A Model of Lakatos's Philosophy of Mathematics

Alison Pease[1], Simon Colton[2], Alan Smaill[3] and John Lee[4]

1 Introduction

Lakatos attacked the view that mathematical knowledge is timeless, certain and *a priori* (Lakatos, 1976). Lakatos's work in the philosophy of mathematics is a controversial mathematical analogy of Hume's problem of induction combined with Popper's theory of falsification. That is, Lakatos both identified the problem of the impossibility of mathematical knowledge, and suggested a solution. His solution consisted of heuristic methods which guide the development of mathematical conjectures, concepts and proofs. These evolve through dialectic and analysis sparked by counterexamples. Counterexamples therefore, play a vital role in (Lakatos, 1976), though they are a starting, rather than finishing point: criticism has to be constructive if it is to be valuable.

This work has been described as a "masterpiece" (Kadvany, 2001, p.1); "the first major bridge between historical philosophy and serious mathematics" (Kadvany, 2001, p.14); a "brilliantly sustained tour de force" (Feferman, 1978, p.311); and "a philosophical and literary achievement of the stature of Hume on natural religion or Berkeley's Hylas and Philonous" (Hacking, 1981, p.135). Lakatos himself has been described as "one of the most original philosophers of science of the twentieth century" (Kadvany, 2001, p.1).

We have implemented aspects of Lakatos's theory as a computer model. We believe that doing so helps us to raise questions and identify ambiguities and gaps in the theory. The computational approach also allows us to propose and evaluate answers to these questions.

2 Lakatos's logic of discovery

Lakatos's work on philosophy of mathematics had three major influences. Firstly Hegel's dialectic, in which the *thesis* corresponds to a naive mathematical conjecture and proof; the *antithesis* to a mathematical counterexample; and the *synthesis* to a refined theorem and proof (described in these terms on [pp. 144-145](Lakatos, 1976)). Lakatos emphasises the dialectical aspect in the style of his book, which takes the form of a dialogue in a classroom. Thus he is able to represent different mathematical and philosophical positions by using the voices of different students. The role of the teacher in the book is to ensure that the discussion keeps moving and they do not get caught up in petty asides or dead end avenues. Secondly, Lakatos used Popper's ideas on the impossibility of certainty in science and the importance of finding anomalies. Lakatos argued that Hegel and Popper "represent the only fallibilist traditions in modern philosophy, but even they both made the mistake of preserving a privileged infallible status for mathematics" (Lakatos, 1976, p.139). Thirdly, Polya (Polya, 1954) and his work on mathematical heuristic — the study of the methods and rules of discovery and invention — was also a major influence; in particular his work on defining an initial problem and finding a conjecture to develop. Lakatos claims that his own work starts where Polya's leaves off.

Rather than being concerned with whether mathematical knowledge is possible (the argument between dogmatists – who claim that we can know – and sceptics – who claim that we cannot know, or at least that we cannot know that we know), or what type of knowledge it might be, Lakatos emphasised the importance of guessing. In (Lakatos, 1978), he argued that the important question is not *how do we know?*, but rather *how can we improve our guesses?* He presented a fallibilist approach to mathematics, in which proofs, conjectures and concepts are fluid and open to negotiation. Lakatos strongly criticised the deductivist approach in mathematics, in which definitions, axioms and theorem statements are presented with no explanation about their development, and considered to be eternal, immutable truths. Instead, Lakatos saw mathematics as an adventure in which — via patterns of analysis — conjectures and proofs are gradually refined but never certain. He warned that hiding this process makes the subject impenetrable to students and prevents experts from developing concepts or conjectures which may arise out of earlier versions of a theorem statement. Lakatos demonstrated his argument by presenting case studies of the development of Euler's conjecture that for any polyhedron, the number of vertices (V) minus the number of edges (E) plus the number of faces (F) is equal to two; and Cauchy's proof of the conjecture that the limit of any convergent series of continuous functions is itself continuous. (Lakatos, 1976) is a

[1]University of Edinburgh; alisonp@dai.ed.ac.uk
[2]Imperial College London; sgc@doc.ic.ac.uk
[3]University of Edinburgh; A.Smaill@ed.ac.uk
[4]University of Edinburgh; J.Lee@ed.ac.uk

rational reconstruction of the history of philosophy of mathematics as well as these two mathematical conjectures, tracing psychologism, intuitionism, rationalism, historicism, pragmatism, dogmatism, Kant's idea of infallible mathematics, refutationism, inductivism and deductivism. As one of the characters puts it (Lakatos, 1976, p.55), they discuss the packaging — the philosophical framework, as well as what's in the packet — the mathematical content.

Lakatos held an essentially optimistic view of mathematics, in which the process of mathematics traditionally thought of as impenetrable and inexplicable by rational laws — those which come down to lucky guess work or intuition, are seen in a rationalist light, thereby opening up new arenas of rational thought. He challenged Popper's view (Popper, 1972) that philosophers can form theories about how to evaluate conjectures, but not how to generate them, which should be left to psychologists and sociologists. He did this in two ways - arguing that *(i)* there *is* a logic of discovery, the process of generating conjectures and proof ideas *is* subject to rational laws; and *(ii)* the distinction between discovery and justification is misleading as each affects the other; *i.e.*, the way in which we discover a conjecture affects our proof (justification) of it, and proof ideas affect what it is that we are trying to prove (see (Larvor, 1998)). This happens to such an extent that the boundaries of each are blurred.

The first chapter of (Lakatos, 1976) was published as (Lakatos, 3 64); however the second chapter and the appendices of (Lakatos, 1976) were not published during Lakatos's lifetime, as he saw this work as an unfinished project (perhaps analogous to his view of mathematics). One drawback of this failure to publish is that he could not answer criticisms of the book. The fact that Lakatos never pronounced himself completely happy with his theory does, however, strengthen our argument that this work is worth implementing, as it implies that there may be gaps in the theory which we can hopefully identify and fill.

3 Why implement Lakatos's ideas?

Sloman (Sloman, 1978) argues that the computational paradigm provides new tools for understanding the processes which philosophers study, including the philosophy of mathematics. Thagard (Thagard, 1993) also emphasises that philosophy of science and artificial intelligence have much to learn from each other. In particular, we believe that since (Lakatos, 1976) was the first attempt to characterise informal mathematics (see (Corfield, 1997) and (Feferman, 1978)), it is likely to be incomplete, and hence be open to criticism and extension. We argue that, in accordance with the computational philosophy paradigm, implementing Lakatos's theory has enabled us to improve upon it. This dialogue format enables us to model social processes, and hence our implementation contrasts programmes such as BACON (Langley et al., 1987) and PI (Thagard, 1993) which model the thought processes of an individual. A further benefit of implementing Lakatos's ideas is that they suggest ways of improving the fields of automated theory formation and theorem proving (see § 8).

4 Lakatos's three main methods

Lakatos (Lakatos, 1976) explicitly outlines six methods for modifying mathematical ideas and guiding communication: surrender, monster-barring, exception-barring, monster-adjusting, lemma-incorporation and proofs and refutations. Of these, the three main methods of theorem formation are monster-barring, exception-barring, and the method of proofs and refutations (Lakatos, 1976, p.83). Crudely speaking, monster-barring is concerned with concept development, exception-barring with conjecture development, and the method of proofs and refutations with proof development. However, these are not independent processes; much of Lakatos's work stressed the interdependence of these three aspects of theory formation.

We are currently implementing all three main methods but for the purposes of this paper concentrate on the first two only, namely monster-barring and exception-barring.

4.1 The method of monster-barring

Monster-barring is a way of excluding an unwanted counterexample. This method starts with the argument that a 'counterexample' can be ignored because it is *not* a counterexample, because it is not within the claimed concept definition. Rather, the object is seen as a monster which should not be allowed to disrupt a harmonious theorem. For instance, one of the students suggests that the hollow cube (a cube with a cube-shaped hole in it) is a counterexample to Euler's conjecture, since $V - E + F = 16 - 24 + 12 = 4$. Another student uses monster-barring to argue that the hollow cube does not threaten the conjecture as it is not in fact a polyhedron. The concept polyhedron then becomes the focus of the discussion, with the definition being formulated explicitly for the first time; as 'a solid whose surface consists of polygonal

faces' (according to which, the hollow cube *is* a polyhedron), and 'a surface consisting of a system of polygons' (according to which, the hollow cube is *not* a polyhedron) (Lakatos, 1976, p.14). Using this method, the original conjecture is unchanged, but the meaning of the terms in it may change.

4.2 The method of exception-barring

Lakatos's treatment of exceptions is noteworthy for two reasons. Firstly, he highlights their role in mathematics — traditionally thought of as an exact subject in which the occurrence of exceptions would force a mathematician to abandon a conjecture. Secondly, Lakatos showed how exceptions, rather than simply being annoying problem cases, which we may be able to dismiss as monsters, can be used to further knowledge. *Piecemeal exclusion* is one way to deal with exceptions. It does this by excluding a type of polyhedron from the conjecture, in order to exclude a whole class of counterexamples. This is done by generalising from a counterexample to a class of counterexamples which have certain properties. For instance, the students generalise from the hollow cube to *polyhedra with cavities*, and then modify Euler's conjecture to 'for any polyhedra without cavities, $V - E + F = 2$'. Thus exceptions are seen as objects which are valid (as opposed to monsters) and force us to modify a faulty conjecture by changing the domain to which it refers. *Strategic withdrawal* is the only one of the methods which does not directly use counterexamples. Instead, it uses positive examples of a conjecture and generalises from these to a class of object, and then limits the domain of the conjecture to this class. For instance, the students generalise from the regular polyhedra to *convex polyhedra*, and then modify Euler's conjecture to 'for any convex polyhedra, $V - E + F = 2$'.

4.3 The method of proofs and refutations

As the title of the book suggests, this is the most important method, to the extent that the rest of the book is often seen by commentators and critics as a lead up to this method, for instance (Feferman, 1978). It starts off as the method of *lemma incorporation*, and is developed via the dialectic into the method of *proofs and refutations*.

Lemma incorporation works by distinguishing global and local counterexamples. The former is one which is a counterexample to the main conjecture, and the latter is a counterexample to one of the proof steps (or lemmas). A counterexample may be both global and local, or one and not the other. When faced with a counterexample, the first step is to determine which type it is. If it is both global and local, *i.e.* there is a problem both with the argument and the conclusion, then one should modify the conjecture by incorporating the guilty proof step as a condition. If it is local but not global, *i.e.* the conclusion may still be correct but the reasons for believing it are flawed, then one should modify the guilty proof step but leave the conjecture unchanged. If it is global but not local, *i.e.* there is a problem with the conclusion but no obvious flaw in the reasoning which led to the conclusion, then one should look for a hidden assumption in the proof step, then modify the proof and the conjecture by making the assumption an explicit condition.

Proofs and refutations consists of using the proof steps to suggest counterexamples (by looking for objects which would violate them). For any counterexamples found, it is determined whether they are local or global counterexamples, and then lemma incorporation is performed.

5 Implementing a model of Lakatos's theory

It is not our purpose here to give a detailed description of our system, which is a work in progress. Rather, we hope to use a brief description of the system to highlight questions about Lakatos's theory which our system has helped us to raise and to answer. Modelling Lakatos's theory has two benefits:

• the process of having to write an algorithm for the methods forces us to identify areas in which Lakatos was vague, and aspects he omitted; and

• running the model allows us to test hypotheses about the methods, for instance that they apply to scientific thinking, or that one method is more useful than another. These hypotheses may be claims that Lakatos or other commentators have made, or new ones.

In §'s 5.2 and 5.3 we discuss questions and answers which have arisen during our implementation of the methods of monster-barring and piecemeal-exclusion, *i.e.*, of type*(i)*; and in §6 we consider more general questions, of type *(ii)*.

5.1 System details

Our system is an extended version of the theory formation program HR (Colton, 2002). HR starts with objects of interest (*e.g.*, integers) and initial concepts (*e.g.*, divisors, multiplication and addition) and uses production rules to transform either one or two existing concepts into new ones. For example the production rule *size*, would take the concept 'divisors of an integer' and produce the Tau function 'number of divisors of an integer'. HR could then use the *split* production rule to produce the concept 'number of divisors of an integer = 2', *i.e.*, the concept of a prime number. The production rules are usually applied automatically, according to search strategies which the user inputs at the start of the run, but the user can also force the application of a production rule at any given time, in order to produce a desired concept. This is done by selecting one (or two) concepts in the theory, the production rule and the parameters which determine how the rule applies, and putting this step to be carried out at the top of the agenda. Forcing is a way of fast tracking: finding concepts which would eventually have been found automatically, to be found sooner.

All concepts are represented by a definition, data-table (giving the values of every object of interest in the theory for the given concept), and categorisation (in which all objects of interest with the same value are categorised together). For instance the concept *prime number*, with objects of interest 1-5 would be represented as the **definition:** a is an integer & $|\{b : b \text{ is an integer } \wedge b|a\}| = 2$; **data-table:** $1 = false; 2 = true; 3 = true; 4 = false; 5 = true;$ and **categorisation:** $[[2, 3, 5], [1, 4]]$. Conjectures, such as concept X implies concept Y, are made empirically by comparing the example sets of different concepts. For instance, HR has made the conjecture that if the sum of the divisors of n is prime, then the number of divisors of n is prime, by noticing that the data-table of the first concept is a sub-table of the second (Colton, 2002). HR also uses third party automated reasoning software in order to prove or disprove its conjectures. HR evaluates its conjectures and concepts using various ways of measuring interestingness (Colton et al., 2000), and this drives the heuristic search.

Our extended version is implemented in an agent architecture consisting of a number of students and a teacher, in keeping with the dialectical aspect of (Lakatos, 1976). Each agent has a copy of HR, and starts with a different database of objects of interest to work with, and different interestingness measures. Making the evaluation subjective agrees with Larvor's point[5] that Lakatos considered mathematics to be a matter of taste: "Why not have mathematical critics just as you have literary critics, to develop mathematical taste by public criticism?" *Gamma* — (Lakatos, 1976, p. 98). Students send conjectures, concepts, counterexamples, or requests such as barring a specific object of interest from the theory, to the teacher. The teacher sends requests to the students such as "work independently", "send a concept to cover counterexamples $[x, y, z]$", or "modify faulty conjecture C". The students use the methods prescribed by Lakatos to modify a faulty conjecture. Below we describe Lakatos's monster-barring and exception-barring methods.

5.2 Implementing and extending monster-barring

In (Lakatos, 1976), a series of definitions of polyhedron are suggested and negotiated. Students who want to defend the conjecture argue for definitions which *exclude* a proposed counterexample, or monster. Students who want to attack the conjecture argue for definitions which *include* a given counterexample, *i.e.* which would mean that the conjecture is false. The teacher resolves each such discussion by asking the class to accept the strictest definition, *i.e.* that which excludes the monster, leaving the conjecture open.

5.2.1 Which mathematical objects can be ambiguous?

In order to implement monster-barring we have to introduce ambiguity into our model, which forces us to answer questions about what sort of thing can be ambiguous. In mathematical theories, at least two types of component may be ambiguous — objects of interest and concepts. For instance, it may be ambiguous whether the object of interest $\aleph_0$ (the size of the set of all integers — the first transfinite number) is really a number or not; and it may be ambiguous whether the definition of the concept of prime number is 'any number with exactly two divisors', or 'a number which is only divisible by itself and one' (the difference being whether we consider the number 1 to be prime or not). The type of ambiguity can also take different forms. For instance, an object of interest could be ambiguous in two ways. Firstly, there may be two different objects with the same name, such as one object with six faces, eight vertices and twelve straight edges, and another object with six faces, eight vertices and twelve curved edges both being referred to as a cube and represented by the number of faces, vertices and edges. Secondly, there may be a single object which is represented in multiple ways, such the object star polyhedron being represented

[5] Personal communication.

as having twelve vertices, thirty edges and twelve faces (where a single face is seen as a star polygon), as well as having sixty triangular faces, forming thirty two vertices and ninety edges. The second case is described in Lakatos's method of monster-adjusting, which is where a counterexample is reinterpreted so that it no longer violates a conjecture — in this method subconcepts such as the concept *face* are shown to be ambiguous.

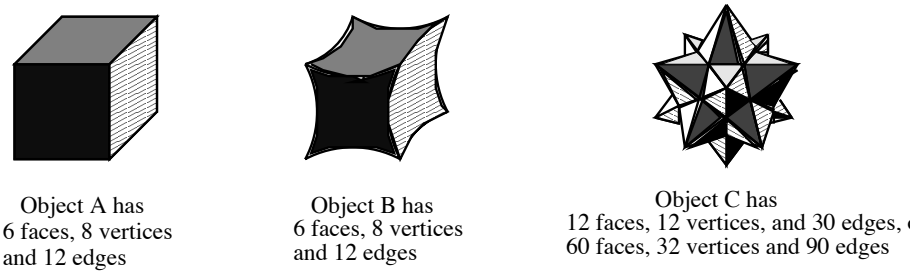

Figure 1: If we represent polyhedra in terms of the number of faces, vertices and edges, then objects A and B have the same representation, even though they are different objects, and object C has two different interpretations

In our implementation objects of interest and core concepts can be ambiguous. Students can question whether the definition of a core concept includes a specific object or not, for instance whether the concept of number should include zero, or whether a platypus is really an animal.

5.2.2 When should monster-barring be performed?

Implicit in the method of monster-barring is the fact that each party has a reason for wanting to define a concept in a certain way. This is a common phenomenon in everyday reasoning; for instance, politicians will define 'unemployment' or 'violent crime' differently, depending on whether they wish to argue that the figures have risen or fallen. In (Lakatos, 1976) the students discuss what was their original, unexpressed, intended definition and whether it corresponds to the explicit definition they are currently defending. Very little time is spent on why an alternative definition is first proposed, or why it may eventually be accepted or rejected; in (Lakatos, 1976) the teacher always instructs the class to accept the monster-barring definition, *i.e.* that which excludes the monster. This is a gap in the theory; always accepting the strictest definition is an unrealistic way of settling the dispute. As Corfield says; "Mathematicians have an intuitive feeling of the behaviour of objects they try to define - the process of discovery involves the struggle to find a good or 'right' definition" (Corfield, 1997, p.113). We have addressed this gap in our algorithm for the way in which students decide: *(i)* whether they want propose an alternative definition or not, and *(ii)* given a proposed new definition, whether they want to accept it or not. There are a range of motives which mathematicians have behind a decision to reject or accept a concept definition. In the context which Lakatos presents, the most obvious is to defend or attack a given conjecture (*i.e.*, Euler's conjecture). Another factor (which Lakatos does not consider) is the effect that choosing one of two competing definitions will have on the rest of a mathematician's theories or beliefs (this is known as the degree of *entrenchment* in belief revision (Gärdenfors, 1992)). In our implementation we have extended Lakatos's theory to reflect this.

Suppose that a student is sent an object of interest which is a counterexample to a conjecture that the group is currently discussing, and the object is new to the student. If this object is presented as an example of a concept C which the student is familiar with, then it is clear that there is some ambiguity over the definition of C. For instance, suppose a student receives the 'number' 0 when it has only previously seen positive examples of number (1,2, etc.). In this case the concept 'number' is ambiguous. The student then has two ways to decide whether it wants to bar the object (where the user decides which of the two ways should be used, at the start of the run). The first way is to test whether the counterexample breaks more than a (user-defined) percentage of all the conjectures in the student's theory, and if so, it proposes to monster-bar the counterexample. The second way is to test to see whether the new object is a 'culprit breaker'. This means that not only is the object a counterexample to the conjecture under discussion, but if it is allowed into the student's theory then it forces other objects in the theory which previously supported the conjecture, to become counterexamples to the conjecture. For instance, suppose that the conjecture under discussion is the conjecture that there do not exist integers a, b, c such that $a + b = c$ and $c|a$. A student which has the integers $1 - 10$ in its theory may make this conjecture, as it is true of all of its objects. However if the number 0 is proposed as a counterexample by another student, the first student will find that if it allows 0 into its theory, it does not just have a single counterexample – 0 – to this conjecture, but that the existence of 0 has forced all of the other objects to become counterexamples as well. For instance, if we take $b = 0$ and $a = c$, then 1 is a counterexample since $1 + 0 = 1$ and $1|1$, and similarly 2 is also a counterexample since $2 + 0 = 2$ and $2|2$, etc. So allowing 0 into the theory means that

there are now 11 sets of counterexamples a, b and c. In our algorithm, if the object in question forces other objects into being counterexamples for a higher number of conjectures than a minimum, user set proportion, then the object is called a 'culprit breaker', and monster-barred.

Rather than the teacher instructing the students to use the narrowest definition, once a concept has been raised as being ambiguous and two definitions suggested, each student decides which definition they prefer and votes accordingly. The definition is then decided democratically, based on these votes. If the votes are equal, then we follow Lakatos's principle of taking the narrowest definition. The students make the decision based on the proportion of conjectures in their theories which still hold under each of the rival definitions. Clearly this way of determining a definition means that we have to be able to accept the 'monster'. In our algorithm, if the consensus between the students is to extend a definition, then the teacher asks them all to perform monster-*accepting* by agreeing on the new, wider definition. (Lakatos, 1976, p. 83 - 99) does raise this issue, calling it concept stretching: however the discussion in this part of the book principally concerns the semantics and methodology of monster-barring rather than reasons for proposing and accepting a rival definition.

5.2.3 What sorts of definitions are proposed?

In (Lakatos, 1976), there are two types of concept definitions: an initial vague concept, which is not explicitly defined but some positives are known; and an explicit definition, for which the extension of the concept should be easier to determine. The two ways in which HR (Colton, 2002) can represent concepts corresponds to these — a core concept has no explicit definition, and a concept which HR has generated does.

The process of monster-barring in (Lakatos, 1976) might start with a vague definition and become more specific, or start with a specifically defined concept and by discussion reach agreement to define it in a different, yet still specific way. We have extended this by implementing the case which starts and finishes with a vague concept, *i.e.* a specific definition is not reached, but agreement is reached about whether a concept includes a given object or not. This is useful as explicit and precise definitions cannot always be reached nor agreed upon — even so called specific definitions are really just removing the level of vagueness. For instance defining a polyhedron as 'a solid whose surface consists of polygonal faces' (Lakatos, 1976, p.14) *is* being more explicit about what is meant by the concept polyhedron, but there is still ambiguity in the subconcepts solid, surface, polygon and face. We have implemented the process of generating a specific definition from a vague concept C, by finding all of the concepts which are conjectured to be equivalent to C, and then selecting the most interesting of these.

5.2.4 Our monster-handling algorithm

Students send each other objects of interest if they arise as counterexamples to a conjecture which the group is discussing. If a student is sent an object of interest which it has not seen before, as an example of a concept C, then it will first check to see whether its user-given flag to monster-bar is set. If it is not set, then the student simply adds the new object to its theory. If it is set, then the student:

1) decides whether to perform monster-barring;

It does this by performing one of the two tests above: if the new object is either a counterexample to more than a user-defined percentage of the student's total conjectures, or if it is a culprit breaker, then the student performs monster-barring. Otherwise it adds the new object to its theory and reject the conjecture under discussion as being false.

2) generates a new definition of the concept;

It can either generate a new, explicit definition of C, which excludes the monster; or it can generate a new, vague definition of C which excludes the monster from the list of objects which C covers.

When a student receives a proposal to bar a monster, it:

3) evaluates whether it agrees or disagrees with the proposal.

It does this by calculating the percentage of its own conjectures that the proposed 'monster' breaks, and voting on whether to bar the object or not.

When the teacher receives votes on barring or accepting an object, it waits until it has received a vote from each of the students, and then counts them. If the votes to bar the object outweigh those to accept

it, then the teacher tells the students to downgrade the 'monster' from object of interest to 'pseudo object' in their theories. Pseudo objects do not count as counterexamples so cannot threaten conjectures, but are around in the theory and can be upgraded to object at a future stage. If an explicit definition is given, then the students have to replace their old concept definition with the new one. Alternatively, if the votes to accept the object outweigh those to bar it, the teacher will tell the students to *add* the object to their theories.

5.2.5 Illustrative examples

As we have developed our system in number theory, we looked for instances of ambiguity in this domain, which we could model. The most ambiguous concept was that of number itself, with plentiful examples of monster-barring and eventual monster-accepting. For instance, the number 1 was initially barred by the Pythagoreans as it challenged their belief that all numbers increase other numbers by multiplication; $\sqrt{2}$ violated the Greek belief that all numbers describe a collection of objects; and $x = \sqrt{-1}$ violated the law that you cannot multiply two numbers together to give a negative number. Now of course 0, 1, irrational and imaginary numbers are accepted as numbers, and the concept of number has been generalised to complex numbers and beyond (quarternions). Another example is Cantor's introduction of transfinite numbers, which were not considered to be valid numbers by most mathematicians in Cantor's time, such as Kronecker. The 'number' $\aleph_0$, for instance, is a counterexample to the conjecture that if you add a non-zero number to another number, then the result is bigger than the second number, since $\aleph_0 + \aleph_0 = \aleph_0$. Similarly it violates the conjecture that any positive number multiplied by integer $n > 1$ is bigger than the number, as $\aleph_0.n = \aleph_0$. The law of monotonicity, that for all numbers a, b and c, if $b < c$, then $a + b < a + c$ fails if $a = \aleph_0$ (for any finite b and c). For these and other reasons, initial reaction to Cantor's work was hostile, and $\aleph_0$ was branded a monster, and barred from the concept of number. Today, however, these objects are accepted as numbers, and laws of arithmetic previously thought to hold for all numbers are now limited to a specified subset of number, such as the natural numbers or the reals. Clearly number theory, and other areas of mathematics, have been greatly enhanced by all of these additions.

Example to demonstrate the culprit breaker

The students debate whether the object 0 is a number or not.

Input information:
We ran the agency with two students and a teacher. The first student started with the integers $0-10$ and the other student started with the integers $1-10$ (*i.e.*, they did not know the number 0). Both students started with the background concepts of integers, divisors, and multiplication. The teacher requested non-existence conjectures, *i.e.* conjectures about a concept which has no known examples. The students were set to work individually for 20 steps and then enter into discussion. The students were both set to use monster-barring, and specifically to testing to see whether an entity was a culprit breaker when deciding whether to propose monster-barring or not. The monster-barring minimum was set to 15%, *i.e.* if a proposal to monster-bar an entity was made, a student would evaluate it by testing to see whether the entity was a counterexample to more than 15% of its conjectures (in which case the student would agree to bar it).

Results of run:
The second student made the conjecture that there do not exist integers a, b such that $b + a = a$ and $a + b = a$, and sent it to the teacher, who put it in the agenda for discussion. The teacher then asked for counterexamples to the conjecture, and the first student sent back all its integers, since having 0 in its theory meant that *every* number is a counterexample, e.g. 1 is a counterexample since $1 + 0 = 1$, similarly 2 is also a counterexample since $2 + 0 = 2$, etc. The teacher then asked for responses to the counterexamples, and the student with 0 tested to see whether there was a single 'culprit' entity which was forcing all of its objects of interest to be counterexamples, and concluded that 0 was a culprit entity. As a consequence it then sent the request to the teacher to monster-bar 0. The teacher put this request into the agenda, and sent it to the second student, who tested to see how many of its conjectures the number 0 broke. It found that 0 broke 63% of its conjectures, and as that was more than 15%, voted to monster-bar 0. The teacher counted the votes and told both of the students to down-grade 0 to a pseudo-entity. Both students then added 0 to their pseudo-objects of interest list, which meant that it was now in both of their theories but did not count as a valid integer.

Example to demonstrate the generation of an explicit concept definition

The students debate whether the object $A1 =< P(\{a, b\}) \setminus \phi, \cup >$ is a group or not. That is, $A1$ is an algebra consisting of the set of subsets of $\{a, b\}$ except the empty set, and the operation set union. Since the empty set is not included, this object has no identity and therefore no inverse.

Input information:
We ran the agency with three students and a teacher. The first student started with 14 examples of groups, up to size 8, and the other two students started with two algebras, $A1$, which is of size 3 and has no identity and no inverse for any of the elements, and $A2$, which is of size 4 and has an identity but no inverse element for two of the elements in it. All students start with the core concepts being an element of a group, the operator function, identity and inverse. In order to get the illustrative example, we forced the concept the existence of an identity element in a group, in the first student's theory. If we did not do this, it would find the concept automatically within 40 theory formation steps; we do it for the purposes of this example only. The monster-barring minimum is set to 15%, and the students are set to suggest an explicit definition if they perform monster-barring. The teacher asked the students to work individually for 40 steps and then send in their best conjectures.

Results of run:
The first student sent the conjecture that something is a group if and only if it has an identity element. The second and third students both sent the first algebra in their theories, $A1$ as a counterexample. As this is a new object for the first student, it checks how many of its conjectures it breaks and finds that it breaks 19%. Since this is greater than the minimum set, the student proposes to bar $A1$ as a monster and sends the statement that $A1$ is not a group, as a group must have an inverse element for every element in the group. The other two students check whether $A1$ is a counterexample to any of their conjectures, find that it is not, and reject the proposal to bar it, saying that it *is* a group, as a group is any set of objects with an element in it. As the first student is out-voted, the teacher tells it to add $A1$ to its theory.

It is in the method of monster-barring that the dialectic is most at play, where concepts develop from simple, often poorly understood, vague and ambiguous ideas to rich and sophisticated notions — analogous, as Larvor points out, to the dialectical pattern in Plato's Republic where they discuss and develop the concept of justice. This concept development is done by discussing propositions or conjectures in which the concept features — for instance that the *just* man is a happy man, or that for all *polyhedra*, V-E+F=2 ((Larvor, 1998, p.10)).

Lakatos thought that monster-barring was not a productive reaction to a counterexample; he calls it a 'usually barren Euclidean defence mechanism' (Lakatos, 1978, p.15), associated with dogmatists who defend or protect a conjecture at all costs. He criticises it because as well as specialising (hence reducing the domain — and therefore the value — of a conjecture), rather than generalising a conjecture, it results in long and complex definitions, whose history mathematicians have no idea about; this is especially unhelpful for students. (Lakatos gives the example of the definition of ordinary polyhedron which takes up 45 lines in the 1962 edition of the *Encyclopaedia Britannica* (Lakatos, 1976, p.53, footnote 3).) One advantage of this method of course, is that by excluding objects on the concept level rather than at the conjecture level, all conjecture statements involving this concept can be concisely expressed.

5.3 Implementing the method of exception-barring

5.3.1 Introducing a further distinction

Piecemeal exclusion in (Lakatos, 1976) always consists of barring a class of object. However, in our implementation we introduce a further distinction. We differentiate between concept-barring, where a concept is excluded, and counterexample-barring, where counterexamples are listed separately in the conjecture as exceptions. Using counterexample-barring, therefore, no overall concept is found which covers the counterexample(s). Introducing this distinction raises the question of *when* we should use concept-barring and when we should use counterexample-barring. We have resolved this by looking to see whether a student already has a concept in its theory which exactly covers the counterexamples, in which case it uses concept-barring. If unsuccessful, and there are few counterexamples, say $[x, y, z]$, it makes the concept X of being x, y or z and then uses counterexample-barring.

5.3.2 Other types of conjecture

Piecemeal exclusion in (Lakatos, 1976) is applied only to one conjecture, in particular that one concept (polyhedra) almost implies another (shapes which satisfy the Euler equation). We represent this as $poly(x) \rightsquigarrow euler(x)$. Clearly there are other types of conjecture than implication in mathematics. Equivalence conjectures are another type, in which the definitions of two concepts are logically equivalent ($P \leftrightarrow Q$). This raises the question of how to apply piecemeal exclusion to other types of conjecture such as equivalences, and what sort of conjectures would result. We answer these questions below.

As described above (§5), one way concepts are represented in the HR system (Colton, 2002) is as a data-table, *i.e.*, examples with corresponding values (eg *prime(5) = true*, or $\tau(5) = 2$). We can represent an

equivalence or implication conjecture as two sets of examples (corresponding to the two concepts) where the intersection contains those examples which share the same values. The example in (Lakatos, 1976) is represented as such in the left hand diagram in figure 2:

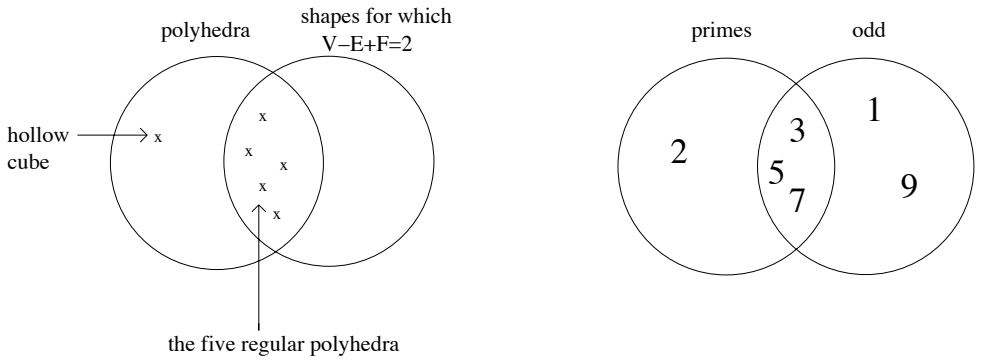

Figure 2: In the left hand diagram, we represent Euler's conjecture as overlapping sets. In the right hand diagram, we see represent the conjecture that primes $\leftrightsquigarrow$ odds as overlapping sets.

As this example is an implication we only have counterexamples on one side, *i.e.*, in one set. However, for equivalences, we may get counterexamples on both sides, as in the example on the right hand side of figure 2. In this case we might want to apply piecemeal exclusion *twice*. This would result in two implication conjectures. For instance, from the faulty conjecture "$\forall x.prime(x) \leftrightarrow odd(x)$", (using integers 1-10) suppose the system used counterexample-barring to produce the concepts *primes except 2* and concept-barring to produce the concept of *odd non-square numbers*. This would result in the generation of two conjectures: "all primes except 2 are odd" (which is true) and "all odd non-square numbers are prime" (which is false — the first counterexample is 15).

In addition to implication and equivalence conjectures, (Colton, 2002, chapter 7) suggests two further types: *non-exists*, where no examples satisfy the definition of a given concept ($\nexists x$ such that $P(x)$), and *applicability*, where examples satisfying a definition are restricted to a particular finite set (concept $P(x)$ is only applicable to examples in set S). We are currently considering the application of piecemeal exclusion to faulty conjectures in these formats.

5.3.3 Our piecemeal exclusion algorithm

Given a near equivalence $P \leftrightsquigarrow Q$, a student agent will get the list of counterexamples and determine whether each counterexample satisfies the definition of P or Q (using the set analogy, we say that a counterexample is *in* P or Q). There are three cases:

Case 1: all counterexamples are in P
(i) look for a concept X in the theory which exactly covers the counters. If unsuccessful, and there are few counterexamples, say $[x, y, z]$, then make the concept X of being x, y or z.
(ii) form the new concept $P \wedge \neg X$
(iii) add the new concept to the theory, which will force the formation of this conjecture: $P \wedge \neg X \leftrightarrow Q$

Case 2: all counterexamples are in Q
As for case 1, instead forming the concept $Q \wedge \neg X$ and the conjecture $P \leftrightarrow Q \wedge \neg X$

Case 3: there are counterexamples in both P and Q
(i) find a concept X in the theory which exactly covers the counters in P. If unsuccessful, and there are few counterexamples, say $[x, y, z]$, then make the concept X of being x, y or z.
(ii) form the new concept $P \wedge \neg X$
(ii) form the concept $P \wedge \neg X$
(iii) add this new concept to the theory, which will force the formation of this conjecture: $P \wedge \neg X \rightarrow Q$
(iv) repeat steps (i) to (iii), swapping P and Q.

If a student gets a near implication then there can only be counterexamples on one side, *i.e.* either in concept P or in Q (but not both), and so the steps in either case 1 or case 2 are performed.

5.3.4 Illustrative examples

Example to demonstrate concept-barring on a near-equivalence conjecture

Using three student agents and a teacher, we get the conjecture that an integer is non-square if and only if it has an even number of divisors.

Initial information
Student 1 starts with objects of interest $1-10$, and core concepts integer, divisor and multiplication. It is set to make implications from subsumptions, and to use piecemeal exclusion. Before running it independently we force the concepts square and integers which have an even number of divisors. Student 2 starts with the same input as Student 1, except with integers $11-50$. Student 3 starts with integers $51-60$, core concepts integer, divisor and multiplication, and the forced concept of integers which have an even number of divisors.

The run
The teacher sends a request to work independently for 20 steps, and then send back their best implication conjecture. Student 3 forms the conjecture that all integers have an even number of divisors, and sends it to the teacher, who puts it on the group agenda for modifications. The other two students then find counterexamples [1,4,9] and [16, 25, 36, 49] respectively, then find the concept of squares and form the new concept of non-squares. They then use this new concept to form the conjecture that all non-squares have an even number of divisors.

Example to demonstrate counterexample-barring on a near-implication conjecture

Using two students and a teacher, we get the conjecture that *all even numbers except 2 are the sum of two primes* (Goldbach's conjecture).

Initial information
Student 1 starts with the integers $1-10$ and core concepts integers and divisors. It is set to make implications from subsumptions and to use piecemeal exclusion. We force the concepts even numbers and integers which are the sum of two primes. Student 2 starts with the same information except integers $11-20$.

The run
The teacher sends a request to work independently for 20 steps, and then send back their best implication conjecture. Student 2 forms the conjecture that all even numbers can be expressed as the sum of two primes. The teacher sends a request for modifications, and Student 1 finds the counterexample 2, and makes the concept even numbers except 2 and the conjecture that all even numbers except 2 are the sum of two primes.

6 Answers suggested by the computational approach

Having a computational model enables us to test hypotheses about Lakatos's theory. These may arise during implementation or have been suggested by Lakatos or his critics. In this section we outline questions and criticisms of the theory, and ways in which our model has allowed — or might allow us — to answer them. In particular we refer to Feferman's ten criticisms of (Lakatos, 1976), in (Feferman, 1978, pp.316-320) (we note the criticism number in italics).

6.1 The scope of the methods

Although Lakatos is praised for the extremely detailed and in-depth analysis of his case studies — in particular of Euler's conjecture, he has been criticised (see, for instance, (Feferman, 1978)) for only considering two examples. Certainly it is difficult to claim to have found patterns general to mathematical and even other types of discovery from such a small sample. We see determining the scope of the methods as one of the major contributions of our work, and suggest an alternative to (Larvor, 1998, p.11) who claims that "this type of dispute [whether Lakatos's methods are typical or atypical of mathematical reasoning] can only be resolved by extensive historical research". Producing a computer model gives us an obvious way of testing the variety of domains to which the methods can be usefully applied.

It is worth noting that while Lakatos did claim that his methods are general enough to apply to other domains, both mathematical and non-mathematical, he did not believe them to be the sole explanation of mathematical discovery. Indeed, as Larvor argues, Lakatos did not believe that there is a unique logic

of mathematical discovery, much less that he had found it. (Larvor points out that it was the editors, rather than Lakatos, who gave the book the subtitle *the* logic of mathematical discovery.) Instead — in his original thesis at least — he states the more modest aim of pointing out 'some tentative rules which may help us to avoid some deeply entrenched wrong heuristic habits' (Thesis, p. 75; quoted in (Larvor, 1998, p.12)). This view, that there are many ways of practicing mathematics is echoed in philosophy of science — for instance Bird (Bird, 1998, chap 8) and Feyerabend (Feyerabend, 1975) argue that there is no such thing as *the* scientific method.

Applying the methods to other areas of mathematics

Feferman (Feferman, 1978) argues that the methods only explain a small fraction of mathematical reasoning. For instance, he argues that the main method — proofs and refutations — fails to account for foundational changes before 1847 *(i)*. He also questions Lakatos's claim that the method of proofs and refutations is most appropriate to young, growing theories *(ii)* — arguing that *(a)* Lakatos's main example of the method — Euler's conjecture — was not a young, growing theory, and *(b)* there are examples of young, growing theories - such as continuous probability measures — which progressed without recourse to counterexamples and therefore without recourse to Lakatos's method. Hacking (Hacking, 1981) argues that since Lakatos's reasoning assumes the hypothetico-deductive model, its relevance is restricted to this type of knowledge — and there are other styles of knowledge such as Crombie's six styles of reasoning (see (Crombie, 1994)). He warns us not to let "the eternal verities depend on a mere episode in the history of human knowledge" (Hacking, 1981, p.143).

Our implementation strategy has been to develop the methods in other mathematical domains, mainly number theory, but also group, ring and field theory, and to use the domain of algebraic topology as a test domain. In this way, we can ensure that we implement them in a general way. This also lets us investigate whether the methods are sufficiently general to produce interesting mathematics in areas other than topology, vector algebra and real analysis.

Applying the methods to empirical sciences

(Lakatos, 1976) is often seen as Lakatos's attempt to apply Popper's philosophy of science to mathematics. There are both methodological and epistemological parallels: the view that mathematics advances by studying refutations and therefore practitioners should focus on finding counterexamples and anomalies, and the belief that there is no certain knowledge — both mathematics and science are fallibilist. We have already noted one key difference between Popper and Lakatos — the discovery/justification distinction; another (pointed out by Larvor (Larvor, 1998)) is what we should do with the refutations once we have found them — Popper's naive falsificationism tells us to reject the hypothesis, whereas this reaction is the first and most naive method in (Lakatos, 1976) — the method of surrender — to which Lakatos devotes only one out of the total 120 pages in the book. More sophisticated methods use the counterexample to refine the conjecture and concepts in it.

Lakatos partially addresses this problem by inheriting Kuhn's ideas on demarcation. Kuhn argued (Kuhn, 1970) that the boundaries between scientific and non-scientific knowledge are not sharp; and Lakatos thought that the degree to which mathematics and science are the same type of (empirical) knowledge, corresponds to the degree to which his methods apply to science as well as mathematics. Lakatos claimed that "mathematical heuristic is very like scientific knowledge — not because both are inductive, but because both are characterised by conjectures, proofs, and refutations. The — important — difference lies in the nature of the respective conjectures, proofs (or, in science, explanations), and counterexamples" (Lakatos, 1976, p.74). He credits Polya's stress on the similarities between scientific and mathematical heuristic as one of the most important contributions of his work (see (Lakatos, 1976, p.74), footnote 1).

Feferman (Feferman, 1978) also asks *(x) what is distinctive about mathematics?* He argues that Lakatos's methods do generalise to other domains, that his logic of mathematical discovery is really a logic of *rational* discovery. However, Feferman sees this as a shortcoming, claiming that they would then be overly general and "could account only for a few gross features of the actual growth of mathematics"(Feferman, 1978, p.320).

In order to see whether our model could effectively be applied to non-mathematical domains, we tested it on a machine learning data-set. This consisted of 18 animals, with information on whether they were covered by hair, scales or feathers; the number of legs they have; whether they are homeothermic; whether they produce milk/lay eggs/ have gills; what sort of habitat they live in; and what class of animal they are — mammal, fish, reptile or bird. We ran our agency with two students, where the first was given the platypus as an example and the second student was not. The platypus arose instantly as a counterexample to a conjecture which the first student proposed, and the second student found that it violated many

of its conjectures, and requested that it be monster-barred. Upon examining its own conjectures, the first student agreed and narrowed its definition of animal to exclude this problem case. This is a nice example of how monster-barring works in non-mathematical domains. It mirrors the situation in the 19th century when the platypus was first brought from Australia to Britain, and was initially thought to be a hoax played by taxidermists. It was described by Darwin as a 'funny sort' (Mozley Moyal, 2001), and Burrell claims that "No animal has given rise to so much controversy among both layman and professed zoologists" (Burrell, 1927, p. 1). This example shows how we can extend Lakatos's work by testing his claims, for instance, that his methods do extend to empirical science.

6.2 Applying Lakatos's methods to other types of conjecture

Feferman (Feferman, 1978) points out *(vii)* that all the examples of conjectures given by Lakatos are of the form $\forall x[A(x) \rightarrow B(x)]$ and gives examples of other types of conjecture found in mathematics and the form their refinement might take. As we discussed in § 5.3.2, this is an aspect which has arisen in our implementation and we have suggested and implemented ways of applying the methods of other types of conjecture.

A related criticism is that, with the exception of strategic withdrawal, they are only applicable to the type of conjecture which could be falsified by counterexample (Feferman, 1978); *(ii)* and *(iii)*. This corresponds to the criticism of Popper's falsificationism (e.g., see (Bird, 1998)), that it only applies to scientific hypotheses which are generalisations. This excludes, for instance statistical hypotheses (nothing can falsify a probabilistic hypothesis). While we cannot avoid this criticism, we can investigate how limiting this is, by running the model on a wide variety of domains, and testing to see how many of the theorems and conjectures which are considered interesting in the field it generates.

6.3 How should we apply the methods?

How do the methods compare?

In both Lakatos's and commentators' writings, *e.g.* (Corfield, 1997), there is a clear hierarchy of methods, where they are presented as being increasingly sophisticated. Therefore much work focuses purely on the final method to be described — proofs and refutations. Indeed the first method, of surrendering a conjecture as false when faced with a counterexample, is not mentioned in any book on Lakatos that we have seen, and Lakatos wrote that "Mere 'falsification' (in Popper's sense) must not imply rejection" (Lakatos, 1981, p.116). He called exception-barring, monster-barring and monster-adjusting 'conventionalist strategems' (Lakatos, 1981, p.117), and thought that they are *ad hoc* in the sense that once applied, the new conjecture has no excess empirical content than the old one. Therefore they cannot form part of a progressive research programme (*i.e.* one which successfully predicts novel facts). Commentators usually make scant reference to these methods, with the assumption often being made that (Lakatos, 1976) is solely about the final method.

By running our system on different combinations of the methods and evaluating the resulting mathematical theories, we are able to investigate these views. For instance, we hold that surrender is useful in preventing resources from being wasted on dead end conjectures. The challenge is to find ways of knowing when to use a counterexample to surrender a conjecture, and when to use it as a catalyst for refining the conjecture. Similarly we hold that the method of exception-barring has enabled us to generate interesting conjectures (see §5.3.4); and by allowing us to explore concepts more fully, monster-barring can generate interesting discussion.

Our agency currently operates at three levels of increasing autonomy:

- the user sets flags which instruct student agents to use a given method;

- the teacher agent can request that the students use a given method, and

- students can choose themselves by deciding whether to defend or attack a given conjecture, based on how many counterexamples they have in their own theories.

We are investigating which method works best in a given situation by empirical testing.

When should we stop applying them?

Feferman claims that guesswork in mathematics finishes with the mathematician's successful struggle to solve a problem, as opposed to the picture which Lakatos paints of endless guessing *(iv)*. Certainly there

needs to be some limit on our system as to when it can apply each method. Conjectures which have been over-modified will become dull or too specific (for instance after repeated application of piecemeal exclusion). This will also prevent the system from investigating more interesting paths. The question of what exactly over-modified means now arises. Our system currently stops modifying a theorem either when no more counterexamples can be found, or when the theorem prover Otter (McCune, 1990) (to which it has recourse) has proved a conjecture. Additionally, the student agents are not allowed to attempt to bar an object if a vote has been agreed to accept it, *i.e.*, monster-barring is only allowed once per object. Each agent keeps track of the history of a modified conjecture, *i.e.*, which methods have been applied to it. We plan to use this history to put further checks on the number of times a student can perform a given method on the same conjecture. The best frequency for these checks will be determined by empirical testing.

7 Evaluating theories within the philosophy of science

In the philosophy of science, ideas on evaluating a theory or hypothesis came before ideas on how the hypothesis is discovered. In computational philosophy of science it appears to be the other way around. That is, ideas on how to generate, or discover programs which model scientific progress have come before ideas on how we should evaluate these programs. Clearly much work in the philosophy of science examines what makes a scientific theory good, for example (Popper, 1972). Yet, although much of philosophy concerns evaluating different arguments, there seems to have been little explicitly written about how to evaluate meta-theories, *i.e.*, philosophical theories about scientific theories. If the field of computational philosophy of science is to progress, there has to be discussion and agreement on the criteria by which we judge computational theories (philosophical theories which are at least partially derived using computational techniques). Additionally, these criteria must be formal enough that comparative claims can be supported and progress measured. Unsurprisingly, this mirrors the situation in the machine creativity field, in which attempts are being made to find a framework which is both practically useful and theoretically feasible, *i.e.,* formal but not oversimplified (for example (Pease et al., 2001), (Ritchie, 2001)).

We have not seen criteria set out purely for this purpose. For instance, Thagard (Thagard, 1993) suggests criteria for evaluating explanatory theories. These are intended for evaluating scientific, rather than philosophical theories, and have been extracted from studying examples of scientific theories. However, Thagard also claims that they can be used to determine the best explanation in metaphysical theories (Thagard, 1993, p.99). The criteria are consilience, simplicity and analogy. Consilience is a measure of how many observables a theory explains, and the variety and importance of the facts explained. The notion of simplicity is a way of constraining consilience by ensuring that the theory is not *ad hoc*. This means that the theory explains more than just the data which it was introduced to explain, *i.e.*, it is not fine-tuned. Hence the first and second criteria need to be taken in conjunction with each other. Lakatos does briefly use analogical reasoning in (Lakatos, 1976, p 70). This is where one of the students tries to find a conjecture which differs from Euler's conjecture to discuss, having decided that more discussion about the same conjecture will be fruitless. He recalls that the original conjecture was found by considering the theorem that for all polygons, the number of edges is equal to the number of vertices; and looking for an analogous relation in the polyhedra domain. However, as this idea is not central to the book, and HR already has the functionality to generate initial conjectures in other ways, we have not implemented this aspect yet. Hence Thagard's third criterion is not relevant to the evaluation of our extended theory.

Since we are extending Lakatos's work, our computational theory should explain everything his does and more. This makes the comparison easier as we do not have to show that our theory explains a wider variety or more important facts than those which Lakatos's explains. Rather, we need to show that it explains aspects of mathematical development which are omitted in Lakatos's theory.

Evaluating our computational model of monster-barring

Our addition of monster-accepting clearly explains aspects of theory formation which Lakatos did not address. This satisfies Thagard's criteria of consilience. In order to evaluate whether we have increased simplicity, *i.e.*, whether it explains more than the specific data in the domain of algebraic topology, we look to another mathematical domain which has arisen during our discussion, number theory, to see whether it applies there. Burton (Burton, 1985) asserts that the number zero, first appeared in the Western number system as a place holder in around 150 *A.D.* (in Babylonian positional notation, the number 1, for instance, would have been ambiguous as it could equally represent 10, 100, etc.). However, he states, it was not held to be a valid counting number for centuries, only being commonly used in practical calculations in the 1500s. This was partially due to the Greek reluctance to accept it — they branded it a

monster for various reasons, including its violation of the conjecture that if you add a number to another number, then the second number always changes. (When zero was eventually accepted as a number, this conjecture was modified to exclude zero, *i.e.* if you add a *non-zero* number to another number, then the second number always changes.) In §5.2.5 we saw a similar story with the introduction, initial hostility towards, ambiguous status and eventual acceptance of: the number 1, irrational and imaginary numbers, and $\aleph_0$. Thus it is clear that concepts within mathematics sometimes widen to include an object, rather than always narrowing to exclude it, and therefore our addition of monster-accepting not only satisfies Thagard's criteria of consilience but also his principle of simplicity.

Evaluating our computational model of piecemeal-exclusion

Implementing piecemeal-exclusion has led to a more fine-grained approach which differentiates between excluding counterexamples and excluding concepts. Further examples of conjectures in which counterexamples are explicitly excepted, as opposed to finding a concept which covers them and excepting that, include: all even numbers except 2 are expressible as the sum of two primes (this is Goldbach's famous conjecture); and all primes except 2 are odd. The first is an open conjecture (although discovered in 1742, it is still unproved), and the second is a theorem. Additionally we have seen examples in which applying the method to an equivalence, as opposed to an implication conjecture, suggests further conjectures of interest. Thus we argue that our extension has improved on Lakatos's original work.

We also hold that our implementation has enabled us to answer general questions about Lakatos's work. For instance, the question arose in §6.1 of whether his ideas translate to non-mathematical domains.

8 Other benefits of the computational approach

There are other benefits to implementing Lakatos's work (and other theories within the philosophy of science). For instance, we can use his ideas to help us to develop new techniques which aid scientists in their work (Langley, 2002). As with much work in the automated reasoning field, HR (Colton, 2002) was originally developed for this purpose, *i.e.*, to help mathematicians discover new results, and we consider that our extended version of automated theory formation contributes to this purpose. Research — for instance (Fielder, 2001) and (Langley, 2002) — has shown that scientists prefer to know the background of a claim made by a computer program - rather than take it on trust. This fits perfectly with Lakatos's philosophy of presenting the history of a result rather than isolated results with no explanation as to the thinking behind them.

A final motivation behind computational philosophy of science is to develop new techniques which are useful in finding new knowledge of interest to experts (Langley, 1999). Lakatos's ideas have provided us with this inspiration: we have developed a system — Theorem Modifier, or TM — which uses Lakatos-style methods to modify faulty conjectures into true ones. Given a conjecture, the system uses the Otter theorem prover (McCune, 1990) to first try and prove the conjecture. If it fails, the system uses the Mace model generator (McCune, 2001) to produce examples which support the conjecture, and examples which falsify the conjecture. We then use concept barring and strategic withdrawal methods implemented within the HR system to find concepts covering a subset of the falsifying examples and/or concepts covering a subset of the supporting examples. The system uses piecemeal exclusion on the first type of concept to withdraw into a more specialised conjecture, and strategic withdrawal on the second type of concept, again to specialise the conjecture. A set of modified conjectures is generated this way, and each is tested for theorem-hood by Otter. The user is shown only those modified theorems which Otter has proved.

Full details and results are reported in (Colton and Pease, 2004); here we give two illustrative examples to give a flavour of our results. In the TPTP library of first order theorems (Sutcliffe and Suttner, 1998), the first non-theorem in group theory states that, given the definition of the commutator operator on two elements x and y being $comm(x,y) = x * y * x^{-1} * y^{-1}$, then this operator is associative if and only if the product of the commutator is always in the centre of the group (defined to be the set of elements which commute with all others). Hence this theorem states that: $\forall\ x,y,z\ (comm(comm(x,y)),z) = comm(x,comm(y,x)) \Leftrightarrow \forall\ u,v,w\ (comm(u,v)*w = w*comm(u,v)))$. Mace could not find any counterexamples to this, but it did find four groups for which the conjecture is true. As strategic withdrawal does not need any counterexamples, TM could continue. It found that, with the extra axiom that the groups are self inverse (*i.e.*, $\forall\ x\ (x = x^{-1})$), the conjecture actually holds. As an example in ring theory, one non-theorem in (Sutcliffe and Suttner, 1998) states that the following property, P, holds for all rings: $\forall\ w,x\ ((((w*w)*x)*(w*w)) = id)$ where id is the additive identity element. Mace found 7 supporting examples for this, and 6 falsifying examples. HR produced a single specialisation concept which was true of 3 supporting examples: $\nexists\ b,c\ (b*b = c \wedge b+b \neq c)$. Otter then proved that P holds in rings for which HR's invented property holds. Hence, while TM could not prove the original theorem, it did prove that, in rings for which $\forall\ x\,(x*x = x+x)$, property P holds. The specialisation here has an appealing symmetry.

9 Conclusion

Lakatos's philosophy of mathematics provides us with a rich source of ideas on how mathematical theories can evolve. We have argued that the process of implementing his philosophy of mathematics has forced us to ask searching questions, such as: what types of object can be ambiguous; how ambiguity arises and how it is resolved; how we can decide between rival definitions, and how to widen a definition; what sorts of things can usefully be barred in a conjecture statement, and how we might apply the methods to types of conjecture other than implications; and to suggest answers. This process has led to an extended computational theory of mathematical development. Running the model has enabled us to answer questions about the applicability of the methods to other mathematical, and non-mathematical domains.

This project is a work in progress and we hope to implement further methods, increase the sophistication of the agents, and test more hypotheses. Our goals include testing to see whether the age of the theory affects the efficiency of the methods[6]. For instance, a new concept which is suggested in monster-accepting may break the current conjecture but show the promise of exciting new theories. HR records the number of steps is has been running for, so the age of its theories is easily determined. We are also currently completing our implementation of the method of proofs and refutations.

Given that Lakatos emphasised the informal nature of mathematics, our attempt to implement and therefore formalise it could be seen as being as objectionable as the editors addition of the 'final' chapter (chapter 2) in the history of Euler's conjecture — thus presenting his work as a finished philosophy rather than a step on the path of Hegel's dialectic. Our justification for this is that we by no means finalise the work, we investigate which parts can be formalised and whether and how that adds to Lakatos's work. We do not see modelling informal mathematics as paradoxical, rather as a contribution both to philosophy in terms in investigating what can be formalised and how that affects a theory, and to automated theory formation and mathematical reasoning by modelling mathematics as it is actually done by humans.

Lakatos's is an essentially optimistic doctrine: he believed that knowledge does grow, and the growth of knowledge provides a demarcation between rational and irrational thought. His Hegelian influences, that any methodological precept is open to revision, suggest that we can apply his philosophy to his own philosophy, as well as to mathematics and science. We believe that developing his theory by implementing it as a computer model provides a new and exciting perspective on his work.

Acknowledgements

Thanks to everyone at the ECAP conference for interesting and helpful discussion, in particular to Brendon Larvor.

References

Bird, A. (1998). *Philosophy of Science*. Routledge, London.

Burrell, H. (1927). *The Platypus*. Angus and Robertson Ltd., Sydney.

Burton, D. (1985). *The History of Mathematics*. Allyn and Bacon, Inc, Boston, USA.

Colton, S. (2002). *Automated Theory Formation in Pure Mathematics*. Springer-Verlag.

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

Colton, S. and Pease, A. (2004). The TM system for repairing non-theorems. In *Workshop on Disproving, Proceedings of IJCAR'04*, pages 13–26.

Corfield, D. (1997). Assaying Lakatos's philosophy of mathematics. *Studies in History and Philosophy of Science*, 28(1):99–121.

Crombie, A. C. (1994). *Styles of Scientific Thinking in the European Tradition: the History of Argument and Explanation especially in the Mathematical and Biomedical Sciences and Arts*. Gerald Duckworth & Co, Ltd.

[6] Thanks to Brendon Larvor for this suggestion.

Feferman, S. (1978). The logic of mathematical discovery vs. the logical structure of mathematics. In Asquith, P. D. and Hacking, I., editors, *Proceedings of the 1978 Biennial Meeting of the Philosophy of Science Association*, volume 2, pages 309–327. Philosophy of Science Association, East Lansing, Michigan.

Feyerabend, P. (1975). *Against Method*. Verso, London.

Fielder, A. (2001). Dialog-driven adaptation of explanations of proofs. In Nebel, B., editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1296–1300, Seattle, WA. Morgan Kaufmann.

Gärdenfors (1992). *Belief revision*. Cambridge University Press, Cambridge.

Hacking, I. (1981). Lakatos's philosophy of science. In Hacking, I., editor, *Scientific Revolutions*, pages 128 – 1443. Oxford University Press, Oxford.

Kadvany, J. (2001). *Imre Lakatos and the Guises of Reason*. Duke University Press, Durham and London.

Kuhn, T. (1970). *The Structure of Scientific Revolutions*. The University of Chicago Press, Chicago, USA.

Lakatos, I. (1963-64). Proofs and refutations. *The British Journal for the Philosophy of Science*, 14(53-56).

Lakatos, I. (1976). *Proofs and Refutations*. CUP, Cambridge, UK.

Lakatos, I. (1978). Infinite regress and foundations of mathematics. In Worral, J. and Currie, G., editors, *Mathematics, Science and Epistemology*, pages 3 – 23. Cambridge University Press, Cambridge.

Lakatos, I. (1981). History of science and its rational reconstructions. In Hacking, I., editor, *Scientific Revolutions*, pages 107 – 127. Oxford University Press, Oxford.

Langley, P. (1999). The computer-aided discovery of scientific knowledge. In *Proceedings of the First International Conference on Discovery Science*, Fukuoka, Japan. Springer.

Langley, P. (2002). Lessons for the computational discovery of scientific knowledge. In *Proceedings of First International Workshop on Data Mining Lessons Learned*, pages 9–12, Sydney.

Langley, P., Simon, H., Bradshaw, G., and Żytkow, J. (1987). *Scientific Discovery*. Cambridge, MA: MIT Press/Bradford Books.

Larvor, B. (1998). *Lakatos: An Introduction*. Routledge, London.

McCune, W. (1990). The OTTER user's guide. Technical Report ANL/90/9, Argonne National Laboratories.

McCune, W. (2001). Mace 2 Reference Manual. Technical Report ANL/MCS-TM-249, Argonne National Laboratories.

Mozley Moyal, A. (2001). *Platypus: The Extraordinary Story of How a Curious Creature Baffled the World*. Smithsonian Institution Press.

Pease, A., Winterstein, D., and Colton, S. (2001). Evaluating machine creativity. In Weber, R. and von Wangenheim, C. G., editors, *Case–Based Reasoning: Papers from the Workshop Programme at ICCBR'01*, pages 129–137. Washington, DC: Naval Research Laboratory, Navy Centre for Applied Research in Artificial Intelligence.

Polya, G. (1954). *Mathematics and plausible reasoning*, volume Vol. 1, Induction and analogy in mathematics. Princeton University Press.

Popper, K. R. (1972). *Objective Knowledge*. OUP, Ely House, London.

Ritchie, G. (2001). Assessing creativity. In Wiggins, G., editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, pages 3 – 11. SSAISB.

Sloman, A. (1978). *The Computer Revolution in Philosophy*. The Harvester Press, Ltd.

Sutcliffe, G. and Suttner, C. (1998). The TPTP problem library: CNF release v1.2.1. *Journal of Automated Reasoning*, 2(21):177–203.

Thagard, P. (1993). *Computational Philosophy of Science*. MIT Press, Cambridge, Mass.