Experiments in Meta-theory Formation

Simon Colton
Division of Informatics
University of Edinburgh
Edinburgh EH1 1HN
Scotland
simonco@dai.ed.ac.uk

Abstract

An ability to reason at a meta-level is widely regarded as an important aspect of human creativity which is often missing from creative computer programs. We discuss recent experiments with the HR theory formation program where it formed meta-theories about previously formed theories. We report how HR re-invented aspects of how it forms theories and reflected on the nature of the theories it produces. Additionally, the meta-theories contains higher level concepts than those produced using HR normally. We discuss how HR's meta-level abilities were enabled by changing domains, rather than writing new programs, which was the model previously employed in the Meta-DENDRAL and Eurisko programs. These experiments suggest an improved model of theory formation where meta-theories are produced alongside theories, with information from the meta-theory being used to improve the search in the original theory.

1 Introduction

In his presidential address, "Creativity at the Meta-Level" at AAAI-2000 (1), Bruce Buchanan left no doubt about how important he feels meta-level reasoning is in the computer modelling of creativity:

> 'The key to building more creative programs is to give them the ability to reflect at the meta-level on their own framework and criteria. ... I believe that creativity is at the meta-level.'

We took this talk as a challenge to equip our HR program, (3; 4) with some meta-level abilities. We approached this problem in two stages: firstly, enable HR to produce meta-theories and then enable HR to utilise information from the meta-theories to improve it's theory formation abilities in general. We discuss meta-theories in detail later, and it suffices at present that a meta-theory is taken to be a theory about a theory.

Our work towards the second stage is too preliminary to report on here, and we concentrate on how HR forms meta-theories, rather than how it utilises them. We first motivate this by looking at previous programs with meta-theoretic abilities. Following this, we describe certain important aspects of the HR program. We then describe HR's meta-theoretic abilities by discussing what we mean by a meta-theory and providing details of three experiments where HR formed meta-theories. We present the results from these experiments and conclude by discussing the implications of these results and by looking at the future directions we intend to take to improve HR's meta-theoretic abilities.

1.1 Background

Much has been written about meta-level reasoning, and we concentrate here on two discovery programs which had meta-level capabilities. Buchanan's heuristic DENDRAL program (2) (shortened to just DENDRAL) was designed as an assistant for structure elucidation in organic chemistry. Given a chemical for which the structure was unknown, various chemical, physical and spectroscopic data were collected and interpreted in terms of fragments for the whole molecule, and this information constrained a search for candidate structures. DENDRAL generated the candidate structures based on both the structural information available and other constraints the user added. DENDRAL worked in a cyclic, interactive process, with the structures produced by DENDRAL suggesting further constraints (or the removal of constraints) and further experimentation to gain more structural information. All the new information was given to DENDRAL so that the search produced fewer candidate structures until eventually the resulting set was manageable and meaningful. The heuristic aspect of DENDRAL came in the form of a planner which used spectroscopic knowledge to infer constraints from the data available, e.g. on page 10 of (2) Buchanan gives this example:

> '... [DENDRAL] may infer that the unknown molecule is probably a ketone, but definitely not a methylketone.'

DENDRAL was a celebrated early success for Artificial Intelligence as it equalled human experts at its task, and made discoveries in many areas of chemistry which outperformed humans, for example (14).

Buchanan et al. progressed to the Meta-DENDRAL program. Much of DENDRAL's success was due to handcrafting of domain specific information and Meta-DENDRAL was constructed to help in the construction of the domain-specific production rules which DENDRAL relied upon. In particular, the version of Meta-DENDRAL reported in (2) helped chemists determine relationships between mass spectrometer data and structural features of molecules. This worked in a similar way to DENDRAL, with a highly customisable rule generator for which the user could change the vocabulary of the rules, the syntax of the rules and the semantic constraints governing the plausibility of the rules. Meta-DENDRAL was also able to test and reject rules using the spectroscopy data, refine the rules and reduce the redundancy of the set of rules produced. In this fashion, Meta-DENDRAL successfully rediscovered published rules of mass spectrometry and discovered new rules which had not previously been published. More importantly, the rules generated by Meta-DENDRAL significantly improved the performance of DENDRAL, as it was designed to do.

In the AAAI-00 keynote speech, Buchanan cited the Eurisko program (10) as another with meta-level abilities. Eurisko was Douglas Lenat's successor to his AM program (8). AM was designed to explore elementary number theory and set theory by forming concepts and making conjectures. Tasks for completion were put on an agenda, with the position on the agenda determined by worth values for both the concept in the task and the nature of the task. The worth values were calculated by some of the 242 heuristics available to AM. The heuristics also suggested new tasks to put on the agenda, and when to invent new concepts. AM successfully re-invented concepts such as prime numbers and highly composite numbers, and found classically interesting conjectures such as Goldbach's conjecture. However, there have been many criticisms aimed both at how AM operated and Lenat's reporting of the project (3; 13).

One of the AM's failings, which Lenat freely admitted (e.g. (9), page 7) was that it's performance degraded as it progressed further from its initial base of concepts. In (3) we argue that this is because the heuristics were fine-tuned to find particular classically interesting results (which Lenat goes to great lengths to deny in (11), page 289). However, even though AM had 242 heuristics already, Lenat believed that AM's failure was:

> '... due to its inability to discover, new, powerful, domain-specific heuristics for the various new fields it uncovered.' (10), page 61.

Lenat wrote Eurisko to solve this problem, namely to form concepts about heuristics in the same way that it formed concepts about the objects in the domain it was studying. Hence Eurisko was able to work at the domain-level and meta-level at the same time. Eurisko operated in a similar way to AM, with a similar agenda mechanism, although with fewer given heuristics. In Eurisko, however, the theory formation process could be applied to itself. In particular, the control algorithm for the agenda was represented in Eurisko as a set of concepts, so that Eurisko could modify them, thus modifying how it operated. Similarly, the heuristics were represented as concepts, so that Eurisko could modify them and invent new ones. For example, Eurisko invented the heuristics that it is worthwhile to search for a fast algorithm for computing inverse functions, if that function is going to be used (paraphrased from (10), page 84).

Eurisko was initially applied to the same domain as AM, elementary mathematics. However, Lenat admits:

> 'Sadly, no powerful new heuristics, specific to set theory or number theory, were devised. This may reflect the 'well-trodden' character of elementary mathematics ...'

and the Eurisko project appears to have added little to our understanding of automated mathematical discovery. Eurisko was successful, however, in the domain of Naval Fleet Design, and Eurisko competed in and won the Traveller Trillion Credit Squadron wargame (10).

Buchanan and Lenat both chose to write new programs to facilitate meta-level reasoning. Their original programs were creative in their original domains, but were not general enough to transfer their skills to a meta-level. In both cases, the meta-level program and the original had similar top-level processes, and it was possible to design a generalised program able to work at both levels, which was realised with Eurisko, but not with Meta-DENDRAL. As we shall see, we have adopted a different approach to equipping our HR program with meta-level abilities. Rather than changing programs, HR was general enough for us to simply change domains. So, rather than writing a 'meta-HR' program, we simply enabled HR to produce information about how it operates and what it produces, and then used HR to form a theory about this information. We discuss this process after an overview of HR.

2 An Overview of HR

We will be discussing later how the meta-theories produced by HR identify certain qualities of the theories produced as well as aspects of how HR works internally. Therefore it is important to describe both what constitutes the theories HR produces and how they are produced.

2.1 What HR Produces

HR is designed to form theories about a given set of objects of interest, using background information also supplied by the user. The background information consists of initial concepts, each supplied with examples and a definition. Theories are built by inventing new concepts based on the user-supplied ones and by making conjectures involving the concepts. Theories therefore consist

of (i) concepts, which have both a set of examples and a definition (ii) conjectures made empirically using the data available to HR, and (iii) where possible, proofs of the conjectures made which turned out to be true.

Primarily, HR has worked in mathematical domains and has been given mathematical objects of interest such as integers, graphs and groups. However, we have recently enabled HR to translate files which were meant as input to the Progol program (12). In effect, this means that the user can supply first order predicates indicating both the objects of interest and some initial concepts, and HR will form a theory from this information. In particular, one of the example files which comes with the Progol distribution contains information about a set of animals. The file describes 18 animals (bat, cat, crocodile, dog, dolphin, dragon, eagle, eel, herring, lizard, ostrich, penguin, platypus, shark, snake, t_rex, trout and turtle), using the following nine predicates:

- animal(A)
[A is an animal, e.g. cat is an animal]
- of_class(A, B)
[animal A is of class B, e.g. eagle is a bird]
- has_covering(A, B)
[animal A is covered by B, e.g. cat is covered by hair]
- has_legs(A, B)
[animal A has B legs, e.g. eagle has two legs]
- has_milk(A)
[animal A produces milk, e.g. cat produces milk]
- homeothermic(A)
[animal A is homeothermic, e.g. eagle is homeothermic]
- in_habitat(A, B)
[animal A lives in habitat B, e.g. cat lives on land]
- has_eggs(A)
[animal A produces eggs, e.g. eagle produces eggs]
- has_gills(A)
[animal A has gills, e.g. trout has gills]

The classification of animals is given as positive and negative examples, for instance that trout is a fish (positive) but cat is not a fish (negative). The task set for Progol is to discover why we classify animals into mammals, reptiles, fish and birds. Progol performs well: it learns, for example, that if an animal produces milk, it is a mammal, or if an animal has gills, it is a fish and so on.

The task we set HR is slightly different: to form a theory of animals, rather than to learn a particular concept. The theories HR produces include new concepts such as bipedal and quadrupedal animals, and animals which have more than one habitat, e.g. crocodiles, which live in land and water. An important aspect of concepts is their *arity* which is the number of entities in the tuples of examples for the concept. For instance, the in_habitat concept above discusses pairs of [animal, habitat], hence the concept is of arity 2. HR's theories also include some conjectures about animals, such as (i) an animal is a mammal if and only if it produces milk and (ii) an animal is a reptile if and only if it has scales but no gills. Hence HR completes the learning task performed by Progol, but these results are just one of many in the theory HR builds. HR produces many other conjectures including some fundamentals of the nature of animals, such as: no animal has scales and is homeothermic, and no animal has feathers and gills. HR also notices facts such as: platypus is the only animal which produces both milk and eggs.

We will be using the animals example throughout, as the concepts and conjectures produced are typical of those produced by HR and are in general easier to understand than those produced in mathematical domains. In mathematical domains, HR also proves theorems and generates counterexamples to non-theorems, but we will not be discussing this functionality here.

2.2 How HR Produces Theories

HR has 7 general production rules which turn one (or two) old concepts into a new one. For example, the split production rule performs instantiation, e.g. taking the user given concept of the habitat of an animal and specialising this into the concept of animals which live on land. The split production rule is a 'unary' production rule, as it builds new concepts from a single old concept. Other production rules build new concepts from two old concepts, and we call these 'binary' rules. Theories are produced by completing successive theory formation steps, which are kept on an agenda. A theory formation step involves using a production rule to attempt to produce a new concept from old ones. This may result in a new concept, a conjecture of possible interest to the user, or a conjecture which HR knows will be of no interest and so is discarded. For example, conjectures of the form $\forall a\ p(a) \iff p(a)$ are obviously uninteresting and so are discarded. This is a feature of the new Java implementation of HR which was not present in the Prolog version reported in (3).

Given the production rule and old concepts to use in the theory formation step, HR also generates a set of parameterisations which dictate exactly how the construction is to occur. Hence, many constructions can be made from the same pair of (old concepts, production rule). For example, when using the split production rule, the parameterisation dictates which variable should be instantiated to which value. Each production rule produces concepts which differ only slightly from those input to it, and each construction is general enough to use in any domain. Complicated concepts can be built incrementally in this way, e.g. see (3) for details of how the number theoretic ϕ function — which counts the number of integers less than and co-prime to a given integer — is built from just the concepts of division and less-than.

The production rules generate both a definition and a set of examples for every new concept. HR uses the examples to make conjectures about the nature of the con-

cepts it produces. In particular, whenever HR produces a new concept which has exactly the same examples as a previous concept, it makes the conjecture that the two definitions are equivalent. For instance, when HR specialises the concept of animal to the concept of fish by instantiating the class to fish, it notices that the examples of this new concept are exactly the same as those for the user-supplied concept of animals which have gills. Using this evidence, HR makes the following equivalence conjecture: an animal is a fish if and only if it has gills. HR also produces non-existence conjectures whenever it tries to invent a new concept, but finds that there are no examples which satisfy the definition of the new concept. For instance, when HR uses the compose production rule to invent the concept of animals which have gills and produce milk, it finds that, of the 18 example animals it has, none have both these properties. HR uses this evidence to make the following non-existence conjecture: no animal has gills and produces milk.

Because each new concept is built from previous ones and the seven production rules have different parameterisations, one concept can be used as the basis for many others, and HR runs into a combinatorial explosion. A major aspect of our work has been to enable HR to perform a heuristic search whereby new concepts are built from the more interesting old concepts before the less interesting ones. HR performs many calculations to find numerical values of worth for each concept, with an overall value for the interestingness of the concepts being calculated as a weighted sum of all the individual measures (with the weights set by the user).

As discussed in (7), the measures of interestingness are based on many different aspects of the concepts, including their definition, how they were built , their examples and the conjectures made about them. For instance, the *complexity* of a concept is calculated as the reciprocal of the number of theory formation steps undertaken to build the concept. This roughly approximates how comprehensible the definition of the concept will be. Some of the measures of interestingness are based on the categorisation of the objects of interest which the concepts achieve. Each concept describes the objects of interest, so categorising together all objects given the same description will produce a categorisation for every concept. For instance, the user given concept of has_legs(A, B) described above, categorises the animals thus:

[[bat, eagle, ostrich, penguin, platypus],
[cat, crocodile, dog, dragon, lizard, t_rex, turtle],
[dolphin, eel, herring, shark, snake, trout]]

This categorisation has three categories: animals with two legs, animals with four legs and animals with no legs. The *novelty* of a concept is calculated as the reciprocal of the number of other concepts in the theory which achieve the same categorisation. For instance, in a recent theory produced by running HR with a breadth first search for 1000 steps with the data on animals, the above categorisation was seen 7 times, so concepts achieving this categorisation scored $\frac{1}{7} \approx 0.142$ for novelty. Another measure based on categorisations is the *variety* of the concept, which is calculated as the number of distinct categories in the categorisation achieved by the concept. The has_legs concept above would score 3 for variety.

Other measures of interestingness involve the examples of the concepts. For example, the applicability of a concept is taken as the number of objects of interest in the theory which have non-trivial examples for the concept. For example, the concept of the habitat of birds has applicability 3, because the only non-trivial examples are:

{[eagle,air],[eagle,land],[ostrich,land],[penguin,water]}

As only eagle, ostrich and penguin have non-trivial examples, the concept scores 3 for applicability.

Another measure of interestingness is the number of conjectures involving a concept. HR also has some measures of interestingness for conjectures, so that it can measure the *quality* as well as the *quantity* of conjectures which involve a concept. Note that, during one session, the user may decided that concepts involved in many conjectures are the most interesting because there is much to say about those concepts. Hence the user would weight this measure positively in the weighted sum. However, in another session, the user may decide that concepts involved in fewer conjectures are more interesting because they are somewhat mysterious. Hence the user would weight the number of conjectures measure negatively for this heuristic search. This highlights the subjectiveness inherent in the usage of the measures.

The evaluation function guides the search by determining the most interesting concepts so that they can be used as the basis for further theory formation steps. HR also restricts the search using *forbidden paths*. In the code for HR, the forbidden paths are specific restrictions on which theory formation steps can be used for concepts which have been constructed in particular ways. For example, the negate production rule introduces negation to a concept's definition, for instance taking the concept of animals which have gills to the concept of animals with no gills. Repeating the negation step will produce a double negation: the concept of animals which don't not have gills. This would result in the trivial equivalence conjecture: an animal has gills if and only if it doesn't not have gills. Such conjectures are instantiations of the tautology:

$$\forall a \; p(a) \iff -(-p(a)),$$

which is true for any predicate p and any objects a (where $-p(a)$ signifies the negation of p). Such conjectures are undesirable because they are obviously true and in general add nothing to the theory. For this reason, one of HR's forbidden paths forbids the negation of a concept which is already a negation. HR has many such forbidden paths, which we have implemented in an ad-hoc manner, by identifying tautologies and determining how to eliminate them by restricting certain constructions.

3 Three Meta-theory Experiments

There are many different architectures for producing a theory which is somehow at a higher level than those normally produced by HR. We discuss alternatives in §5, but have so far restricted ourselves to experiments involving the production of a single meta-theory which is built from information taken from a single theory. Just like any other theory, the meta-theory will contain concepts and conjectures, but we distinguish them by calling them meta-concepts and meta-conjectures. The meta-concepts and meta- conjectures will be about the concepts and conjectures (and other information) from the original theory. At present, the model we have adopted is to form a theory and then form a meta theory, but we speculate on how to form the meta-theory alongside the theory in §5.

To facilitate a meta-level, we added a module to HR to enable it to output information about the concepts and conjectures in a theory as predicates in the format it is able to read as input. Thus, the information about the theory was used as the background information input to the meta-theory. For example, suppose an original theory contains concepts $c_1, \ldots, c_{10}$, and, as in experiment 1 below, HR writes the information about how each concept was constructed as ten predicates such as this:

- $construction(c_9, compose, [1, 0, 2], c_1, c_2)$

This specifies that concept c_9 was constructed using the compose production rule with parameterisation $[1, 0, 2]$ from concepts c_1 and c_2 (see (6) for details of the compose rule). These construction predicates become the background information for the meta-theory, and HR will form meta-concepts such as: concepts constructed using the compose production rule. This is typical of the kind of meta-concepts found in the meta-theories.

In all 3 experiments described below, a theory of animals was produced using a best first search for 150 steps. The small number of steps was to limit the size of the theory, as information from the theory was input to the meta-theory as background information. HR is not yet efficient at forming theories from a large amount of background information, as it was designed to form a theory from the bare minimum of information, e.g. the axioms of a finite algebra. As discussed in (3), we intend to improve HR in this respect, but for these experiments, we used a small number of steps to produce a compact theory to be input to the meta-theory. We used the novelty measure weighted at 0.7 and the applicability measure weighted at 0.3 in the evaluation function for the best first search. This combination has been successful in other domains — in terms of finding interesting concepts — and as we knew nothing about the animals domain, we decided to use this.

For the three experiments, different properties of the theory were used as input to the meta-theory. For each experiment, we explain our choices and describe what we hoped to achieve with the meta-theory. The results from these experiments are given in §4 below.

3.1 Experiment 1

In this experiment, we enabled HR to output predicates describing how the concepts in the animal theory were constructed and which constructions led to the conjectures in the theory. HR wrote these details using the following four predicates:

- concept(C)

[C was a concept in the theory]

- construction(C,P,Pa,X1,X2)

[In the theory, concept C was constructed from old concepts X1 and X2 using production rule P with parameterisation Pa. Note that X2 may be the word "none" if P was a unary production rule]

- equivalence(X1,X2,P,Pa,X3)

[In the theory, when concepts X1 and X2 were used in production rule P with parameterisation Pa, this led to the old concept X3, producing an equivalence conjecture]

- nonexists(X1,X2,P,Pa)

[In the theory, when concepts X1 and X2 were used in production rule P with parameterisation Pa, the result had no examples, leading to a non-existence conjecture]

With this experiment, we wanted to see whether HR could find some meta-concepts and meta-conjectures of importance to the forbidden paths, and perhaps even find some new forbidden paths. In particular, we hoped that HR would make meta-conjectures that certain constructions in the original theory always lead to conjectures. The meta-theory was formed using 5000 theory formation steps using a unary-first search. A unary first search is an exhaustive search which uses the unary production rules greedily, only performing a binary production rule step if there are no unary steps on the agenda. This search tends to develop concepts to a certain extent before combining them with other concepts. This approach produces more specialisations of concepts than other search strategies, which we felt would be a good strategy given our aims. We also suppressed the forbidden path restricting double negation, hoping that HR would re-invent it.

3.2 Experiment 2

For this experiment, we enabled HR to output predicates concerning the categorisations produced by the concepts in the original theory. HR wrote these predicates:

- concept(C)
- catn(X)

[X was a categorisation of the objects of interest achieved by at least one concept in the theory].

- cgy(Y)

[Y was a category in a categorisation of the objects of interest seen in the theory]

- cgy(X,Y)

[categorisation X contains category Y]

- obj_in_cgy(Y, E)

[category Y contains object of interest E]

As many measures of interestingness involve the categorisations achieved by the concepts, we hoped HR would re-invent some of them as meta-concepts in the meta-theory produced in this experiment. Additionally, we hoped that HR might suggest some new measures for concepts and conjectures. Using the animals theory again as input, HR formed a meta-theory over 5000 theory formation steps using a breadth-first search. We chose a breadth first search because they are guaranteed to produce the simplest concepts first, and many of the measures of interestingness we wanted HR to re-invent are fairly simple. Furthermore, we would be more interested in new measures if they were also easy to understand. Moreover, as we were forming a meta-theory in the hope of re-inventing measures of interestingness, we didn't want to cloud the issue by using such measures to produce the meta-theory.

3.3 Experiment 3

For this experiment, we enabled HR to output predicates concerning the tuples of objects in the original theory. The predicates used were:

- concept(C)
- 1tuple(C,A)

[the singleton [A] was an example of concept C]

- 2tuple(C, A, B)

[the pair [A, B] was an example of concept C]

- 3tuple(C, A, B, C)

[the triple [A, B, C] was an example of concept C]

In the first two experiments, it was our intention to enable HR to reflect upon what it produces and how it operates. In this final experiment, the background information for the meta-theory was closer to the background information than in the previous two experiments. We hoped that this would enable HR to produce a theory which contained interesting concepts and conjectures which provided more insight into the original domain (animals). As we were looking for meta-concepts of interest, we used a heuristic search based on novelty and applicability, weighted 0.7 to 0.3, as in the original theory of animals.

4 Results

The theory of animals about which the meta-theories were produced contained 65 concepts, 37 equivalence conjectures and 7 non-existence conjectures. 41 different categorisations of the 18 animals were achieved by the concepts. Experiment 1 took 393 seconds on a Pentium 3 500Mhz processor, and the meta-theory contained 397 meta-concepts, 1570 equivalence meta- conjectures and 2036 non-existence meta-conjectures. Experiment 2 took 542 seconds and the meta-theory contained 666 meta-concepts, 1872 equivalence meta-conjectures and 273 non-existence meta-conjectures. Experiment 3 took 1608 seconds and the meta-theory contained 535 meta-concepts, 2353 equivalence meta- conjectures and 364 non-existence meta-conjectures. The results from all experiments are collected below into three categories: (i) the identification of forbidden paths (ii) measures of interestingness and (iii) higher level concept formation.

4.1 Identification of Forbidden Paths

The meta-theories pointed out what appeared to be some unusual features of the original theory. In particular, in experiment 1, HR made the meta-conjecture that no concept can be constructed using the 'exists' production rule (see (6) for details of this rule). This rule is usually responsible for the introduction of many concepts, but in the original animals theory, it always produced a conjecture rather than a concept. We discovered that this is because the exists production rule normally results in conjectures during the initial stages of theory formation, because the concepts it produces early on are often the user-given concepts. In fact, of the 37 equivalence conjectures in the original theory, 24 (65%) of them were produced using the exists production rule. After a little contemplation, we have realised that this is fairly obvious, and can happen in other theories, not just this one. Moreover, we were certainly not aware of this quirk of theory formation before HR formed the meta-theory.

Similarly, the meta-theory from experiment 1 highlighted that using the size production rule never leads to a non-existence conjecture. This is because the rule counts objects (for example counting the habitats which an animal lives in). When there is nothing to count, the answer is zero, hence the size rule will never produce a concept with no examples. Again, while this is a fairly obvious statement, it had not occurred to us before we read HR's meta-theory from experiment 1. Another meta-conjecture we found was that each concept has a unique construction, which is true, as only one construction is recorded for each concept, with other constructions which led to the same concept recorded as equivalence conjectures.

Unfortunately, we cannot report that our original goal with the meta-theory in experiment 1 was achieved: for HR to re-discover the double negation forbidden path, and possibly to find some ones we were not aware of. We are still studying the output to explain this failure, but it is likely that we need to enable HR to provide more details of it's internal processes for it to be able to hypothesise new forbidden paths (and to rediscover the old ones).

4.2 Measures of Interestingness

We hoped HR would re-invent some measures of interestingness in experiment 2 and the results were very encouraging, with re-inventions actually occurring in all three meta-theories. In experiment 1, HR invented the meta-concept of the number of equivalence conjectures which involve a concept, and did likewise for the number of non-existence conjectures. At present, HR only counts

the number of conjectures which involve a concept, but we have often thought of splitting this measure into the two measures suggested by HR in the meta-theory. In experiment 3, we noticed that HR had invented measures which were actually the applicability measure specialised for concepts of particular arities. For example, HR invented the meta-concept of the number of objects of interest appearing in the 1tuples of concepts of arity 1, and HR found similar meta-concepts for concepts of arity 2 and 3. Also in experiment 3, HR made meta-concepts which were the parsimony measure specialised for concepts of particular arities.

In experiment 2, HR re-invented the variety measure, by using the size production rule to count the number of categories in a categorisation, and applying this function to the categorisation achieved by concepts. HR also re-invented the novelty measure by using the compose rule to invent the meta-concept of two concepts which share the same categorisation, followed by the size rule to invent the meta-concept of the number of other concepts sharing the same categorisation as a given concept. HR also invented the meta-concept of concepts for which all objects of interest are categorised together. HR identified that all the concepts of this type were user-given except the 18th concept in the original theory, which was this:

$$[\text{a, n}] : \text{animal(a)} \ \& \ n = |\{b : of\_class(a, b)\}|$$

This simply counts the number of classes to which the animals belong. As the answer is always one (because no animal is both a reptile and a mammal, etc.), all animals were categorised together.

Also in experiment 2, HR invented a new measure of interestingness for concepts, which had not previously occurred to us. HR defined this meta-concept:

$$[c, n] : \text{concept}(c) \ \& \ \text{exists } x \ (\text{catn } (c, x) \ \& \\ n = |\{y : cgy(x, y) \& 1 = |\{e : obj\_in\_cgy(y, e)\}|\}|$$

Upon examination, this meta-concept describes the following calculation: take a concept, c, look at the categorisation, x, that c achieves over the objects of interest and count the number of categories within x which have only one object of interest in them. For instance, the following concept in the original theory of animals:

$$[a, b, c] : \text{animal}(a) \ \& \ \text{of\_class}(a, b) \ \& \ \text{in\_habitat}(a, c)$$

achieved this categorisation of the 18 animals:

[[bat], [cat, dog], [crocodile], [dolphin, platypus], [dragon], [eagle], [eel, herring, shark, trout], [lizard, snake, t_rex], [ostrich], [penguin], [turtle]]

This concept looks at the classification of animals into mammal, fish, reptile and bird *alongside* the habitat the animal lives in. For this reason, bats are categorised apart from the other animals as they are the only mammal (in the 18 supplied) which live in caves. Similarly, crocodiles are the only reptile for which land and water are both habitats. The result of the calculation described in HR's meta-concept for this concept is therefore 7, because 7 objects are categorised in their own category, namely bat, crocodile, dragon, eagle, ostrich, penguin and turtle.

This new calculation for concepts – which we call the *speciality* of a concept — is interesting, because (as indicated by the name we assigned) it provides some indication of how specialised a concept has become. This measure is similar to the variety measure of a concept which, as discussed above, counts the number of categories in the categorisation achieved by the concept. In general, given a broad set of objects of interest, we find that the most interesting concepts produce a categorisation which is either binary — in which case the concept probably describes a specialisation of the objects of interest — or has many categories. In the latter case, whereas many categories are desirable, if this produces many categories containing single elements, the concept may be over-specialised. This suggests using the variety and speciality measures together, with the former weighted positively (so that HR prefers concepts achieving concepts with many categories in their categorisation of the objects of interest) and the latter weighted negatively (so that concepts which specialise so much that many objects appear in a category of their own are discriminated against). Therefore, HR has invented a new measure of interestingness which compliments one it has already, and we have now fully implemented this speciality measure in HR.

HR also invented some related meta-concepts, such as the specialisation of concepts into those which have a category in their categorisation with only one object in it, and the further specialisation into concepts with exactly 1 category in their categorisation which has a single object in it. HR identified that only 6 concepts in the original theory had this latter property, including the concept of the number of different habitat of birds. HR also invented another measure of interestingness: the number of categories in a concept's categorisation which contain more than one object (a compliment to the speciality measure).

Returning to the speciality measure, as well as providing us with a new way of measuring concepts in a theory, it also suggests an interesting way of ordering the objects of interest in a theory. Given a theory, T, we define the *distinctiveness* of an object of interest, I, to be the number of concepts in T which categorise I in a category of its own. This gives some indication of how distinctive an object of interest is: if it is classed apart from the other objects by many of the concepts in the theory, it must have some distinctive qualities. HR now presents the objects of interest in a theory ordered by their distinctiveness, which can provide a valuable insight into the theory. As an example, we produced a theory of animals using a breadth first search for 5000 steps. The theory contained 1204 concepts, and using these, HR pointed out that the animals are ordered in the following manner in terms of distinctiveness: eagle (278), bat (247), dolphin (209), dragon (194), crocodile (191), platypus (173), ostrich (156), pen-

guin (133), snake (124), turtle (107), cat (0), dog (0), eel (0), herring (0), lizard (0), shark (0), t_rex (0), trout (0).

The coefficients in brackets indicate the number of concepts which categorised the animal apart from all the other animals. Therefore, in this theory of animals, eagles are the most distinctive as they are categorised apart by 278 of the 1204 concepts. It was not obvious to us from the input file that eagles and bats are more distinctive (given the data supplied) than snakes and turtles. We also note that 8 of the animals are never categorised on their own. This is because there is another animal in the theory with all the same characteristics. For example, cats and dogs are not distinguished in the background predicates supplied by the user, hence they are not distinguished in the theory, as all the new concepts are based on the given ones. For this reason, HR did not invent the meta-concept of concepts which achieve a categorisation where all the categories contain single objects, rather it made the meta-conjecture that no such concepts exist.

We have also enabled HR to indicate to which other objects of interest a particular one is most related. We say that a concept relates two objects of interest if it categorises them in the same category. HR can count how many concepts in a given theory categorise two objects together. Thus, given a particular object, it can order the other objects in terms of the number of concepts which relate them to the given object. For example, in the theory of animals discussed above with 1204 concepts, eagle was most related to ostrich, as 884 concepts related these. After ostrich, eagle was most related to penguin and then bat with 809 and 467 concepts relating them respectively. We hope that in domains where the objects of interest are more intractable (possibly in bio-chemical domains), the distinctiveness measure may help identify outliers and ordering the objects in relation to each other may provide further insight into the original objects of interest.

HR did not re-invent the entire set of measures of interestingness which we have implemented. In particular, in equivalence conjectures, where one concept is conjectured to be equivalent to another, the surprisingness of the conjecture is calculated as the number of concepts which appear in the construction history of one the equivalent concepts, but not the other. This gives some indication of how different the left hand and right hand side of the equivalence conjecture is, and conjectures where the difference is more marked are more surprising. HR does not re-invent this measure of interestingness, but we believe it will do so when it has a new production rule, namely the path rule, which will construct concepts with recursive definitions (as discussed in (3)). Similarly, HR does not re-invent the complexity of a concept, which is calculated as the number of previous concepts in the construction history of a concept. Again, we believe this meta-concept will be re-invented when HR has the path production rule.

From these experiments, we can generalise the notion of a measure of interestingness to the notion of any meta-concept which calculates a numerical value for concepts. Our subjective judgements have led us to choose certain meta-concepts as those to implement as measures of interestingness in HR, but any similar meta-concept could be used to drive a heuristic search.

4.3 Higher Level Concept Formation

One of the initial motivations behind this investigation was the desire for HR to re-invent higher level concepts. The concept of 'function' is very important in mathematics, and we hoped that HR would re-invent this, for instance. We cannot report that HR has re-invented the notion of function yet, although in experiment 3 HR invented the concepts of functions of arity 2 and 3. In experiment 3, using the forall production rule, which introduces universal quantification, HR formed this concept:

[c] : concept(c) & (exists f s.t. 2tuple(c, e, f)
$\rightarrow (1 = |\{g : 2tuple(c, e, g)\}|)$)

This is the meta-concept of concepts of arity 2, which can be thought of as functions, i.e. there is a unique second object for each first object found in the pairs which make up the examples for these concepts. HR identified that the user-given concept of has_legs(A, B) was a function of arity two, because for each animal A, the number of legs, B, was unique. HR similarly invented the meta-concept of functions for which two objects input produce a unique output. It may be possible for HR to generalise these to the concept of function with either improved production rules, or possibly by advancing to another meta-level (theories about meta-theories) which would be an interesting experiment.

HR also invented many higher level concepts, including the following:

- Specialisation, i.e. a type of animal. This was simply the meta-concepts of concepts of arity 1, which must be a type of animal. HR also re-invented the meta-concepts of concepts with arity 2 and 3.
- User-given concepts, i.e. those which were not constructed, hence supplied by the user.
- Concepts produced by combining two previous concepts, i.e. produced using a binary production rule. Similarly, concepts produced using a unary production rule.
- Concepts produced by combining a previous concept with itself (i.e. using the previous concept twice in a binary production rule)
- Concepts produced by particular production rules.

While none of these concepts are particularly difficult or enlightening, they are fundamental to how the code for HR is written. If HR is ever going to contribute to its own development, such meta-concepts will be very important. We intend to further experiment by giving HR the background concepts of 'concept' and 'conjecture' for every theory formation session, so that meta-level notions such as 'function' can be developed alongside the concepts about the objects of interest. This is a possibility we intend to pursue in future meta-theory experiments.

5 Conclusions and Further Work

Our work with meta-theories is still very much in the development stage, and there is much work we intend to do. The results presented have been mixed: HR re-invented many measures of interestingness, but not all of them; it re-invented concepts similar to the notion of 'function' but not exactly this meta-concept; and it re-invented some concepts which enable it to reflect upon how it operates, but it did not identify any forbidden paths. We also have to be careful not to overstate any of HR's meta-theoretic inventions, and to assess our role in its results. For instance, as we chose to give HR information about categorisations, we clearly played an important part in HR's re-discovery of its measures of interestingness and its invention of the speciality measure. This indicates that the problem of forming and utilising meta-theories is very difficult and needs much further study. However, we have demonstrated that HR can form meta-theories, and this new information is potentially useful for improving HR's theory formation.

Perhaps the most important result is that meta-level functionality was given to HR in a simpler way than it was given to DENDRAL and AM. That is, HR formed the meta-theories in exactly the same way as normal theories, and we only needed to change domains, rather than write a new program (Buchanan wrote Meta-DENDRAL and Lenat wrote Eurisko to extend the DENDRAL and AM programs with meta-level abilities respectively). No major changes to how HR forms theories were made to HR in order for it to produce the meta-theories discussed above. HR does rely on a new ability to read Progol input files, but this work was completed as part of a project to place HR within the context of machine learning, a process described further in (6). The only addition we made was simply to enable HR to output predicates describing information from a theory to use as the background predicates for a meta-theory. Therefore, we can claim that it was probably easier to equip HR with meta-level abilities than AM and DENDRAL. Also, our approach also appears to be more extendible: there is no reason why HR should stop at the meta-level, and we have already suggested a possible application for meta-meta-level reasoning: for defining the function meta-concept.

Our experiments here emphasise how general HR is as a theory formation program: so general that it can form theories at a meta-level about how it works and what it produces. Lenat admitted in (10), page 61, that:

> '... it seemed straight-forward to simply add 'Heuristics' as one more field in which to let AM explore, observe, define and develop. That task ... turned out to be much more difficult than was realized initially ...'

By using HR with little modification to produce meta-theories, we have done with HR what Lenat couldn't with AM. However, we must emphasise that at present, HR does not utilise it's meta-theories as Eurisko did and we recognise that this may be difficult.

There are many different architectures for meta-theory formation that we intend to experiment with. At present, HR forms a meta-theory about a single theory, but we also intend to use a set of theories as the objects of interest in a meta-theory. We hope that the meta-concepts about theories will enable HR to reflect further upon the nature of the theories it produces. For instance, it may make the meta-conjecture that every theory contains concepts supplied by the user, and so on.

Another important possibility that we have not yet developed is for HR to invent new production rules. This will be difficult, requiring a meaningful way to represent what each production rule does in a way acceptable as input to HR. An intermediate step would be for HR to suggest sequences of theory formation steps. For example, HR might invent the meta-concept of concepts produced using the size production rule followed by the split production rule. Examples of such concepts include prime numbers, where we count the number of divisors, then specialise the concept of integer to integers where this coefficient is exactly two. If HR could identify useful sequences of theory formation steps such as this, we could enable HR to replace single steps with sequences of steps, so that it could make greater progress into the search space.

It is our aim to enable HR to utilise meta-level information to improve its theory formation with a similarly small amount of additional code. We intend to do this by using HR's existing multi-agent architecture as described in (5). We plan to have one agent produce a theory while another agent produces a meta-theory (in near-parallel, because some of the theory must have been formed before a meta-theory about it could emerge, so there will be a certain time lag between the formation of the theory and the formation of the meta-theory). As the meta-theory forms, the agent forming the original theory will read certain parts of the meta-theory and alter its search accordingly, a point we expand on below.

As discussed above, in programs such as AM and HR, measuring the quality of new concepts is an important functionality. Obviously, interestingness is a very subjective matter — which is why we allow the user to set the weights in HR's weighted sum for the overall measure of worth. Regardless of this, however, heuristic searches are useful because breadth first searches take too long to reach any complicated concepts (and the most interesting ones are often fairly complicated), but depth first searches produces theories which are over-specialised. In contrast, heuristic searches tend to produce theories which have complicated concepts, but are not overly specialised. We have observed that such desirable theories are produced largely regardless of which measures are actually used in the overall assessment of interestingness. This phenomenon was also noticed in AM's searches: on page 128 of (8), Lenat notes that:

> 'The general answer, then, is *No*, the initial settings of the Worth values are not crucial.'

Hence AM was fairly robust to changes in its evaluation function for the worth of concepts.

So far, we have supplied the code to facilitate measuring the interestingness of concepts. However, we have observed that (i) heuristics searches using measures of any nature in general produce more interesting theories than vanilla searches and (ii) HR re-invented many of the measures of interestingness as numerical meta-concepts in the meta-theories described above. This suggests that we employ a meta-theory to automate the task of deriving measures of interestingness. Much experimentation will be required, but, using a theory agent and a meta-theory agent as above, we initially propose the following scheme:

- Using a vanilla search, the theory agent produces a theory containing, say 100 concepts.
- Then the meta-theory agent produces a meta-theory containing, say, 10 meta-concepts which are numerical values calculated for each concept in the original theory.
- The theory agent reads these meta-concepts as measures of interestingness and uses them with equal weights in a weighted sum for the overall worth of concepts.

Using this new evaluation function, the theory agent will be able to change from its vanilla search to a heuristic search, which is desirable for reasons given above. This approach will reduce the subjectiveness due to the program's author/user deciding which measures to use in the evaluation function and how to weight the measures. We believe this will be an important step in making HR more creative, as it will not only invent new concepts, but invent new reasons why they are interesting.

As Buchanan pointed out in (1), meta-level abilities are very important in creative programs. By making us aware of the speciality measure discussed above, HR has already made a discovery which has improved it as a program. We hope that more experimentation with meta-theories will further improve HR's creativity, in terms of richer concept formation, more efficient search, new heuristics for evaluating concepts, new production rules for making concepts and in ways not yet obvious to us. Eventually, we hope that with meta-level abilities, HR will play an important part in its own development.

Acknowledgements

We would like to thank Bruce Buchanan for a very informative discussion about creativity and scientific discovery earlier this year in Pittsburgh. We would also like to thank Geraint Wiggins for once again organising a very interesting symposium on creativity in arts and science. The author is also affiliated to the Department of Computer Science at the University of York and this work is supported by EPSRC grant GR/M98012.

References

[1] B Buchanan. Creativity at the meta-level. In *Keynote speech at AAAI-2000*. Available on audio tape from the American Association for Artificial Intelligence, 2000.

[2] B Buchanan and E Feigenbaum. Dendral and Meta-Dendral: Their applications dimension. *Artificial Intelligence*, 11, 1978.

[3] S Colton. *Automated Theory Formation in Pure Mathematics*. PhD thesis, Division of Informatics, University of Edinburgh, 2001.

[4] S Colton, A Bundy, and T Walsh. HR: Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[5] S Colton, A Bundy, and T Walsh. Agent based cooperative theory formation in pure mathematics. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, 2000.

[6] S Colton, A Bundy, and T Walsh. Automatic identification of mathematical concepts. In *Machine Learning: Proceedings of the 17th International Conference*, 2000.

[7] 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.

[8] D Lenat. *AM: An Artificial Intelligence approach to discovery in mathematics*. PhD thesis, Stanford University, 1976.

[9] D Lenat. AM: Discovery in mathematics as heuristic search. In D Lenat and R Davis, editors, *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill Advanced Computer Science Series, 1982.

[10] D Lenat. Eurisko: A program which learns new heuristics and domain concepts. *Artificial Intelligence*, 21, 1983.

[11] D Lenat and J Brown. Why AM and EURISKO appear to work. *Artificial Intelligence*, 23, 1984.

[12] S Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[13] G Ritchie and F Hanna. AM: A case study in methodology. *Artificial Intelligence*, 23, 1984.

[14] D Smith, B Buchanan, R Engelmore, H Adlercreutz, and C Djerassi. Applications of Artificial Intelligence for chemical inference. *Journal of the American Chemical Society*, 95, 1973.