The Use of Classification in Automated Mathematical Concept Formation

Simon Colton, Stephen Cresswell and Alan Bundy
Department of Artificial Intelligence
University of Edinburgh
(simonco@dai.ed.ac.uk, stephenc@dai.ed.ac.uk, bundy@dai.ed.ac.uk)

Abstract

Concept formation programs aim to produce a high yield of concepts which are considered interesting. One intelligent way to do this is to base a new concept on one or more concepts which are already known to be interesting. This requires a concrete notion of the 'interestingness' of a particular concept. Restricting the concepts formed to mathematical definitions in finite group theory, we derive three measures of the importance of a concept. These measures are based on how much the concept improves a classification of finite groups.

Introduction

One approach to automatic mathematical concept formation is to perform a heuristic search through a space of sentences which define mathematical concepts. In the space, there will be some sentences which are rubbish, some which are plausible but not very exciting, and some which are important. In order to be able to do an effective search, reducing the number of rubbish sentences, and increasing the yield of important concepts, it is necessary to have a notion of 'interestingness,' which can provide a reason to accept or reject a newly formed concept. We can then use the heuristic of preferring paths in the search space from an interesting concept, and discouraging paths from uninteresting concepts in the hope that interesting concepts lead to further interesting concepts.

The most cited work in automated mathematical concept formation is Lenat's work on his AM[1] computer program which invented new definitions and made conjectures based on empirical evidence. This was an exploratory program designed to work in elementary set theory which actually delved into elementary number theory. The notion of interestingness was used to maintain an agenda of tasks to do next, but was based on many things and is difficult to pin down.

Other attempts to formalise the notion of interestingness concentrated on reducing the search space so that exploration only occurred in a narrow band of concepts. With each of the newly formed concepts having a similar nature, it is easier to make a comparison of two concepts, and interestingness can be measured more easily. For instance, in the BACON[2] programs written by Langley et al, the concepts formed were polynomial relations between variables in physical systems. A concept was interesting if the polynomial relation was observable in the data, and uninteresting if not. In this case, they were able to guarantee some level of interestingness by looking at the data first, spotting patterns and trends and forming the new relations in this data driven manner.

Another example of narrowing the search space is Sims' IL[3] program, which accepted specifications for an operator (for instance the multiplication operator on Conway numbers), and formed concepts towards this goal. The number of the specifications satisfied by the invented concept gives an indication of its importance, and the process can stop once one has been invented which satisfies all the specifications.

Therefore, finding an effective measure of interestingness of concepts is a two step process; imposing a similar format between all the concepts produced, and then defining why one concept in this general format is better than another in the same format.

To begin to impose a similar format between the concepts, we first restricted the mathematical domain in which the concepts were formed to finite group theory. This choice was due to our mathematical background and the fact that many other algebras, such as ring theory, Galois theory and infinite group theory depend on the concepts from finite group theory. We further restricted the type of concept formed to the definitions in finite group theory, loosely speaking, the sentences which appear under the 'definition' heading of finite group theory texts. As detailed later, more similarity is added to these sentences by thinking of them all as functions mapping a group to some output.

Having decided to work with the definitions from finite group theory, it is possible to state that a major reason why the theory was constructed in the first place was to classify finite groups. This has historical backing, as in 1980, the classification of finite simple groups was achieved, which must surely be regarded as one of the major intellectual achievements ever.[4] Finite simple groups can be used to construct any finite group, in a similar way to prime numbers being used to construct any integer. Hence the classification of finite simple groups goes a long way to classifying finite groups, and the classification of groups is a major driving force behind the formation of concepts in group theory.

If the reason to form concepts is to produce a classification of finite groups, we can judge how important a particular concept is by how much it improves a classification. By thinking of a classification as a way to describe objects, it is possible to specify three intrinsic measures of how good a particular classification is. These amount to determining;

[1]See [Davis & Lenat 82].

[2]See [Langley *et al* 87].

[3]See [Sims 90].

[4]To quote John Humphreys in [Humphreys 96]. For details of the classification of finite simple groups, see [Gorenstein 82].

- How well the descriptions differentiate between two objects.
- How closely each description represents the object.
- How succinctly the descriptions can be stated.

The work here develops calculations which can be performed to measure these aspects of a classification, and how much of an improvement a particular concept makes. This allows us to associate three values to each concept, the *acuity improving value,* the *representation improving value* and the *representation space expansion value.* These can be used to measure the interestingness of any concept in the theory.

Note that, from this point, we will abbreviate 'finite group' to just 'group'.

Classifications

The first step to introducing a formal notion of interestingness is to add some homogeneity to the concepts so that the similar structure makes it easier to compare two concepts. As previously stated, the concepts being formed here are definitions in group theory. The three most common ways of introducing new definitions in group theory books are as follows:

A) By specialising the concept of group using a black and white test on the group. For instance: "A group, G, is Abelian if and only if $\forall\ a, b \in G, a * b = b * a$."

B) By specifying a calculation which can be performed on the group. For instance: "The centre of a group, G, is given by

$$Z(G) = \{a \in G : \forall\ b \in G, a * b = b * a\}."$$

C) By detailing a construction which is possible from a group. For instance: "Given a group, G, then H is a subgroup of G if and only if $H \subseteq G$ and H forms a group itself."

It is possible to think of the first and last formats in terms of the second. Firstly, the black and white test on groups can be written as a boolean function, which takes a group as input, and outputs 'true' or 'false'. For instance, we could write the Abelian property of groups given above as:

$$IsAbelian(G) = \begin{cases} \text{true} & \text{if } \forall\ a, b \in G, a * b = b * a \\ \text{false} & \text{otherwise} \end{cases}$$

Secondly, to use constructions as a function on a group we can make a function which outputs all the possible examples of the construction for a given input group. Eg.

$$SubgroupCollection(G) = \{H : H \text{ is a subgroup of } G\}.$$

Then, as soon as a construction definition is given, we can make this second, functional, definition which can be used to assess the construction definition.

Note that in practice there are other things we can do with the construction definition. For instance, we could make a function which tests whether a given group has any examples of a particular construction. Eg.

$$HasSubgroup(G) = \begin{cases} \text{true} & \text{if } \exists\ H \text{ such that} \\ & H \text{ is a subgroup of } G \\ \text{false} & \text{otherwise} \end{cases}$$

Whatever we choose to do with the construction, the result is always a function which takes a group as input and gives some kind of output. Hence all the sentences given in the definitions can be written as functions taking a single group as input, and outputting something based on the group. To add more similarity to the format of the concept, we make the restriction that this output is a nested vector. This is true anyway of the functions made from definitions of type A and C above, and we only have to worry about those of type B, which are the functions occurring naturally in group theory. As group theory is built heavily on set theory, most of the outputs are elements, sets or groups, all of which can be written as nested vectors, and in practice this imposition is not too restrictive.

We now have a starting point for a formalisation, and can continue by noting that the output of a function can be used to describe any input group. For instance, given a particular group, G, instead of saying "G is Abelian", we can say: "$IsAbelian(G) = \text{true}$". Further, a set of such functions can be used to build a more complete description of groups. We can then use these descriptions to classify individual groups within a set of groups. This can be formalised as follows;

Definition 1

- A **classifying function** in group theory is a unary function which takes a single group as input, and outputs a nested vector.
- A **classifying theory** of groups, C, is a vector of (the names of) classifying functions, $C = < f_1, \ldots, f_k >$.
- Given a group, G, then the **description of G by C,** represented by $desc_C(G)$, is the vector of output values given when G is used as input to all the functions in C. ie.

$$desc_C(G) = < f_1(G), \ldots, f_n(G) > .$$

- Given a set of groups, S, **the classification of S by C,** represented by $class_C(S)$, is the set of descriptions of the members of S. ie.

$$class_C(S) = \{desc_C(G) : G \in S\}.$$

Example 1

The following are classifying functions in group theory:

$Order(G) = |G|$ *[The size of the underlying set].*

$$IsAbelian(G) = \begin{cases} \textit{true} & \textit{if } \forall\ a, b \in G, a * b = b * a \\ \textit{false} & \textit{otherwise} \end{cases}$$

$NSIE(G) = |\{a \in G : a * a = 1\}|$

($NSIE$ stands for the Number of Self Inverse Elements).

---

and using $S = \{G_1, G_2, G_3, G_4\}$, with

G_1	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

G_2	1	2	3	4
1	1	2	3	4
2	2	3	4	1
3	3	4	1	2
4	4	1	2	3

G_3	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	3	4	5	6	1
3	3	4	5	6	1	2
4	4	5	6	1	2	3
5	5	6	1	2	3	4
6	6	1	2	3	4	5

G_4	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	1	5	6	3	4
3	3	6	1	5	4	2
4	4	5	6	1	2	3
5	5	4	2	3	6	1
6	6	3	4	2	1	5

it is easy to calculate:

$$\begin{array}{lll} Order(G_1) = 3 & IsAbelian(G_1) = \text{true} & NSIE(G_1) = 1 \\ Order(G_2) = 4 & IsAbelian(G_2) = \text{true} & NSIE(G_2) = 2 \\ Order(G_3) = 6 & IsAbelian(G_3) = \text{true} & NSIE(G_3) = 2 \\ Order(G_4) = 6 & IsAbelian(G_4) = \text{false} & NSIE(G_4) = 4 \end{array}$$

So, if we let $C = \{Order, IsAbelian, NSIE\}$, we get:

$$desc_C(G_1) = < 3, \textit{true}, 1 >, \; desc_C(G_2) = < 4, \textit{true}, 2 >,$$
$$desc_C(G_3) = < 6, \textit{true}, 2 >, \; desc_C(G_4) = < 6, \textit{false}, 4 >,$$

and this gives us:

$$class_C(S) = \{< 3, \textit{true}, 1 >, < 4, \textit{true}, 2 >, < 6, \textit{true}, 2 >, < 6, \textit{false}, 4 >\}.$$

Measures of Importance

Having defined a classification of a set of groups, we can now find some measurable properties of it, and compare the classification given by a single function against that given by a set of functions, to gauge the importance of the single function. It is therefore necessary to identify what is desirable in a classification. To help with this, there are two special classifications which represent the worst and best cases. Firstly, there is the trivial classification, which describes each group with the same sentence, "G is a group". Secondly, there is the explicit classification which describes each group by giving its group table.

Acuity of a Classification

If we look at the trivial classification, we see that it describes every group in the same way, which is clearly a defect. This identifies the first purpose of a classification of groups, to describe them, and it would be desirable to have a classification which describes each group differently. We can approximate how well the descriptions differentiate between any two groups by looking at a set of groups, and judging how many have different descriptions. This leads to the following definitions. (Note that each measure introduced is normalised to give a value between 0 and 1, with 0 being the worst case, and 1 being the best case.)

Definition 2

• Given a set of m groups, S, and a classifying theory, C, then a particular group, $G \in S$, is **uniquely identified in S by C** if there is no other group, $H \in S$, which has the same description by C as G. ie. $\nexists H \in S$ s.t. $desc_C(H) = desc_C(G)$.

• If all the groups in S are uniquely identified by C, then we say that C forms a **complete classification of S.**

• The **acuity of a classifying theory** is a measure of the number of groups which are uniquely identified by C. We can approximate this using S with the S_acuity of C, which is given by:

$$S\_acuity(C) = \begin{cases} 1 & \text{if } m = 1 \\ \frac{|class_C(S)|-1}{m-1} & \text{Otherwise} \end{cases}$$

[Note that $class_C(S)$ does not necessarily have m members, because if two groups have the same description using C, then as $class_C(S)$ is a set, the description will only appear once in the set, making $|class_C(S)| < m$.]

• Given a classifying function, $f \in C$, then the **acuity improving value of f in C approximated using S,** represented by $S\_aiv_C(f)$, is a measure of the acuity of the function if considered alone. It is given by:

$$S\_aiv_C(f) = \begin{cases} \frac{S\_acuity(<f>)}{S\_acuity(C)} & \text{if } S\_acuity(C) > 0 \\ 0 & Otherwise \end{cases}$$

Note that by writing the description of the group instead of its group table, we are actually writing it in a code, which provides a link between this work and information theory.[5] The information source entropy of a code measures the average probability of a codeword occurring, and as such is similar to the acuity measure given above. As we are thinking of the descriptions as a classification of groups, rather than a method of coding them, we use the acuity terminology instead of referring to the entropy of a code.

Example 2

Using S and C from the previous example, we see that each G_i has a different description using C, and so they are all uniquely identified by C. As this is the ideal case, we should find that the acuity of C on S is 1. We can check this:

There are 4 groups in S, so $m = 4$, and we have already calculated $class_C(S)$, which had 4 elements.

Hence $S\_acuity(C) = \frac{|class_C(S)|-1}{m-1} = \frac{4-1}{4-1} = 1,$

and we see that C forms a complete classification of S. We can use this to work out the acuity improving values of the three classifying functions in C. Noting that

$$class_{<f>}(S) = \{f(G) : G \in S\},$$

we see that

$$class_{<Order>}(S) = \{3, 4, 6\},$$
$$class_{<IsAbelian>}(S) = \{\textit{true, false}\},$$
$$class_{<NSIE>}(S) = \{1, 2, 4\}.$$

Hence $S\_acuity(< Order >) = \frac{|class_{<Order>}|-1}{4-1} = \frac{3-1}{3} = \frac{2}{3},$

and with similar calculations, we find that

$$S\_acuity(< IsAbelian >) = \tfrac{1}{3}, \; S\_acuity(< NSIE >) = \tfrac{2}{3}.$$

So we can calculate:

$$S\_aiv(Order) = \frac{S\_acuity(<Order>)}{S\_acuity(C)} = \frac{\frac{2}{3}}{1} = \frac{2}{3}$$

and similarly,

$S\_aiv(IsAbelian) = \frac{1}{3}$ *and* $S\_aiv(NSIE) = \frac{2}{3}.$

[5] As described in [Pierce 80].

Representation Content of a Classification

Look now at the explicit classification of groups, where each group is described by giving its group table. From this description, it is possible to read the underlying set of elements, and we can find the product $a * b$ for every a and b in the group. As groups are abstract entities themselves it is not obvious that this description presents the group in full, but this is the case. Given the group table, we know (or can find out) everything about the group, and any description from which we can read the underlying set and group operation $*$ gives an explicit classification of groups.

Hence the second purpose of a classification is to represent groups. Some representations, like the group table, are explicit, and the underlying set and group operation can be written down with no work. Other representations will require a search to find the underlying set and group operations (or equivalently a search to find the group table). Using the description as constraints on the search through possible group tables, one description might rule out a lot of possibilities, making the search manageable, but another might not rule out so many possibilities, making the search less efficient. The size of this search can therefore be used to measure how much information the descriptions contain about the groups they describe, and hence how well they represent the groups.

Note that if we know the order of the group, a search for the group table of the group will be finite, and, while losing generality slightly, things are made much easier if we assume that the order of the group is a constraint which is always given in the description. Given the assumption that the order of the group is known to be n, one implausible way that a mathematician might construct a group table for a group is to write down all the possible multiplication tables for a set of n elements, and check each one to see if it fits the description given. There are n^{n^2} possible multiplication tables, and noting that $4^{16} \simeq 4.3 \times 10^9$, we see that this method is impractical for groups of any decent size.

A much more plausible method would be to fill in the group table one entry at a time, and check that the incomplete multiplication table satisfies the description. This means that the description of the group has to be interpreted as ways of ruling out incomplete multiplication tables. This interpretation is not always straightforward but is usually possible. The advantage to this method is that once an incomplete multiplication table has been rejected by the description, any larger table built by adding more elements to this table will also be rejected, and so there is no point trying those possibilities. This has the potential to drastically cut down the search space, and we see that the problem of finding the group table given a non-explicit description of the group is a Constraint Satisfaction Problem (CSP).[6]

Now, in constrained searches, two major components of a search strategy are (i) the order of the variables to make assignments to (in our case, the order in which the entries in the group table are filled in), and (ii) the order in which the possible values are assigned to the variables (in our case the order of the set members to try in the table entries). For instance, a particular search strategy might assign entries in the top row of the table, left to right, then the second row left to right and so on. For every entry, it might try to assign the number 1, if this fails, then it might try the number 2, and so on. While these orderings are necessary to find a solution in practice, if we measure the size of the constrained search for a particular search strategy, this may add a bias in favour of one description, and the value will not be intrinsic to the description.

A fairer method is to find all the incomplete multiplication tables which the description rejects, and measure the increase over the number of multiplication tables which are rejected using the group axioms alone. This will measure the number of times that we can say for certain that a table with any number of entries can be rejected, and will give an estimate of how large the search will be using any search method. This can be formalised as follows:

Definition 3

- An **incomplete multiplication table of order** n [Shorthand: $imt(n)$] is a multiplication table for a closed binary operation on the first n integers, with 1 or 2 or ... or n^2 entries filled in.
- Given a set of constraints, X, then the **violation number for** X **on the set of** $imt(n)$**'s** is given by:

$$V_X(n) = |\{T : T \text{ is an } imt(n) \text{ and } T \text{ violates } X\}|.$$

- Note that the group axioms impose constraints on the imt's and $V_{Axioms}(n)$ denotes the violation number of the group axioms (and the order of the group) on $imt(n)$'s.
- Given a group of order n, G, and a classifying theory, C, then $desc_C(G)$ imposes constraints on the $imt(n)$'s. They must be used alongside the constraints given by the group axioms, as the classifying functions are defined on groups, and so the group axioms are assumed. Hence, when finding the violation number in this case (denoted $V_{desc_C(G)}(n, k)$), the group axiom constraints are also used.
- Further, the group table of G could be used to impose constraints on the $imt(n)$'s. In this case, there are only ${}^{n^2}C_k$ incomplete multiplication tables with k entries filled in which do not violate the constraints (ie. the possible k-tuples lifted from the group table). There are also ${}^{n^2}C_k n^k$ possible $imt(n)$'s with k entries filled in. Hence, writing $V_{Explicit}(n)$ for the violation number of the group table of G on the set of $imt(n)$'s, we find that

$$V_{Explicit}(n) = \sum_{k=1}^{n^2} \left({}^{n^2}C_k n^k - {}^{n^2}C_k \right) = \sum_{k=1}^{n^2} (n^k - 1)\, {}^{n^2}C_k.$$

- The **closeness of representation of** G **using** C**,** represented by $cr_C(G)$ is a measure of how many more imt's are rejected using the constraints given by $desc_C(G)$ than just using the group axiom constraints. It is normalised using the Explicit constraints, and given by:

$$cr_C(G) = \frac{V_{desc_C(G)}(n) - V_{Axioms}(n)}{V_{Explicit}(n) - V_{Axioms}(n)}.$$

[6] See [Tsang 93].

• Next, given a set of m groups, S, we can define the **representation content of C approximated using S** as:

$$S\_RC(C) = \frac{1}{m}\sum_{G \in S} cr_C(G),$$

• Given a classifying function, $f \in C$, The **representation improving value of f in C approximated using S,** represented by $S\_riv_C(f)$, is a measure of the representation content of the function if considered alone. It is given by:

$$S\_riv_C(f) = \begin{cases} \frac{S\_RC(<Order,f>)}{S\_RC(C)} & \text{if } S\_RC(C) > 0 \\ 0 & Otherwise \end{cases}$$

Example 3

We are going to be using the set of groups, S, and the classifying theory, C, from the previous examples, and we will be calculating $S\_RC(C)$, $S\_riv(Order)$, $S\_riv(IsAbelian)$ and $S\_riv(NSIE)$.

Firstly look at the group G_1 of order 3. We need to find $cr_C(G_1)$, and so we require $V_{desc_C(G_1)}(3)$, $V_{Axioms}(3)$ and $V_{Explicit}(3)$. Now, $desc_C(G_1) =< 3, true, 1 >$ and to find $V_{desc_C(G_1)}(3)$ we have to find the number of incomplete multiplication tables which either violate the group axioms or this description. To tell if an imt violates the $IsAbelian(G_1) = true$ part of the description, we just have to find two elements such that $a * b \neq b * a$. Also, an imt violates the $NSIE(G_1) = 1$ part of the description if it has two or three self inverse elements, or no self inverse elements. Of course, if an entry of the multiplication table is not filled in, we can make no assumptions about the value that entry will have, and this has to be taken into account. For details of how the axioms are violated by an imt, see [Colton 97].

A Prolog program was written which enumerated all the incomplete multiplication tables of order 3, testing each one first against the axiom constraints only, and then against the axiom and the description constraints. The following results were recorded:

$V_{Axioms}(3) = 254767$ *and* $V_{desc_C(G_1)}(3) = 256929.$

We can also calculate

$V_{Explicit}(3) = 261632,$

using the formula given in the definition. This allows us to calculate:

$$\begin{aligned} cr_C(G_1) &= \frac{V_{desc_C(G_1)}(3) - V_{Axioms}(3)}{V_{Explicit}(3) - V_{Axioms}(3)} \\ &= \frac{256929 - 254767}{261632 - 254767} = 0.315 \ (3 \ d.p). \end{aligned}$$

Next, look at the group G_2 of order 4. Here the set of incomplete multiplication tables of order 4 is too large to enumerate and test every imt, so estimates of the violation numbers were made using a randomly generated sample set of imt's. The estimates were calculated by first generating a sample set of imt's with only 1 entry, and recording the proportion, P_1 of imt's which violate the group axioms, and the proportion, Q_1 of imt's which violate the description of G_4. Then a sample set of imt's with only 2 entries was generated and the proportions P_2 and Q_2 were recorded, and this was repeated until P_k and Q_k were recorded for $k = 1, \ldots, 16$. The sample sizes were 50,000, which is more than is needed for the low values of k, but is a small sample for the higher values of k.[7]

The results were:

k	P_k	Q_k
1	0.0000	0.0000
2	0.0262	0.0633
3	0.2142	0.3132
4	0.4969	0.6128
5	0.7479	0.8335
6	0.8971	0.9428
7	0.9668	0.9843
8	0.9909	0.9963
9	0.9983	0.9994
10	0.9997	0.9999
11	1.0000	1.0000
12	1.0000	1.0000
13	1.0000	1.0000
14	1.0000	1.0000
15	1.0000	1.0000
16	1.0000	1.0000

These were then used to approximate $V_{Axioms}(4)$ and $V_{desc_C(G_2)}(4)$ by calculating

$$\hat{V}_{Axioms}(4) = \sum_{k=1}^{16} {}^{16}C_k(4^k - 1)P_k$$

and

$$\hat{V}_{desc_C(G_2)}(4) = \sum_{k=1}^{16} {}^{16}C_k(4^k - 1)Q_k,$$

which estimate the total number of imt's which violate the constraints. Then, using

$$\hat{cr}_C(G_2) = \frac{\hat{V}_{desc_C(G_2)}(4) - \hat{V}_{Axioms}(4)}{V_{Explicit}(4) - \hat{V}_{Axioms}(4)},$$

as an estimate for $cr_C(G_2)$, we calculated:

$$\hat{cr}_C(G_2) = 0.561 \ (3d.p).$$

Using a similar sampling and approximation process, we recorded:

$\hat{cr}_C(G_3) = 0.959$ *and* $\hat{cr}_C(G_4) = 0.786,$

and this gave us enough information to estimate $S\_RC(C)$:

$$\begin{aligned} \hat{S}\_RC(C) &= \tfrac{1}{4}(0.315 + 0.561 + 0.959 + 0.786) \\ &= 0.655. \end{aligned}$$

Finally, using the same processes, but using only the single functions $IsAbelian$ and $NSIE$ to describe the groups, it was possible to estimate that:

$\hat{S}\_riv(IsAbelian) = 0.383$ and $\hat{S}\_riv(NSIE) = 0.296.$

Further, because we have to include the order of the group in the constraints for V_{Axioms}, it is clear that

$S\_riv(Order) = 0.$

[7] For details of the standard error, see [Colton 97].

Interpreting these results, we see that as the order of the group increases, the constraining power of the axioms decreases, and so there is more possibility for the description constraints to reject imt's which the axioms accept. This accounts for the increase in the value of $cr_C(G)$ as the order of the group increases.

Representation Space of a Function

So far we have defined a classification as a way of describing groups, and have derived two measures to see how good those descriptions are. Firstly we measure how good the descriptions are at telling two groups apart, and secondly, we measure how easily we can recover an explicit representation (eg. the group table) from the descriptive representation. In both cases if we describe every group with its group table, then the acuity and representation content are both 1, ie. this is a perfect way of describing groups. However, writing a group table every time to represent the group is a very cumbersome way to go about things, and so it is preferable to have a more subtle description of the group. This can be viewed as one reason why a classification is sought in the first place.

Therefore, if two ways of describing groups are equally good at differentiating groups and the descriptions can be used to retrieve explicit representations equally well, then if the first description is more subtle, it will be preferred to the second. Measuring how subtle a description is amounts to measuring how difficult it is to write down. We can now use the fact that the output of the classifying functions was restricted to be a nested vector. As each description is just a collection of outputs, we can measure how many vectors there are and how many elements they contain, and this will give us an idea of how much space the description takes to write down. This is formalised as follows:

Definition 4

• Given a classifying function, f, and a group, G, then $f(G)$ will be a nested vector. If we then flatten this vector completely into a list of atoms, and call the flattened vector L, then the **storage space required to describe** G **with** f**,** *represented by* $stor_f(G)$ *is the size of* L*. ie.*

$$stor_f(G) = |L|.$$

• *Given a classifying theory,* C*, then the* **storage space required by** G **using** C**,** represented by $stor_C(G)$ is a measure of the space required to describe G using all the functions available in C, and is given by:

$$stor_C(G) = \sum_{f \in C} stor_f(G).$$

• Given a set of m groups, S, the **representation space required for** S **using** C**,** represented by $S\_repspace(C)$, is a measure of the average amount of space needed to write down the descriptions of the members of S, when using C. It is given by:

$$S\_repspace(C) = \frac{1}{m} \sum_{G \in S} \frac{1}{stor_C(G)}.$$

• Given a particular classifying function, $g \in C$, then the **representation space expansion value of** g **in** C **using** S**,** represented by $S\_rsev_C(g)$ is a measure of the proportion of the total representation space that g requires. It is given by:

$$S\_rsev(g) = S\_repspace(< g >).$$

Example 4

The outputs from the $Order$*,* $IsAbelian$ *and* $NSIE$ *functions are all single atoms, either the word true, or the word false, or an integer. Hence the nested vector outputs in these cases contain only one element and so we find that, with* S *and* C *as before;*

$stor_C(G_i) = \sum_{f \in C} stor_f(G_i) = 1 + 1 + 1 = 3$

for $i = 1, 2, 3, 4$.

So, $S\_repspace(C) = \frac{1}{4}\left(\frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3}\right) = \frac{1}{3}$

and it is easy to check that:

$S\_rsev(Order) = 1,\ S\_rsev(IsAbelian) = 1$

and $S\_rsev(NSIE) = 1,$

We can contrast these with the $Centre$ *function given by:*

$$Centre(G) = \{x \in G : \forall\ y \in G, x * y = y * x\}$$

as here, the output is a set, which can be written as a vector.

Calculations show that:

$Centre(G_1) =< 1, 2, 3 >, Centre(G_2) =< 1, 2, 3, 4 >,$

$Centre(G_3) =< 1, 2, 3, 4, 5, 6 >$ *and* $Centre(G_4) =< 1 >$

Hence

$stor_{Centre}(G_1) = 3,\ \ stor_{Centre}(G_2) = 4,$

$stor_{Centre}(G_3) = 6,$ *and* $stor_{Centre}(G_4) = 1.$

From these, using the set of 4 *groups,* S*, as before, we can calculate*

$S\_rsev(Center) = \frac{1}{4}\left(\frac{1}{3} + \frac{1}{4} + \frac{1}{6} + \frac{1}{1}\right) = \frac{7}{16}.$

Conclusions and Further Work

Given the assumption that the classification of groups is a major driving force behind the formation of finite group theory, the definitions of concepts found there have been interpreted as descriptions of groups. Following on from this, we have been trying to formalise how well the descriptions differentiate between two objects, how closely each description represents the object, and how succinctly the descriptions can be stated. These three properties of classifications would appear to be intrinsic but it is unlikely that values could be calculated for them in practice for a general classification.

By imposing certain restrictions, we have been able to formalise enough to calculate some values. Although the example calculations are fairly poor approximations, due to the small sample of groups used, it is possible to compare the classification given by three functions, $Order$, $IsAbelian$ and $NSIE$ against an explicit classification and the trivial classification.

Firstly, instead of the explicit classification which gives the group table for each group, we will use the function

$$Explicit(G) = \{< a, b, a * b >: a, b \in G\},$$

as this has output which can be written as a nested vector. Also, the trivial classification will be thought of as using the function $IsGroup(G)$, for which the output is always 'true' if a group is input. Then the values for acuity, representation content and representation space are approximately:

	Acuity	Repn Content	Repn Space
Explicit	1	1	0.02
C	1	0.66	0.33
Trivial	0	0	1

Remembering that these values have 0 as the worst case and 1 as the best case, we see that in order to decrease the space required to represent the groups, the classification loses some of the content, which will make it harder to retrieve the group operation. With the addition of more groups to the sample, we would find that the classification would also lose some acuity, and would describe two or more distinct groups in the same way. Note also that, although the trivial classification takes the minimal amount of space to describe each group, it gives no information about a group whatsoever.

As part of a larger project to perform automated concept formation in group theory, regarding each of the concepts as functions, it is more important to judge how much each function adds to the classification. This can be done by comparing a classification produced by only using the single function against a classification using all the functions. Again this has been turned into concrete calculations of the acuity improving value (aiv), the representation improving value (riv) and the representation space expansion value (rsev) of each function. The three functions in our example classification gave the following values:

	aiv	riv	rsev
Order	0.67	0	1
IsAbelian	0.33	0.38	1
NSIE	0.67	0.29	1

Now, as we can measure how good a particular concept (function) is, it is possible to write heuristic methods which produce new functions, with the heuristic designed to increase the average aiv, riv or rsev of the functions. One such heuristic is to start with a known concept which has a high aiv and syntactically mutate it using a template[8] into another concept. The hope is that the new concept will also have a high aiv. Hence if the acuity of the overall classification is lower than we require, we can use a known concept with a high aiv to produce another concept. [With the known functions in the examples looked at, we would probably chose to use the NSIE function to produce another concept]. Similarly whenever the representation content is too low we can use a concept with a high riv to produce another concept and if the representation space is too low, we can use a concept with a high rsev.

[8]For a description of the templates, see [Colton 97].

The writing and testing of the heuristic production rules as a computer program[9] is an ongoing project. Further work will also include relating these rather esoteric definitions to the more general work of information theory and constraint satisfaction problems. The link with information theory is possible because by representing a group with a description, we are writing it in a code. Then having the desired acuity of 1 is equivalent to having a uniquely decodable code, having a representation content of 1 is equivalent to having an instantaneous code, and the representation space of the description corresponds to the average length of the codewords in variable length codes. The method chosen to decode the descriptions is to solve a constraint satisfaction problem using the group axioms and the description of the group as constraints on a search over the space of incomplete multiplication tables.

Acknowledgements

This project has been funded by EPSRC research grant GR/L 11724.

References

[Colton 97] S Colton. Classification driven theory formation in mathemarics. Technical report, Department of Artificial Intelligence, University of Edinburgh, 1997.

[Davis & Lenat 82] R Davis and D Lenat. *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill Advanced Computer Science Series, 1982.

[Gorenstein 82] D Gorenstein. *Finite Simple Groups: An Introduction to Their Classification*. Plenum Press, New York, 1982.

[Humphreys 96] J Humphreys. *A Course in Group Theory*. OUP, 1996.

[Langley *et al* 87] P Langley, H A Simon, G L Bradshaw, and J M Zytkow. *Scientific Discovery - Computational Explorations of the Creative Processes*. MIT Press, 1987.

[Pierce 80] J Pierce. *An Introduction to Information Theory*. Dover Publications, 1980.

[Sims 90] M Sims. *IL: An Artificial Intelligence Approach to Theory Formation in Mathematics*. Unpublished PhD thesis, Department of Computer Science, Rutgers University, 1990.

[Tsang 93] E Tsang. *Foundations of Constraint Satisfaction*. Academic Press, London and San Diego, 1993.

[9]Named HR after Hardy and Ramanujan.