendix 1 of (Steel, 1999).

HR's own measures of interestingness were not referred to at this stage of the evaluation process. By default, HR re-evaluates all the concepts it has invented so far every time it invents another 10 new concepts. After this re-evaluation, the top ten concepts (ranked by interestingness) are displayed on the screen. For this evaluation run, the program was altered to prevent it from displaying the top ten list, to ensure that the concepts were categorised purely on their apparent mathematical merit. This allowed us to compare HR's measure of interestingness with our own later (see section 3.3).

The proportion of concepts fitting into each of the four categories are illustrated in the pie chart in figure 2. We can classify a concept satisfying the criteria for a type 1, type 2 or type 3 classification as being 'acceptable', in that it must at the very least be plausible and of some interest. Of the concepts in the run analysed, 56% were acceptable. This compares favourably with Lenat's 125 out of 300 (42%)'acceptable' concepts in *AM*, and 200 out of 1000 (20%) for *EURISKO*. Few very interesting concepts (i.e. type 1 or type 2) were found in the run (just 8% of the total cross-domain concepts). This is comparable with Lenat's 25 out of 300 'really interesting' concepts in *AM* (8.3%). Of course, we must be wary of attaching excessive importance to subjective judgements of this kind without knowing exactly what Lenat was classing as an 'acceptable' or a 'really interesting' concept.

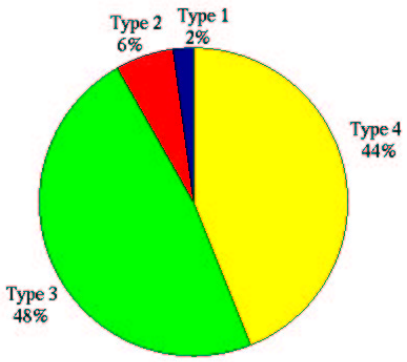

Figure 2: Overall concept quality

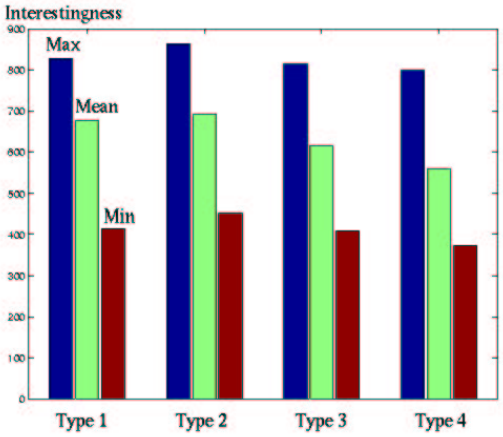

Figure 3: HR's evaluation of concepts' interestingness

3.3 Effectiveness of HR's interestingness measures

An effective concept formation program should rate as interesting the same concepts a human mathematician would rate as interesting. We evaluated this by collating the interestingness measures assigned to the concepts in the fully evaluated run presented in figure 2. The maximal, mean and minimal interestingness scores for each category of concept are illustrated in bar chart form in figure 3. There is a significant fall in the mean level of interestingness (the central, light coloured bar in the chart) from type 2 to type 3 to type 4, but a smaller fall in the maximal and minimal values. The spread of interestingness values assigned between maximum and minimum for any given concept type was more significant than the scale of the decrease in the mean value. This is unfortunate, as it means HR can be misled: it could assess a type 4 concept to be more interesting that a type 1 concept. HR cannot currently undertake any mathematical investigation of these concepts as its proof tools are limited to first order logic. HR has no background knowledge of the mathematical literature. Despite this, we are asking it to give an instant evaluation of the mathematical worth of a concept. Given the scale of this task, achieving a general correlation with the human assessment of interestingness is a good result, but the degree of mis-assessment imposes a limit on the success of the program. Some suggestions for improvements are given in section 4.

3.4 Discovery highlights

A good automated concept formation program should come up with some new novel and interesting concepts. To evaluate this, one interesting looking concept was extracted from the analysed graph theory run. It had the following definition:

> A Graph G with a node $n1$ of order M such that $\forall$ nodes $n2 \in G$, order($n1$) $\geq$ order ($n2$), and $M = \{I | \exists$ a node $n3 \in G$,order($n3$) $= I\}$.

This corresponds to a graph with a node of order M that is the maximal order node in a graph in which there are nodes of M different orders. The definition is fairly complicated, but a moment's thought reveals that such a graph must have at least one node of every order from 1 to M. What is the minimum number of nodes that a simple graph[9] with such a property can be drawn on for a particular M ? We managed to prove the following simple theorem:

Theorem 1 $\forall m \in \mathbb{N}, m \geq 1, \exists G$, *a graph on* $m+1$ *vertices s.t.* $\forall n, 1 \leq n \leq m, \exists$ *a node of order* n *in* G

Proof The proof is by induction on m, details in (Steel, 1999).

This type of graph was new to the authors, but it has appeared in mathematical literature. In (Zeitz, 1999), a problem is posed involving a host inviting 10 couples for a party:

> I ask everyone present, including my wife, how many people they shook hands with. It turns out that everyone shook hands with a different number of people. If we assume that no one shook hands with his or her partner, how many people did my wife shake hands with? (I did not ask myself any questions.)

[9] A simple graph is one with no duplicated connections and no loops.

By drawing a graph, which turns out to be of the type we rediscovered, and applying a little induction, Zeitz shows that the hostess must have shaken 10 hands.

So, our graph theory concept is involved in at least two simple but interesting pieces of mathematics. This is a promising result. The concept was also a complete surprise to the authors - it seems to be a characteristic of HR that it is able to find a way of inventing concepts which at first thought one would not expect it to have the capacity to represent. HR finds them interesting because it can evaluate qualities of the underlying data rather than just the definition.

4 Conclusions and Further Work

The control structure of the cross-domain HR could be modified so that cross-domain concepts are only formed when they are needed to develop a theory further. HR could have a pre-set interestingness limit, and attempt to generate cross-domain concepts only when formation in a single domain was producing concepts below that threshold setting.

As highlighted by the bar chart in figure 3, there is room for improvement in HR's interestingness heuristics. One easy way to improve performance would be to increase the number of models HR is given in each domain. This would make the applicability measure more accurate, and so decrease the interestingness of concepts which consist of a convoluted definition of one particular model. However, it would also slow the program down.

The conjecture making and theorem proving aspects of the HR project have not been extended in this project. Allowing HR access to inductive theorem provers would give it a chance to prove some cross-domain conjectures involving numbers. In particular, many graph theory proofs are based on induction. Being able to prove cross-domain conjectures would also allow HR to judge the worth of cross-domain concepts more accurately, as it does in the single domain version.

Designing and adding further production rules would enhance HR's coverage of mathematics. A useful exercise would be to pick some concepts, say from an index or glossary in an undergraduate text, and analyse what kind of rules would be required to build those concepts. This would at least give a feel for the scale of the problem, i.e. whether we need ten more production rules or a thousand more production rules. It would be useful to carry out this kind of analysis before embarking on further extensions to the work.

The results achieved by the program were generally encouraging. In particular, the ideas behind HR were seen to generalise to a search space with a much larger branching factor without destroying the quality of the concepts constructed. If the cross-domain conjecture making abilities of HR can be extended similarly, and there is every reason to believe that they can, then perhaps a future version of HR will be able to come up with a discovery of real mathematical importance.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. The work reported here was supported by an EPSRC grant.

References

A. Bundy, S. Colton, and T. Walsh. HR - automatic concept formation in finite algebras. Technical Report 920, (Presented at the Machine Discovery Workshop at ECAI 98) Department of Artificial Intelligence, University of Edinburgh, 1998.

S. Colton. *Automated Theory Formation in Pure Mathematics*. PhD thesis, Division of Informatics, University of Edinburgh, 2000.

S. Colton, A. Bundy, and T. Walsh. HR: Automatic concept formation in pure mathematics. In *IJCAI '99*, 1999.

S. Colton, A. Bundy, and T. Walsh. Multi agent theory formation in mathematics. In *Creative and Cultural Aspects of AI*, Birmingham, England, 2000. AISB.

J. Conway and S. Norton. Monstrous moonshine. *Bull. Lond. Math. Soc.*, 11:308 – 339, 1979.

S. Epstein. Learning and discovery: One system's search for mathematical knowledge. *Computational Intelligence*, 4(1):42–53, 1988.

L. Euler. Solutio problematis geometrian situs perinentis. *Opera Omnia*, 7(1):128–140, 1736.

D. Lenat. *AM: An artificial intelligence approach to discovery in mathematics*. PhD thesis, Stanford University, 1976.

D. Lenat. Eurisko: A program which learns new heuristics and domain concepts. *Artificial Intelligence*, 21: 61–98, 1983. The Nature of Heuristics III.

J. O'Connor and E. Robertson. The mactutor history of mathematics. Available on the web at http://www-groups.dcs.st-and.ac.uk/history/index.html, 1999.

G. Steel. Cross-domain concept formation using HR. Master's thesis, University of Edinburgh, 1999.

L. Sylow. Théorèmes sur les groupes de substitutions. *Mathematische Annalen*, 5:584–594, 1872.

P. Zeitz. *The Art and Craft of Problem Solving*, chapter 3, pages 84–85. Wiley, 1999.