Mathematics for Formal Methods, a proposal for education reform

Mícheál Mac an Airchinnigh
University of Dublin,
Trinity College, Dublin, Ireland
e-mail:mmaa@cs.tcd.ie

Abstract

It is widely recognized that there is a body of mathematics which is crucial to the underpinning of computer science, software engineering, and information and communications technology. For the most part, at undergraduate level, such mathematics is generally considered to consist of discrete mathematics, including formal logic. At a research level, one expects to find abstract algebra, category theory, topos theory, etc.

In this paper we discuss a body of mathematics which is foundational to formal methods *per se* and used for the modelling of the usual sort of system artefacts: monoids and their morphisms. In particular we show how a single notion of distribution leads to better insights into many of the standard models in use. This discussion leads naturally to salient remarks on both the need for, and suitable direction to be taken in, a proposed education reform with a particular emphasis on mathematics for Information Technology.

keywords: distribution, education, formal method, modelling, monoid, morphism, structure, VDM, Z

1 Prologue

There can be no scientific or technological progress in a society unless research at University (and other third-level institutes of learning) filters down to the lowest level undergraduate programme and, in many instances, into schools at secondary and primary levels. (A paraphrase on a conversation with a colleague Dr. Paddy Nixon, 1998).

The idea that research must be masticated and redistributed in palatable form to the young is generally taken for granted in academia. The idea lies at the centre of the union between research and teaching even to the extent that there are some who say that research without teaching and teaching without research are barren endeavours. This intertwining of research (*id est, thought*) goes back at least as far as the Academy, to Plato, Socrates (his teacher) and Aristotle (his pupil) (Holland 1991, 9) and that a university is distinguished in its possession of certain essential characteristics: “teaching and research by intelligent, interested people, in appropriate and preferably agreeable surroundings, and with access to an adequate library” (Holland 1991, 10).

Such a filtering down of research *qua* thought takes time, sometimes a (very) long time:

“While this idea, that mathematics involves different categories and their relationships, has been implicit for centuries, it was not until 1945 that Eilenberg and MacLane gave *explicit* definitions of the basic notions in their ground-breaking paper ‘A general theory of natural equivalences’, synthesizing many decades of analysis of the workings of mathematics and the relationships of its parts.” (Lawvere and Schanuel 1997, 3)

In mathematics we have the very ‘recent’ case of the arrival of Category Theory into the first-year undergraduate curriculum with the publication of *Conceptual Mathematics* (Lawvere and Schanuel 1997), from which the above quote is taken.

The filtering often happens by accident. It *can* be planned for.

The subject matter to be passed on is not confined to the sciences and to technology. One notices the filtering in all subjects of learning from theology to poetry.

At the approach of a new century and a new millenium, however artificial such time divisions might be, it seems reasonable to take stock of developments in formal methods over the past twenty five years and to ask what is necessary for a reform of the usual computer science curriculum in order that formal methods might become fully integrated with other computer science courses and at the lowest possible level (in the undergraduate curriculum). We will observe that the field of formal methods requires a considerable body of mathematics and we hypothesize to what extent some of that may be repackaged for use at secondary and primary levels.

We will observe that such packaging has implications for the *status quo* and, in particular, for that well-entrenched body of mathematics which, due largely to historical reasons, derives from the rise of the Age of Reason (dating from the times of da Vinci, Copernicus, Bacon, Descartes *et cetera*) and which was upheld largely by scientific progress and/or progress of science.

The launching of a new national curriculum at any level in schools is always a rare event and one which usually signals or bolsters huge cultural shifts. The lower the level, the more profound and widespread the effect (e.g., the so-called ‘new math’ which ‘arrived’ in Ireland in the late 60’s–early 70’s).

The launching of a new degree programme is an event of a similar nature, even where the degree is nominally different from similar degrees taught by the same Department. Such is the case of the new B.A. (Mod.) in Information and Communications Technology (<http://www.cs.tcd.ie/courses/baict/>), launched in October 1997, in the Department of Computer Science at the University of Dublin, Trinity College. The Department teaches, among other things, the B.A. (Mod.) in Computer Science.

Comparisons are inevitable and questions of the form, “What is the difference between . . .?”, arise quite naturally. From the students’ perspective, those in Computer Science already refer to the new degree as the “B.A. *Lite*.” (The American spelling which I have chosen is deliberate.) Such a remark may readily be dismissed as flippant or humourous. On the other hand it does recall to mind Bourdieu’s analysis which lead to the 1968 crisis in university education in France.

“... the social and academic effects of the increase in numbers are all the more striking in an academic institution (institute, faculty or discipline), the more its position in the hierarchy—and, secondly, the teaching of the content offered—predispose it to serve as a refuge for students, who in the previous state of the system, would have been excluded or would have dropped out.” (Bourdieu 1988, 164)

The new degree in Information and Communications Technology certainly could be argued to have satisfied the conditions observed by Bourdieu. Nevertheless, from the University’s perspective, the new degree *is* a Moderatorship degree just like any other Moderatorship degree. It fell to the author then to ensure that at least the Mathematics component of the first year of the new degree was “up to standard”, albeit different in nature to all other first year mathematics courses for computing students.

In this paper, we will make some reference to this mathematics programme for the new degree in Information and Communications Technology. However, since the author also teaches a fourth year undergraduate course in the mathematics of formal methods (using the Irish VDM as paradigmatic method) for the Computer Science degree, then much of the focus will be at that level and there are salient points which will be made on the likely impact of a proposed mathematics programme for education reform in the computer technology domains.

We give now an outline of the argument. The following section entitled *Which Mathematics should be taught?* reports on the *status quo* in Irish secondary level schools and difficulties encountered by all students entering university and for which mathematics is a requirement. Then in section 3 *Mathematics of Formal Methods* we present some of the standard mathematics currently required of our graduating class in Computer Science. This section is especially noteworthy in that it exhibits planned filtering of research

material over a period of ten years. Section 4, *Foundation Mathematics for Information Technology*, proposes a mathematics curriculum that is distinctively different from that which pertains on the computer science degree course and which will facilitate the filtering of the mathematics of formal methods into the first year. Category theory forms a part of such a course. Finally, we summarize our position in the Epilogue.

2 Which Mathematics should be taught?

“One of the disappointments experienced by most mathematics students is that they never get a course in mathematics. They get courses in calculus, algebra, topology, and so on, but the division of labour in teaching seems to prevent these different topics from being combined into a whole.” (Stillwell 1989, vii).

In Ireland, the key examination in mathematics for entry to third level education is the Leaving Certificate Higher Level Mathematics course. There has been a dramatic change in the syllabus of the current course which was first examined in 1994. Essentially, in the opinion of this author, the course content has been “watered down” and that such watering down was done to facilitate even greater “mass higher education”, to wit to build a broader pool of suitable candidates for entry into third level education, to paraphrase another colleague, Dr. Séamus McGuinness, from the School of Education in the University of Dublin. The details on the “social necessity” for such mass education systems are readily available (Coolahan 1994, 91 et seq.).

Such watering down leads to the concept of “devaluation of diplomas”, which in the context of the mathematics course mentioned means that a C grade today is not of the same standing as the C grade of five years ago and hence that those who obtained their diplomas five years ago may find themselves with devalued diplomas.

Here the term “devalued diploma” is used in the same technical sense as that employed by Bourdieu (1988) in his *Homo Academicus* and which is in danger of being applied to B.A. *Lite* degrees. The devaluation of diplomas in France (May 1968) is identified as one of the facets of the crisis in the university system that erupted in 1968:

“We cannot account for the crisis [...] without mentioning the principal effects of the increase in the number of pupils, that is, a devaluation of academic diplomas which causes a generalized downclassing, particularly intolerable for the more privileged, and, secondarily, the transformations in the functioning of the education system which result from the morphological and social transformation of the public” (Bourdieu 1988, 162).

In the case of science and engineering graduates at the University of Dublin, for which mathematics is an essential subject, we must of course maintain the standard of the degree and their diplomas. We do not propose here to analyze the effects of the devaluation of the Leaving Certificate for these degree courses, and in the case of the degrees of the Department of Mathematics suffice it to say that they now require at least a grade B. The degrees of the Department of Computer Science still require at least a grade C.

What then is the state of the mathematics of the incoming students who have attained at least a grade C?

We hypothesize that with the reduction in the second-level course content (1993–) it would appear that in general the released class time has been utilized for the purpose of doing more drill, more exercises. It seems to be the case that *the majority of students who just achieve the grade C* do not truly comprehend the mathematics which they have been taught. They have learned “to pattern-match”.

In addition to the core of the mathematics which must be learned, there are four special options on offer, (i) further calculus and series, (ii) further probability and statistics, (iii) groups, and (iv) further geometry, of which one is to be chosen **for examination**. Naturally, it is to be expected that in order to maximise one’s grade only one further option will be taken/offered/taught! Currently it is the first option that proves to be the most popular.

We hypothesize that there is undue emphasis which is given to the calculus, an emphasis that is due to two important factors. First, there is the effect of tradition, also known as **the principle of academic inertia**

which declares that teachers are wont to teach what they have been taught, to pass on the tradition as it were (and this is indeed one of the major obligations of a university). Second, the calculus is a foundation for the sciences and engineering and these are in turn the declared foundation for a modern economy.

Where option (ii) is chosen the argument can also be made on grounds of relevancy/usefulness to the modern economy. Surely good government and administration rest on the foundations of economics which in turn rests on probability and statistics (and it goes without saying, on the calculus)?

It may seem hard to offer relevancy arguments for either *groups* (option (iii)) or *further geometry* (option (iv)) and not surprisingly these are not well-favoured. As it turns out, both topics are extremely relevant to computer science. Permutation groups (S_n) lie at the heart of all sorting and it has been said Knuth (1973, v) that “virtually *every* important aspect of programming arises somewhere in the context of sorting or searching!” The free group over an alphabet Σ , denoted by $FG(\Sigma)$, lies at the heart of computability. Multimedia, computer-aided design and robotics attest to the centrality of geometry.

Just as in the sciences and engineering, the degrees of the Department of Computer Science also require the calculus for certain aspects, such as robotics, computer graphics, computer vision, etc. Consequently, the first year mathematics course of the B.A. (Mod.) in Computer Science is very similar to the others. That of the B.A. (Mod.) in Information and Communications Technology is very different, although it also ‘covers calculus’.

From the text it will appear that we have made a distinction between the ‘sciences and engineering’, and ‘computer science’, and indeed we have. Although there are many aspects to computer science which come within the ambit of the discipline of engineering (and in the University of Dublin the Department of Computer Science is situated in the Faculty of Engineering and Systems Sciences), there are aspects which come within the ambit of the discipline of science. Moreover, there are aspects which are as foundational as ‘classical mathematics’, to wit, discrete mathematics incorporating mathematical logic and computability. But there is much more to the discipline.

Computer Science also embraces *language*. It is this which uniquely sets it apart. We usually sum up this peculiar aspect of computer science by declaring that language and machine are the same up to isomorphism, *id est*, what is computable by language is computable by machine and *vice versa*. There can not be (ontologically speaking) any difference.

The launching of a new Moderatorship degree in Information and Communications Technology permits us to explore/exploit this uniqueness not least of all within the context of the content and *modus operandi* of the new mathematics course offered to the incoming students.

We shall give some of the details of this course content in section 4 of the paper. To appreciate the philosophy of the course, and to understand the rationale behind the contents and the structure of the course, the first year of which has now been completed, taught and examined (October 1997–June 1998), it is necessary to have some appreciation of the mathematical maturity expected of the current fourth year undergraduates in the B.A. (Mod.) in Computer Science.

The author of the paper, being the author also of both the new first year mathematics course and of the mathematics course for formal methods in the fourth year computer science degree course, is in a unique position to explain what is necessary, in his opinion, for education reform in mathematics to account for the emerging primacy of computing both as a scholarly discipline and as a key profession in the next century. To bolster this position, we now present a detailed account of one aspect of the aforementioned fourth year course.

3 The Mathematics of Formal Methods

Principle 3.1 [*Consecutive Phases*] *For efficient learning, an exploratory phase should precede the phase of verbalization and concept formation and, eventually, the material learned should be merged in, and contribute to, the integral mental attitude of the learner (Pólya 1981, II 104)*

George Pólya was one of the great mathematicians, and both teacher and pedagogue of mathematics, in

this century. Of all his works, that entitled “*How to Solve It*” (Pólya 1957), is surely well-known not only to lecturers, teachers, and students of mathematics but to those of engineering and computing as well? Not so well-known perhaps is the more extensive two volume work “*Mathematical Discovery, On understanding, learning, and teaching problem solving*” (Pólya 1981) from which the quoted principle of learning (one of three) is taken. Pólya’s works are relevant to the (teaching of) mathematics of computing and *a fortiori* of formal methods.

After ten years spent teaching the VDM to fourth year honors degree students in Computer Science at the University of Dublin, Trinity College, it seemed appropriate to set down on paper some of the reflections that have occurred from time to time throughout that decade. That it should have been VDM and not some other (model-theoretic) formal method such as Z is entirely accidental, in the sense of being an accident of history.

Without wishing to commit to a specific pedagogical theory nor to propose a specific educational philosophy it is nevertheless useful to place the reflections within some definitive context and thus impose some order.

There are two distinct periods which span the decade (1987–97). The earlier period, from 1987 until 1991, may be characterized by the emergence of what became known as the Irish School of the VDM (VDM[♣]). It began with the first VDM Symposium (1987) and culminated with the VDM Symposium in Noordwijkerhout (1991). The culmination of that development is evident in (Mac an Airchinnigh 1990) and (Mac an Airchinnigh 1991). The second period (1991–) is the primary subject matter of this section of the paper.

In the very beginning, that is to say during the first period, the primary pedagogical concern was the acquisition and/or construction of elementary models in order that the students might gain some familiarity with the subject matter, so to speak, of the formal method. The gathering of the models may be likened to the exploratory phase proposed by Pólya in his work on *Mathematical Discovery* (Pólya 1981). Such standard models were to be used as exemplars for other problems to be solved.

Figure 1: Entering a new word into a dictionary.

For example, the model of a class of sets given in terms of the powerset of a set of words W and denoted by $\mathcal{P}W$, was introduced by the notion of the ‘spelling-checker dictionary’ such as might be used in the trademarked game of SCRABBLE. From this we may discuss ‘implementation of set as sequence’ and introduce the model of a class of sequences, W^* . Instead of such **reification**, the idea arose of exploring a model by introducing extra information at the same abstract level as the original model. Such a development step was termed an **elaboration**. In this manner, one might associate sets of definitions with each word in the dictionary, $W \rightarrow \mathcal{P}D$ or the dictionary might be distributed across a collection of locations, $L \rightarrow \mathcal{P}W$.

Having gathered together a collection of standard models and development steps, it was natural to examine the inter-relationships between the different models. It was at this point that some of the power of the emerging notation, being developed and used, was keenly felt. Specifically, by having recourse to

Figure 2: Retrieval from a distribution.

currying for the specification of operations on the models, one was able to exploit the conciseness of the commuting diagram in order to exhibit those inter-relationships. For instance, the proof of the correctness of the operation to enter a new word in a dictionary with respect to set and sequence models took the standard form (Fig. 1). In addition to the commuting diagram frequent use was made of arrows such as in the reconstruction (or retrieval $\mathfrak{R}$) of a dictionary from its distributed form (Fig. 2). The **use** of category theory language and notation may be likened to the second *formalizing* phase mentioned by Polya. At no time was it ever intended that the students should be well-versed in category theory. On the contrary, it was clearly stated that the appropriate level of abstraction was deemed to be that of the monoid and morphism (Mac an Airchinnigh 1990). This is comparable to the general attitude taken in mathematics. The language of category theory is freely used without having to delve into the theory itself. Further, the notation of the School was intended to free one from the burden of always having to “think about what one is doing” (Whitehead 1978, 41).

The third phase of *assimilation* required that there be a grounding of the work done in formal methods with (i) mathematics on the one hand and (ii) computing (and modelling) on the other. In particular, it was clear that the VDM had to cover, in some sense, all of the computing that the students had already absorbed over a three year period in their honors degree course. This could only be achieved by reference to key aspects: algorithms, data structures, hardware components, etc. Similarly, since the students did not take as many mathematics courses as their counterparts in engineering, it was necessary that they be made aware of the richness and relevancy of modern algebra at the very least.

In summary then this first stage of the education programme in formal methods (1987–1991), which was exhibited at length in Noordwijkerhout (Mac an Airchinnigh 1991), consisted of

1. the gathering together of a class of standard abstract models;
2. the introduction of a class of standard development steps, whether reifications or elaborations;
3. the presentation of a variety of fully-developed proofs in the algebraic style.

The second stage (1991–) to be outlined in the remainder of this section of the paper consisted in striving for a greater richness in the mathematics. It seemed clear that the existing models and specifications were, in a sense, far too detailed. There was a need for even greater abstraction. How might that be obtained? As it turned out there has been some modest progress and this may be due in part to

1. the **imperative** that the link between research and teaching be maintained and fostered. This entails the re-presentation of research material in a form that is accessible to undergraduate students. [Similar linkage is used to infuse industry via training.]
2. the **shift** away from a pure constructive approach that allowed the embracing of the totality of mathematics available. [Algebraic topology became a focus of serious activity.]
3. the **belief** that there must be a geometry underlying formal methods. The algebra was there. Where was the Cartesian dual? [A breakthrough was eventually obtained, first in the discovery that tail-recursive forms of Σ^* -morphisms (i.e., free monoid morphisms) were generalized affine translations

and then the application of (albeit trivial) fibre bundles and sheaves to the specification of system components.]

This second stage was characterized by a major conceptual shift. Conscious effort was made to speak the language of the mathematicians, a task that proved difficult. The guiding principle that we have adopted may be pithily phrased thus:

Formal methods will become mainstream software engineering whenever its mathematics becomes mainstream engineering.

To exhibit something of the mathematics of formal methods the main part of this section deals with the single concept of a **distribution**. It is deliberately intended to suggest some relation to ‘distributed system’.

3.1 Distributions

“If we look at older areas of Engineering, we will see that mathematical methods are not introduced by teaching languages” (Parnas 1995a, 3).

With the presentation and publication of the *Tutorial Lecture Notes on the Irish School of the VDM* (Mac an Airchinnigh 1991) a considerable body of material was readily available to the students. The next step was clearly to extend the range of applications and, equally significantly, to deepen the theory.

Reliance upon a formal language with the implied duality of formal syntax/semantics had been completely abandoned in favour of flexibility of notation. If ‘semantics’ were needed then one was directed to algebra.

With experience in applications, it became clear that whether one took the development step of reification or elaboration one often encountered a structure that suggested **distribution**. We now present details of two of those structures: the indexed monoid, and the fibre bundle.

3.1.1 Indexed Monoids

“Looking back at the put command of the file system, I hypothesise that with a little more thought and analysis, it can probably be turned into an operator, a problem I will leave for future work” (Mac an Airchinnigh 1990, 228).

The construction of new monoids from old. The ‘put’ operation of the file system, $\varphi \in (Id \rightarrow FILE)$, quoted above, typified the duality of the specification of operations: either one extended the system with a file f identified by i , $\varphi \sqcup [i \mapsto f]$, or one overrode/overwrote the system, $\varphi \uparrow [i \mapsto f]$, and the override operator covered the extend operator. If the override operator were used for both then a pre-condition was necessary to distinguish the cases/meanings.

The indexed monoid, which first appeared in print, and with the status of a theorem, in the tutorial lecture notes on *Formal Methods & Testing* (Mac an Airchinnigh 1993, 29), and which was formulated in late 1992 or early 1993, arose as a direct consequence of the generalization of the monoids of bags (Mac an Airchinnigh 1990, 208–24) and relations (Mac an Airchinnigh 1990, 224–9).

Formally, it was defined in the context of monoids which were not groups:

Definition 3.1 [*Indexed Monoid I*] Let $(M, *, e)$ denote an arbitrary monoid, which we shall call the base monoid, with identity e , and $(M', *)$ the corresponding semigroup, i.e., with $M' = \triangleleft[e]M$. Then for a space X , the structure $(X \rightarrow M', \otimes, \theta)$ is an indexed monoid which inherits its operator properties from $(M, *, e)$, where for μ in $X \rightarrow M'$,

$$\mu \otimes [x \mapsto m] = \begin{cases} \mu \sqcup [x \mapsto m], & \text{if } \neg \chi_\mu(x) \\ \mu \uparrow [x \mapsto \mu(x) * m], & \text{otherwise} \end{cases}$$

For convenience, we let $X \circ (M, *, e)$ denote the construction of the indexed monoid from the base monoid. With this notation, we may express the additive monoid of bags ($X \rightarrow \mathbb{N}, \oplus, \theta$) as $X \circ (\mathbb{N}_0, +, 0)$ and the monoid of binary relations (expressed functionally) ($X \rightarrow \mathcal{P}'Y, \odot, \theta$) as $X \circ (\mathcal{P}Y, \cup, \emptyset)$.

In teaching the indexed monoid, it became clear that we needed to find new forms of expression. The name ‘indexed monoid’ has stayed, although one can not help but think that the name is totally inappropriate. First, one proposed that there were the natural numbers. Then one suggested that these might be indexed with respect to the names of some domain, thereby giving rise to a bag or set of counters. For example, from 5 one might propose to construct $[s \mapsto 5]$ to denote 5 pieces of silver (in the purse). It seemed strange then as it does now. If one starts with the bag, however, then it is easy to see that the natural numbers arise as an abstraction from the purse of money. This naturalness confirmed that the indexed monoid might be a very important new structure for use in modelling and in specifications.

The first comprehensive application of the indexed monoid to be published was based on a model of some aspects of the relationships between political parties, elected representatives and the Irish Parliament (Hughes and Donnelly 1995).

Although the definition of the indexed monoid given above was *constructive*, it proved not to be amenable for the proof of basic properties, such as for example, the associativity law of the monoid. Indeed, it was not until later that a suitable mathematical framework, due to Arthur Hughes, was constructed (Donnelly, Gallagher, and Hughes 1996, 15). First we introduce the concept of a direct power monoid.

Definition 3.2 [Direct Power Monoid] Let $(M, *, e)$ denote an arbitrary monoid, which we shall call the base monoid, with identity e . Then for an indexing space X , we construct the space of total functions from X to M , denoted by M^X . Let e^X denote the constant function $x \mapsto e$ for all $x \in X$. Then the structure $(M^X, *, e^X)$ is a direct power monoid where for $f, g \in M^X$,

$$(f * g)(x) := f(x) * g(x)$$

Now we are in a position to give a second definition of indexed monoid in terms of the direct power monoid (Donnelly, Gallagher, and Hughes 1996, 20).

Definition 3.3 [Indexed Monoid II] Let $(M, *, e)$ denote an arbitrary monoid, which we shall call the base monoid, with identity e , and $(M', *)$ the corresponding semigroup, i.e., with $M' = \triangleleft[e]M$. Then for a space X , the structure $(X \rightarrow M', \otimes', \theta)$ is an indexed monoid which inherits its operator properties from $(M, *, e)$, where for μ, ν in $X \rightarrow M'$,

$$\mu \otimes' \nu := \left((e^X \dagger \mu) * (e^X \dagger \nu) \right)'$$

The use of primes is necessary to eliminate the occurrence of nulls which might arise as a result of the composition on the right. However, it is important to note that the notion of indexed monoid may be extended to that of an indexed group and other indexed structures, for details of which the reader is referred to the Doctoral thesis of O'Reagan (1997). The original definition was intended to narrow the scope to monoids which were not groups.

However, it is important to note that the original definition is still the one which is first presented to the students in that it is constructive and is immediately accessible to their intuition. Perhaps there will come a day when one will consider the second definition as the ‘proper’ way to introduce the indexed monoid?

Now that we have a sound mathematical basis and some experience in applying it, let us take a look at how it may be presented in the course of a lesson.

The lesson on/of the distributed dictionary. The following set piece, which constitutes a single one hour lecture, provides a useful introduction to the indexed monoid and its morphisms.

We are given a spelling-checker dictionary, an element in the space $\mathcal{P}W$. If we distribute this dictionary over several locations, an element in the space $L \rightarrow \mathcal{P}W$, then we may retrieve the spelling-checker dictionary

$$\begin{array}{ccc}
 & A[w]\delta := \delta \cup \{w\} & \\
 & \xrightarrow{\quad A[w] \quad} & \\
 \mathfrak{R} = \cup / \circ \text{rng} \uparrow & \mathcal{PW} & \mathcal{PW} \\
 & \xrightarrow{\quad A[l, w] \quad} & \\
 L \rightarrow \mathcal{PW} & & L \rightarrow \mathcal{PW} \\
 & \xrightarrow{\quad A[l, w]\delta := \delta \upharpoonright [l \mapsto \delta(l) \cup \{w\}] \quad} &
 \end{array}$$

Figure 3: An invitation to a proof.

in the usual way. Now let us propose to add a new word to the dictionary. The resulting transformations are exhibited as usual in a commuting diagram (Fig. 3). Correctness of the addition of a new word with respect to the distributed system, must now be established. This is a matter of demonstrating that the left side is equal to the right side:

$$A[w] \circ \mathfrak{R} = \mathfrak{R} \circ A[l, w]$$

For the left side we obtain, trivially,

$$\begin{aligned}
 (A[w] \circ \mathfrak{R})(\delta) &= A[w] \circ \cup / \circ \text{rng}(\delta) \\
 &= \cup / \circ \text{rng}(\delta) \cup \{w\}
 \end{aligned}$$

The interesting part is the right side:

$$\begin{aligned}
 (\mathfrak{R} \circ A[l, w])(\delta) &= \mathfrak{R}(\delta \upharpoonright [l \mapsto \delta(l) \cup \{w\}]) \\
 &= \cup / \circ \text{rng}(\delta \upharpoonright [l \mapsto \delta(l) \cup \{w\}])
 \end{aligned}$$

The student is *supposed to know* that the override operator provides a space of maps with the structure of a monoid. Further, the student is *supposed to know* that often the retrieve operation is a morphism of monoids. [**Aside:** *If the student does not know these things, then one is provided with an opportunity to introduce monoids and their morphisms within the context of this practical example.*] Consequently, it is expected that the student should **assume** that application of $\mathfrak{R} = \cup / \circ \text{rng}$ to expressions of the form $\delta_j \upharpoonright \delta_k$ should give $\cup / \circ \text{rng}(\delta_j) \cup \cup / \circ \text{rng}(\delta_k)$. The problem is that, in general, $\cup / \circ \text{rng}$ is not a morphism of the monoid $(L \rightarrow \mathcal{PW}, \upharpoonright, \theta)$. Specifically, $\delta_j = [l \mapsto \{w\}]$ and $\delta_k = [l \mapsto \{\bar{w}\}]$ may be used as a counter-example to show that

$$\cup / \circ \text{rng}(\delta_j \upharpoonright \delta_k) \neq \cup / \circ \text{rng}(\delta_j) \cup \cup / \circ \text{rng}(\delta_k)$$

Nevertheless, for this proof, it **is** the case that we may treat $\cup / \circ \text{rng}$ as if it **were** a morphism:

$$\begin{aligned}
 \mathfrak{R} \circ A[l, w](\delta) &= \dots \\
 &= \cup / \circ \text{rng}(\delta \upharpoonright [l \mapsto \delta(l) \cup \{w\}]) \\
 &= \cup / \circ \text{rng}(\delta) \cup \cup / \circ \text{rng}([l \mapsto \delta(l) \cup \{w\}]) \\
 &= \cup / \circ \text{rng}(\delta) \cup \cup / \circ \{ \delta(l) \cup \{w\} \} \\
 &= \cup / \circ \text{rng}(\delta) \cup \delta(l) \cup \{w\} \\
 &= \cup / \circ \text{rng}(\delta) \cup \{w\}
 \end{aligned}$$

and we are done.

The fact that $\cup / \circ \text{rng}$ behaves like a monoid morphism suggests that it is indeed one! It turns out that we have picked the *wrong* monoid. The monoid one ought to have picked is the classical indexed monoid of binary relations:

$$L \circ (\mathcal{P}W, \cup, \emptyset) = (L \rightarrow \mathcal{P}'W, \odot, \theta)$$

Hence, from this perspective of monoid structure, a more appropriate specification of the add operation would be

$$A[l, w]\delta := \delta \odot [l \mapsto \{w\}]$$

This single expression represents two distinct cases. In order to select that which represents the addition of a new word to an existing location we **must** supply a pre-condition:

$$\text{pre-}A[l, w]\delta := \chi_\delta(l) \wedge \neg \chi_{\delta(l)}(w)$$

where χ is the usual characteristic function for sets, overloaded appropriately.

The other case,

$$\delta \odot [l \mapsto \{w\}] = \delta \sqcup [l \mapsto \{w\}],$$

has the interpretation that the distributed dictionary is extended with a *new* location and a *new* word!

This choice of indexed monoid and morphism *draws the focus away from* an expression of the form $\delta \dagger [l \mapsto \{w\}]$ which is to be interpreted as the replacement of the dictionary at location l with $\{w\}$. Since it is not true, in general, that $\cup / \circ \text{rng}$ is a morphism of the usual monoid of maps, then we are led to infer that the form $\delta \dagger [l \mapsto \{w\}]$ can not correspond to any operation at the original abstract level of dictionary! This is a further justification of our intuition about the ‘rightness’ of the choice of the indexed monoid to model the distribution.

Since we have as great an interest in developing the underlying mathematics as much as in applying it, we note that there are key relationships between the different spaces involved in the modelling of the distribution. Specifically, we have looked at three monoids and one morphism which provided a retrieval function $\mathfrak{R}$. In order to indicate that there are opportunities for further development work, we may introduce a diagram of the form shown in Fig. 4, where f, g, h are functions yet to be determined (if they exist).

$$\begin{array}{ccc}
 & (\mathcal{P}W, \cup, \emptyset) & \\
 h \nearrow & & \nwarrow \mathfrak{R} = \cup / \circ \text{rng} \\
 (L \rightarrow \mathcal{P}W, \dagger, \theta) & \xrightleftharpoons[g]{f} & (L \rightarrow \mathcal{P}'W, \odot, \theta)
 \end{array}$$

Figure 4: A problem to be solved.

Finally, from a pedagogical perspective, and especially for the benefit of the lecturer, one may summarize some of the key elaborations of the models with respect to a particular operation, such as $A[\dots]$, in a single commuting diagram (Fig. 5). In this diagram we have left open the retrieval, $\mathfrak{R}$, from a distribution $L \rightarrow (W \rightarrow \mathcal{P}D)$ to the original $W \rightarrow \mathcal{P}D$. This serves two purposes: (i) it reminds one that the distribution is not uniquely determined and hence that there is not a unique retrieval function, and (ii) it is a placeholder for an exercise, a tutorial, or an examination question.

Among all distributions, there is a unique class, the **partitions**. In the next subsection we deal with this class, but under a different name: **fibre bundles**. Again, the goal is to inform and to open a dialogue.

Figure 5: Opportunities.

3.1.2 Fibre Bundles

“We can then re-capture A_i as the inverse image under p of $\{i\}$, for

$$p^{-1}(\{i\}) = \{x : p(x) = i\} = A_i$$

The set A_i is called the *stalk*, or *fibre* over i . The members of A_i are called the *germs* at i . The whole structure is called a *bundle* of sets over the *base space* I . The set A is called the *stalk space* (l’espace étalé) of the bundle. The reason for the botanical terminology is evident—what we have is a bundle of stalks, each with its own head of germs (think of a bunch of asparagus spears)” (Goldblatt 1984, 90)

Given any model $\mu \in \mathcal{M} = X \rightarrow Y$, we construct its inverse image $\mu^{-1} \in \mathcal{M}^{-1} = Y \rightarrow \mathcal{P}'X$. We are careful to note that the null set can not be an element of the range of an inverse image. This is a critical observation.

That the inverse image of a map was important in modelling was apparent very early on. For example, suppose one is given a temperature chart μ which associates a temperature t with a city c . Then the answer to the question, “which city has temperature t ?”, is immediately answered by the expression $\mu^{-1}(t)$ (Mac an Airchinnigh 1991). In the special case that the temperature was not recorded in the chart, the answer to be returned was given as $\emptyset$.

Such a conventional expression which encodes the conventional response, regularly used in mathematics, raised an apparently serious problem in the constructive mathematics being developed. Specifically, since the evaluation of a map μ at a point x was defined only if x was in the domain of μ , then to evaluate $\mu^{-1}(y)$ where y was **not** in the domain of μ^{-1} was a glaring inconsistency.

Principle 3.2 *Problems are an opportunity for learning and development.*

The resolution of the problem depended on the notion of a totalizer (Mac an Airchinnigh 1997), a notion which seems related to that of cone construction. Specifically, μ^{-1} is completed (i.e., totalized) by constructing the cone $\emptyset^Y$, a constant total function which is constructive, and applying surgery (i.e., cut and paste):

$$\emptyset^Y \dagger \mu^{-1}$$

Since μ^{-1} is a bundle of fibres with base $\text{rng } \mu$, then $\emptyset^Y \dagger \mu^{-1}$ is a cone completion of the fibre bundle space.

From the problem on the distribution of a dictionary in the previous subsection, there was left open the problem of considering the meaning of the $A[l, w, d]$ operation, an operation which typifies a particular class of operations on a distributed system. Those familiar with this problem, will realize that the specification

$$A[l, w, d]\delta := \delta \upharpoonright [l \mapsto \delta(l) \oplus [w \mapsto \{d\}]]$$

may not be the one we want if the distribution permitted the sharing of information at multiple locations. There would be no problem with this specification if the word w only occurs at the location l and nowhere else. Clearly, then the condition that must be satisfied is that

$$\left((\mathcal{I} \rightarrow \text{dom})\delta \right)^{-1}(w) = \{l\}$$

where $\mathcal{I} \rightarrow \text{dom}$ is the morphism that takes us back into $L \rightarrow \mathcal{P}W$ and the inverse is a relational inversion (see Mac an Airchinnigh (1993)). If such a condition is true for all words then the distributed system is a fibre bundle (think asparagus) and specifications and proofs are straight forward.

As a final indication of the pervasiveness of this notion of fibre bundle, let me briefly introduce a very abstract model of a hashing function and hash table with overflow chaining, a model due to Arthur Hughes. The set of all integers modulo a prime p is denoted by $\mathbb{Z}_p$. Let W denote the space of words. Then the total function $h \in \mathbb{Z}_p^W$ denotes a hash function. For each j in $\mathbb{Z}_p$ we have the set of all words $h^{-1}(j)$ which hash to j . Clearly, this is a fibre and the collection of all such fibres a fibre bundle with base $\mathbb{Z}_p$.

At any given moment we may consider a subset S of W to have been hashed into a table. Such a set is an element of the powerset domain $\mathcal{P}W$. Consequently, the pair

$$(h, S) \in (\mathbb{Z}_p^W \times \mathcal{P}W), \quad S \subseteq \text{dom } h$$

denotes a particular hash table model. Full details of this model are available in a technical report (Hughes 1997). Subsequently, a further analysis revealed that this naïve view of a hash table as a fibre bundle of this sort was not the most appropriate one. In fact, we could turn the table ‘inside out’ so to speak. Instead of viewing $\mathbb{Z}_p$ as the base it turned out that it should be viewed as the standard fibre. The more appropriate choice of base is the space of words $\mathcal{P}W$. Space does not permit an elaboration of this particular insight here. A full paper on the subject is available (Mac an Airchinnigh and Hughes 1997).

There is far much more to the notion of fibre bundle than the bare treatment that has been presented here. What we intend to demonstrate is that (i) the notion of fibre bundle is simple (think asparagus), and (ii) powerful. Most importantly it permits entry into a field of mathematics which will provide further enrichment of the base for formal methods.

Now perhaps we are in a position to appreciate something of the ultimate goal of the degree programme towards which we would like the *first year students* to strive. Clearly, they need a good grounding in abstract algebra and must be prepared to accept non-commutativity as the norm in computing (viz. the free monoid, monoid of maps under override, etc.). In place of the use of the (total) function, which is ubiquitous in the sciences, the students of computing will embrace the partial function (usually referred to as a map) as the fundamental modelling tool and in place of the usual composition of functions they will come to understand that there are other natural and constructive operations on partial functions such as extension, glueing and overwrite/override. It is clear from the use of commuting diagrams (and the functor constructors) that some familiarity with the notations and concepts of category theory would be helpful.

4 Foundation Mathematics for Information Technology

“The theory of numbers is generally considered to be the ‘purest’ branch of pure mathematics. It certainly has very few direct applications to other sciences, but it has one feature in common with them, namely the inspiration which it derives from *experiment*, which takes the form of testing possible general theorems by numerical examples. Such experiment, though necessary in some form to progress

in every part of mathematics, has played a greater part in the development of the theory of numbers than elsewhere; for in other branches of mathematics the evidence found in this way is too often fragmentary and misleading” (Davenport 1992, 7)

We have remarked earlier that incoming students (technically known as Junior Freshmen, whether male or female) no longer seem to be as well-prepared in mathematics as in former times (such as prior to 1994) and that this was in part due to the phenomenon of ‘mass-education’ and a certain weakening or dilution of the content of the second level. In this we are not alone.

“It was clear to us at the University of Manchester that we should completely rethink and broaden our curriculum, including material which we had previously expected students to know on entry to the course but also including introductory material on combinatorics, computer skills and numerical mathematics, as well as encouraging the development of problem solving skills” (Eccles 1997, ix).

An outline of the foundation course in mathematics for the new degree in Information and Communications Technology (ICT) was drawn up by colleague Dr. Andrew Butterfield who was a member of the ‘degree design’ committee. There were to be two main aspects: discrete mathematics and continuous mathematics. The latter covered the perceived need for calculus; the former was to deal with sets, sequences and maps. In addition both graph theory and linear algebra were to be introduced.

Working on the general principle of *cuius regio eius religio*, with *studium* replacing *religio* of course, the author determined the actual philosophy and details of the course which are summarized under three topics:

experiment Already from the very beginning it had been determined that the students *must learn to use Mathematica 3.0* (Wolfram Research, Inc. 1996), and that this particular tool, for which there was a campus-wide licence, would become a major aid in assisting them to overcome whatever mathematical inadequacies they had upon entering the university. In addition, *Mathematica* is destined to be their *e-companion* (electronic companion) throughout all of their degree years.

In this Freshman year, 30% of the marks were to be awarded for the work done using *Mathematica*. Assessment of such ‘course work’ was carried out by means of *e-tests* (i.e., open-book electronic exams using *Mathematica*).

Since the students had already completed two years of secondary school calculus it was hypothesized that much of the necessary calculus in this course would also be conveyed in *Mathematica* and that only a small amount of lecture time needed to be devoted to the subject. Some of the key ideas to be acquired were that (i) some functions were defined by integrals (e.g., the natural logarithm $\log x$), (ii) some functions were defined by power series (i.e., an extension of the idea of polynomial function and use of the Taylor series), and (iii) there was an intimate connection between discrete and continuous functions ($\log x$ and $\log n$) which were useful in practice (in Information and Communications Technology).

To reinforce the importance of the use of experiment in mathematics the conventional course structure of two lectures and one tutorial per week was changed to two lectures and one *Mathematica laboratory* per week.

application The course was presented to the students at the time of launch with the idea that one (might) produce a (mathematical) **model** of (some aspects of) the WORLD-WIDE WEB (or InterNet).

Graph theory was the subject matter proposed for this purpose.

ex æquo When two students are awarded the exact same marks in their final Moderatorship examinations they are classified *ex æquo*. The teaching of number theory to the freshmen achieves the exact same result. More colloquially and using sporting terminology we say that they all start on a level playing field. (For this insightful observation I am indebted to a colleague Dr. Hugh Gibbons, 1998).

Number theory became one of **the essential topics** of the foundation course. In their first year the students learned that, following Davenport, they could conduct an enormous number of experiments in number theory using *Mathematica*. For application, they reached the stage where they could do some elementary RSA public key cryptography.

4.1 The Texts

A choice of text had to be made. For the discrete mathematics that of Biggs (1989) was chosen. Not only did it cover most of the material but also (i) it did give a flavour of the kind of (discrete) mathematics that one could expect to find in Information and Communications Technology, and (ii) its language was mathematically sound.

Even though some of the students found the English language of the text difficult, it will continue to be the main text for the next few years and will set the standard for the first year Mathematics course.

The other major text was a basic text on Category Theory (Walters 1991). This is not an easy text for first year students. Nevertheless it seemed to be the best available at the time and to have the right focus on applications to computer science. We have since learned of (Lawvere and Schanuel 1997) and will recommend the first 100 pages as secondary reading. It is essential, in our opinion, that the students arrive at an understanding of ‘function’ that is more in keeping with the field of computer science than is found in the natural sciences. The variety of possibilities available through the concept of arrow is just what is needed.

Finally, at the end of the year the students are able to deal with their first major non-commutative algebraic structure—the finite state machine, the regular language that it recognizes and its associated syntactic monoid. This forms another highlight of the course.

5 Epilogue

“... we did not learn math as an ‘end’ but as a ‘means’. We learned how to use mathematics in developing and analysing product designs” (Parnas 1995b, 473).

The full impact of the new mathematics courses on the students of the degree in Information and Communications Technology will not be made known or become apparent for at least three more years. To begin in first year with a course that focuses on discrete mathematics, has a major number theory component and concludes with the non-commutative algebra of finite state machines is in itself innovative in this country, I believe. Much more important than this, however, is the basic philosophy of experiment in Mathematics which underpins the courses, even to the extent of doing mathematics laboratories rather than mathematics tutorials.

We are not in a position to reverse the trend of mass education and the possibility that second-level diplomas should become devalued. We *are* in a position to determine the nature and standard of the new Moderatorship degree in Information and Communications Technology in the University of Dublin. This paper presents just one aspect of that determination.

Now we come to the relationship between what is in the first year and what we would like to see in the third or fourth year. First we present a terse summary of the lessons learned teaching the mathematics of formal methods at the fourth year. Then we make a few remarks on the expected impact that a different foundation might have on such a course.

5.1 Lessons learned

General remarks.

- There is a considerable body of mathematics which underpins the model-theoretic formal methods and said body easily constitutes at least a full thirteen week semester course at third level.
- The mathematics is generally not provided for by mathematics departments nor computer science departments.
- The mathematics is completely independent of the formal logic which also underpins formal methods. The logic would also constitute a full thirteen week semester course.

Specifics.

- The discovery of the indexed monoid and the shift in the interpretations of expressions led to the realization that the mathematical structure of monoid was not sufficient to capture all of the interesting properties of a model. It seems that there is a sort of ‘natural’ clustering of monoids needed, a clustering which has not yet been uncovered. [Extensions to groups, modules, semi-rings, etc., do not solve the problem.]
- The boldness of using total functions within a constructive framework was truly imaginative and rewarding.

Over the ten year period one single factor has stood out. The students find the course very difficult. One hypothesizes that perhaps they do not have sufficient mathematical maturity or that they do not have the right mathematical foundation. Since this is a fourth year moderatorship course then we assume that the students taking it are as intelligent as any other fourth year moderators-to-be in the University.

The first year mathematics course briefly outlined in the paper is specifically designed to tackle the problem of foundations. The students must be able to hold together the discrete and the continuous branches of mathematics, one in each hand as it were, and to appreciate not only their difference but also their complementarity.

They need sufficient areas of application (in computing) in order to appreciate the mathematics. That is abundantly clear. We know that it is pedagogically sound. The incoming students do not have the right mathematical background for computing. We must change their mind-set in the first year.

Finally, as noted in the prologue, in order to progress we must somehow ‘bring down’ the necessary mathematics into the earlier years. This paper has attempted to give an account of a part of that ‘bringing-down’. Whether the mathematics can be brought down even further is left for the reader to ponder.

5.2 Acknowledgements

My thanks goes to all the members of the *Foundations and Methods Group* (FMG) on the one hand and the students on the other, at the University of Dublin, Trinity College, who have been patient and tolerant enough to have endured much of the development described in the paper.

Particular thanks are due to Alexis Donnelly and Claus Pahl who provided feedback on a very early and rough draft (in 1997).

It is Arthur Hughes who was largely responsible for leading the assault on the heights of classical algebra.

The insightful remarks of the two anonymous referees spurred me on considerably in the development of the material in sections 2 and 4, for which I am extremely grateful. They also pointed out numerous errors (called bugs in computer science, apparently) which had crept into an earlier version. I hope that I have caught and eliminated these and all others. As is customary, I acknowledge that any errors of whatever form that remain in the paper are my sole responsibility.

The commuting diagrams in this paper were produced by the $\text{\TeX}$ package `diagram.tex`¹ of Paul Taylor, Department of Computer Science, Queen Mary and Westfield College, London E1 4NS, UK (e-mail: pt@dcs.qmw.ac.uk).

The bibliography is produced with the aid of $\text{\MacBibTeX}$ ² of Vince Darley, Division of Applied Sciences, Harvard University, Oxford Street, Cambridge MA 02138, USA (e-mail: vince@das.harvard.edu).

¹<ftp://ftp.dcs.qmw.ac.uk/pub/tex/contrib/pt/diagrams/>

²<http://www.fas.harvard.edu/~darley/>

6 Glossary

$\mathbb{B}, \mathbf{2}$	the set of Booleans $\{0, 1\}$.
$\mathbb{N}$	the set of natural numbers $\{1, 2, \dots\}$.
$\mathbb{N}_0, \omega$	the set of numbers $\mathbb{N} \cup \{0\} = \{0, 1, 2, \dots\}$.
$\mathbb{N}_k, \mathbf{k}$	the finite sets $\{1, 2, \dots, k\}, \{0, 1, 2, \dots, k-1\}$.
$\mathbb{Z}$	the set of integers.
$\mathbb{Q}$	the set of rationals.
$\mathbb{R}, \mathbb{R}^+$	the set of real numbers, strictly positive real numbers.
$\mathbb{C}$	the set of complex numbers.
$\emptyset, \{\}$	the unique null set.
$\mathcal{P}-$	the (direct) powerset functor.
$\mathcal{P}f$	the iterating of map f over a set.
$\mathcal{P}X, \mathcal{P}'X$	the powerset of set X , $\mathcal{P}X$ excluding $\emptyset$.
$S \cup T, S \sqcup T, S \triangle T$	union, disjoint union, symmetric difference.
$S \cap T$	intersection.
$\triangleleft_T S, S \setminus T, S - T$	set difference.
$\chi_S(a), \chi[a]S$	characteristic function or subset classifier.
$a \in S$	test for set membership.
$ S , \#S, \text{card } S$	the cardinality of a set S .
$\text{proj}[\epsilon]S$	a projection that selects a random element of a set S .
$1, \Lambda, \langle \rangle$	the unique null sequence.
$-^*$	the sequence or free monoid functor.
f^*	the iterating of map f over a sequence.
Σ^*, Σ^+	sequences of elements over (alphabet) Σ , non-empty sequences.
$\Sigma_{\leq}^*, \Sigma^*_!$	sorted sequences, unique sequences, respectively.
$ \sigma , \#\sigma, \text{len } \sigma$	the length of a sequence σ .
$\sigma \cdot \tau$	concatenation of σ and τ .
$\text{elems } \sigma$	the set of elements in a sequence σ .
$\text{items } \sigma$	the bag of elements in a sequence σ .
$\text{proj}[j]\sigma$	a projection that selects the j th element of a sequence σ .

$\theta, []$	the unique null map.
$- \rightarrow -$	the map functor.
$f \rightarrow g$	the iterating of a pair of maps; f must be 1-1.
$X \rightarrow Y$	the space of maps from X to Y .
Y^X	the space of total maps from X to Y .
$\mathcal{I}$	the identity map.
$\emptyset^X$	the constant null (map) in the space $(\mathcal{P}Y)^X \subset (X \rightarrow \mathcal{P}Y)$.
$\text{dom } \mu, \text{rng } \mu$	the domain, range of the map μ .
$\mu \sqcup \nu$	the extend, or merge of two disjoint maps defined only if $\text{dom } \mu \cap \text{dom } \nu = \emptyset$.
$\mu \dagger \nu$	the override or overwrite of two maps.
$\mu \cup \nu$	the glueing of two maps which agree on $\text{dom } \mu \cap \text{dom } \nu$.
$\nu \circ \mu$	the composition of two maps $\mu \in X \rightarrow Y$ and $\nu \in Y \rightarrow Z$; defined over $\text{rng } \mu \cap \text{dom } \nu$; a strict version requires $\text{rng } \mu = \text{dom } \nu$.
$\mu \bowtie \nu$	the join of two maps $\mu \in X \rightarrow Y$ and $\nu \in X \rightarrow Z$; defined over $\text{dom } \mu \cap \text{dom } \nu$.
μ^{-1}	the inverse of the map μ , where it exists.
$\triangleleft_S \mu, \triangleleft[S]\mu, S \triangleleft \mu$	the removal of μ with respect to S ; classical mathematics uses $\mu \setminus S$.
$\triangleleft_S \mu, \triangleleft[S]\mu, S \triangleleft \mu$	the restriction of μ with respect to S ; classical mathematics uses $\mu \upharpoonright_S$.
$Y \rightarrow \mathcal{P}'X$	the (covering) space of inverse image maps.
$(\mathcal{I} \rightarrow \triangleleft_S)'$	the iterator $(\mathcal{I} \rightarrow \triangleleft_S)$ with removal of $y \mapsto \emptyset$ elements.

References

- Biggs, N. L. (1989). *Discrete Mathematics* (Revised ed.). Oxford: Clarendon Press. [ISBN 0-198-53427-2], Trinity College Dublin Library [510 M51*1 Hamilton Library] [PL-148-776 Stacks].
- Bourdieu, P. (1988). *Homo Academicus*. Cambridge: Polity Press. Peter Collier (trans.) from French “Homo Academicus”, Editions de Minuit, Paris, 1984.
- Coolahan, J. (Ed.) (1994). *Report on the National Education Convention*. Government Publications Sale Office, Dublin: The National Education Convention Secretariat. [ISBN 0-707-60343-9].
- Davenport, H. ([1952] 1992). *The Higher Arithmetic* (Sixth ed.). Cambridge: Cambridge University Press. [ISBN 0-521-41998-0], [Trinity College Dublin Library 512.7 K82*5 Hamilton Library].
- Donnelly, A., D. Gallagher, and A. Hughes (1996, March). On the Inheritance of Monoid Properties in Indexed Structures: A Tale of Three Proofs. Technical Report TCD-CS-96-04, Department of Computer Science, University of Dublin, Trinity College.
- Eccles, P. J. (1997). *An Introduction to Mathematical Reasoning, numbers, sets and functions*. Cambridge: Cambridge University Press. [ISBN 0-521-59718-8].
- Goldblatt, R. ([1979] 1984). *Topoi, The Categorical Analysis of Logic* (Revised ed.). Amsterdam: North-Holland.
- Holland, C. H. (1991). Trinity College Dublin and the idea of a university. In C. H. Holland (Ed.), *Trinity College Dublin and The Idea of a University*, pp. 9–28. Dublin: Trinity College Dublin Press. [ISBN 1-871-40805-9].
- Hughes, A. (1997). The semi-group of hash tables. Technical report, Department of Computer Science, University of Dublin, Trinity College.

- Hughes, A. P. and A. A. Donnelly (1995). An Algebraic Proof in VDM^{*}. In J. P. Bowen and M. G. Hinchey (Eds.), *ZUM'95: The Z Formal Specification Notation, Lecture Notes in Computer Science* **967**, pp. 114–33. Berlin: Springer-Verlag.
- Knuth, D. E. (1973). *The Art of Computer Programming, Vol. 3 Sorting and Searching*. Reading, Massachusetts: Addison-Wesley.
- Lawvere, F. and S. Schanuel (1997). *Conceptual Mathematics, A first introduction to categories*. Cambridge: Cambridge University Press. [ISBN 0-521-47817-0]. An earlier version was published by the Buffalo Workshop Press, 1991, with an Italian translation, Franco Muzzio &c editore spa in 1994.
- Mac an Airchinnigh, M. (1990). *Ph.D. Thesis: Conceptual Models and Computing*. University of Dublin, Trinity College, Dublin, Ireland: Department of Computer Science.
- Mac an Airchinnigh, M. (1991). Tutorial Lecture Notes on the Irish School of the VDM. In S. Prehn and W. J. Toetenel (Eds.), *VDM'91, Formal Software Development Methods Volume 2: Tutorials, Lecture Notes in Computer Science* **552**, pp. 141–237. Berlin: Springer-Verlag.
- Mac an Airchinnigh, M. (1993, May). Formal Methods & Testing. In *Tutorials of the Sixth International Software Quality Week*, 625 Third Street, San Francisco, CA 94107–1997. Software Research Institute.
- Mac an Airchinnigh, M. (1997). Towards a New Conceptual Framework for the Modelling of Dynamically Distributed Systems. In D. Duke and A. Evans (Eds.), *Northern Formal Methods Workshop, Ilkley 1996*, Electronic Workshops in Computing. London: Springer-Verlag. <http://www.springer.co.uk/ewic/workshops/>.
- Mac an Airchinnigh, M. and A. P. Hughes (1997). The Geometry of Distributions in Formal Methods. In D. Duke and A. Evans (Eds.), *2nd BCS-FACS Northern Formal Methods Workshop, Ilkley 1997*, Electronic Workshops in Computing. London: Springer-Verlag. <http://www.springer.co.uk/ewic/workshops/>.
- O'Reagan, G. (1997). *Ph.D. Thesis: Applying Formal Methods to Model Organizations and Structures in the Real World*. University of Dublin, Trinity College, Dublin, Ireland: Department of Computer Science.
- Parnas, D. L. (1995a). Language-Free Mathematical methods for Software Design, an Extend Abstract. In J. P. Bowen and M. G. Hinchey (Eds.), *ZUM'95: The Z Formal Specification Notation, Lecture Notes in Computer Science* **967**, pp. 3–4. Berlin: Springer-Verlag.
- Parnas, D. L. (1995b). Teaching Programming as Engineering. In J. P. Bowen and M. G. Hinchey (Eds.), *ZUM'95: The Z Formal Specification Notation, Lecture Notes in Computer Science* **967**, pp. 472–81. Berlin: Springer-Verlag.
- Pólya, G. ([1945] 1957). *How to Solve It, A New Aspect of Mathematical Method* (Second ed.). Princeton University Press.
- Pólya, G. (1981). *Mathematical Discovery, On understanding, learning, and teaching problem solving, two volumes* (combined ed.). New York: John Wiley & Sons.
- Stillwell, J. (1989). *Mathematics and Its History*. Berlin: Springer-Verlag.
- Walters, R. F. C. (1991). *Categories and Computer Science*. Number 28 in Cambridge Computer Science Texts. Cambridge: Cambridge University Press.
- Whitehead, A. N. ([1911] 1978). *An Introduction to Mathematics*. Oxford: Oxford University Press.
- Wolfram Research, Inc. (1996). *Mathematica* (3.0 ed.). Champaign, Urbana: Wolfram Research, Inc. The Program.