Towards a Topos Theoretic Foundation for the Irish School of Constructive Mathematics ($\mathsf{M}_{\mathsf{C}}^{\clubsuit}$)

Mícheál Mac an Airchinnigh
Draft: August 25, 2000

University of Dublin, Trinity College, mmaa@cs.tcd.ie.

Category of Paper: 2; comprising: *codification of domain knowledge, development of formal methods, theoretical foundations, pedagogy, and extensions.*

Abstract. The Irish School of Constructive Mathematics ($\mathsf{M}_{\mathsf{C}}^{\clubsuit}$), which comprehends and extends the VDM, exploits an algebraic notation based upon monoids and their morphisms for the purposes of abstract modelling (of computing systems). Its method depends upon an operator calculus. It owes its origins and much of its notation to the original VDM and derivations thereof. The School hereto eschewed every form of formal language and formal logic, relying solely upon constructive (and executable) mathematics. For reference it may be considered to be a model-theoretic specification language together with a well-formulated development method in the same camp as B and Z.
In the spirit of the unified theories of programming text of Tony Hoare and Hifeng Je, Cliff Jones' call for unified theories of modelling and/or specification, and the illuminating development of Milner's π-calculus, the School is committed to the development of the modelling of computing (and in general other non-computing) systems in full generality.
This shift was due primarily to research in the geometry of formal methods, initially triggered by the discovery that the tail-recursive map of the length homomorphism of the free monoid could be extended in a natural way to an affine transformation in the usual geometric sense. This contact point, albeit tenuous, was considerably elucidated by the modelling of a hash table as a (trivial) fibre bundle.
From fibre bundles to sheaves was a natural step. Concurrently, the School moved from the algebra of monoids to categories, and from categories to topoi. Finally, the constructive nature of the School is now coming to terms with formalism and logic through the (natural) intuitionistic logic inherently manifest through topoi.
In this paper we exhibit a suitably accessible trajectory or bridge from classical model-theoretic formal methods to the (in our considered opinion) more natural and (consequently self-evident!) universal topos-theoretic formal methods in the goal towards unification.

keywords: cartesian closed category, constructive mathematics, education, Heyting algebra, intuitionistic logic, method, modelling, pedagogy, process calculi, specification, topos, VDM.

1 Prologue

> "[. . .] computer science is in deep crisis, expanding, fragmenting, and specializing faster, faster than any other discipline, faster than anyone can understand, let alone predict. Moreover computer science is increasingly seen as marginal to its applications, and this is particularly true of theoretical computer science" (Goguen 1999, 94).

This quotation of Goguen, taken from his paper *Tossing Algebraic Flowers down the Great Divide*, might it not also be extended to another area of computer science, software engineering and within the embrace of which, formal methods? Has not a great gulf opened up between the overtly very successful practice of software development as manifested especially in recent times on personal computers and by the World-wide Web, and the need to capture, model, codify, and transmit the knowledge gained by this great ongoing scientific computing experiment in which we all participate and that we all experience? How can we build the bridges of various shapes and sizes and function to cross this divide?

As we all well know, every science tries to confirm (and extend) its knowledge by performing experiments based on codified theories expressed in the language of mathematics. There is, of course, a natural relationship been mathematics and logic, and some take the view that 'logic is primary' and conclude that "predicates describing the world are sometimes called Laws of Nature"(Hoare and He 1998, 26). However, this view of the world is *not* shared by all (Körner 1960), as Hoare and He (1998, 28) acknowledge. It is moreover very unlikely that Newton, Einstein, and Feynman (inter alii) would have formed such an opinion as to the (alleged) priority of logic! The supposed centrality is a very recent invention! Nevertheless, the beautiful work of unification in the text cited is undoubtedly one bridge of a particular shape and function.

In his paper describing some of the 'pre-history' of the VDM, Jones (1999, 43) has indicated that Abrial's "Abstract Machine Notation in B does fit more closely with aspects of VDM than Z". He is also reputed to have said recently (2000) that perhaps there might be the possibility of a merging between VDM and B, at least. In his opinion, what really matters is the communication of "the idea of *abstract modelling"* as a way of understanding computer systems (Jones 1999, 43). Jones' efforts are clearly another very welcome attempt to provide a specification bridge for the model-theoretic languages.

The Irish School of Constructive Mathematics ($\mathsf{M}_C^{\clubsuit}$), originally founded upon the Irish School of the VDM (est. 1990) and described at length in (Mac an Airchinnigh 1991), has hitherto deliberately avoided the issue of the need for a formal logic, even the *Logic of Partial Functions* (LPF) asserted to be part of VDM by Cliff Jones (Jones 1999, 42), to underpin both the specification of operations and constraints in the mathematical models, and in the application of the well-formulated development steps of its method.

The rationale for the deliberate omission of formal logic was simply based on two factors: (i) that there was a strong *constructive* nature of the specifications and the developments, and that the performing of proofs were effectively

either constructions, or algebraic transformations into constructions, in other structures, and (ii) that it was very unlikely that the psychology of (software) engineers was compatible with such formal logic! The nearest one might get to an underlying formal logic was in the specification of well-formedness constraints, invariants, and the *pre-conditions* of the operations of a model which were entirely elementarily set-theoretic. It was certainly not customary to use explicit notations for universal and existential quantifiers! Should one wish to use a formal logic in conjunction with the models constructed in the style of the Irish school then it was considered to be *an extremely eccentric complementary* action. The emphasis was entirely on the algebra.

The specification "language" of the school is deliberately not formal in principle in order to achieve greater mathematical flexibility and expressiveness. Thus no distinction is made between syntax and semantics. All is mathematics. This is a central philosophical tenet of the school. There is nothing whatever, on the other hand, which prohibits the elaboration of a formal language with associated syntax and semantics which respects that philosophical tenet. Colman Reilly explored such a relationship through Mathematica (Reilly 1995) and Andrew Butterfield is currently (2000) working on a fully-fledged `Clean` interpretation. In other words being constructive, specifications in the Irish VDM are necessarily executable.

Another *great divide* is that between state-based formal methods (structure-focussed or state-focussed) and process calculi (action-focussed). Our work was always hitherto focussed on structural aspects of computing and avoided process aspects. In (Mac an Airchinnigh 1990) is described an aborted attempt to wed structure with process. We have not ignored process calculi. It is simply the case that we have not been able to reconcile our mathematical understanding of two apparently diverse realities: structure and process, and therefore to find a common language to describe both, in harmony. Recently, and from a very different direction, Malcolm Tyrrell, of our Foundations and Methods Group, has achieved some considerable success in adding *state* to process calculi such as the π-calculus (Tyrrell, Butterfield, and Donnelly 2000). This brings the two sides closer.

Based upon the success of other unification initiatives in specifications and programming languages we are persuaded within the School to move to a *topos* theoretic foundation for several reasons.

1. the universal properties of category theory and its ubiquity in computing provides a sound semantic basis for both structure and process. See (Hoare 1999, 25–6);
2. topos theory and sheaf theory provide a natural unification of algebra, logic, and geometry. See (Mac Lane and Moerdijk 1992, 1);
3. the intuitionistic logic associated with the topos is compatible with the constructive philosophy of the School;
4. perhaps most important of all, accessible textbooks on the subject of category theory and topos theory are now available for first year undergraduates

at University. See especially *Conceptual Mathematics, a first introduction to categories* (Lawvere and Schanuel 1997).

To achieve a successful transition to a topos theoretic foundation we do not wish to lose the current 'user-friendliness' of the existing notation. On the other hand we do want to move to expressive forms that are clearly and unambiguously within constructive mathematics and which are sound from a topos theoretic perspective. In addition it is absolutely essential that we meet

> "the challenge to make toposes as intuitive from the beginning as they are to experts, especially as concerns topos logic" (McLarty 1992, vii–viii).

Mathematics is used to codify scientific knowledge. But there are many different styles of mathematics and notations in which it is expressed. Domain knowledge of computing is codified by abstract modelling using both mathematics and logic. The more views one has of a particular domain concept the better the understanding. That is why algebraic and geometric views of the same concept, say circle, are so valuable. Each provides its own way of understanding and manipulating the object in question. We also need complementary views of computing domain objects. This paper proposes a specific way forward which is being adopted by the School for model-theoretic specification languages.

The rest of the paper is organised as follows.

Section §2 presents three basic operations of the classical model of the spelling checker dictionary: write, read, and remove in order to provide a common basis of understanding for the reason for the need for a bridge to topos theory. In section §3 we introduce the structures of Heyting algebra, cartesian closed category, and topos and demonstrate how the spelling checker dictionary model may be suitably transformed in order that it conform to the newly introduced structures. Then, in section §4, we explore the fibering of a classic VDM map and show how intuitionistic quantifiers can be introduced naturally. The paper then concludes with some remarks on future work.

2 Classical spelling checker dictionary

> "The view that [set and set] membership is primary [in contradistinction to map or function or process] also leads one to believe that [set] membership is global and absolute, whereas in fact it is local and relative" (Lawvere 1975, 5).

Essentially, the dictionary we have in mind is like that used in conjunction with the board game SCRABBLE® such as the Official Scrabble Players Dictionary (Selchow & Righter Company 1978). In the case of a dispute between two players over the spelling of a word there is an agreed procedure whereby the spelling is checked with respect to the occurrence of the word in the "standard dictionary", but only after the player has made the play. If the word is in the dictionary then it is an acceptable word for the play, otherwise ...

In this real world scenario the essential operation from an end-user's point of view is the checking whether or not a given word is in the dictionary. We call such an operation (in the context of this and other models) a lookup. It is the same as the read operation in other computing contexts such as data base lookups.

We will model such a dictionary by using sets. We start with a set of words $WORD$ and then construct the powerset $\mathcal{P}\,WORD$. Elements of the powerset, i.e., sets of words, are considered to be dictionaries. We already know that if n is the size of the set $WORD$ then there are 2^n possible dictionaries. Naturally, what we have just described is a classic *Gedanken* experiment. In practice, we do not start with some set $WORD$ and apply the powerset operator to give us our space of dictionaries. Instead we work from a starting point of the empty dictionary and build whatever we need.

More formally, consider the usual domain equation for the most abstract model of the spelling checker dictionary:

$$\delta \in DICT = \mathcal{P}\,WORD \tag{1}$$

where $\mathcal{P}$ denotes the usual powerset functor. The expression $\mathcal{P}\,WORD$ provides us with a Boolean algebra $\mathcal{B}$ in a natural way. We will be more general and assume an underlying Heyting algebra $\mathcal{H}$ instead. A formal definition is given in the next section. A Boolean algebra is a Heyting algebra. In other words, the property of being Heyting is more general than the property of being Boolean. The main reasons for the change are fourfold.

1. the Heyting algebra provides an algebraic semantics for propositional intuitionistic logic, whereas the Boolean algebra provides an algebraic semantics for propositional classical logic (Mac Lane and Moerdijk 1992, 48–9), (Fitting 1969, 23).
2. the Heyting algebra itself has a semantics in the set of all the open subsets of a topological space (Mac Lane and Moerdijk 1992, 48–9); as a corollary one may introduce topological notions directly into computing via the Heyting algebra; more precisely, a (complete) Heyting algebra is a *frame* (geometric view) or *locale* (algebraic view) (Mac Lane and Moerdijk 1992, 472–5);
3. the Heyting algebra is a cartesian closed category and a cartesian closed category is of particular universal interest because it has, in an elegant manner, essentially the same expressive power as a typed λ-calculus (Barr and Wells 1995, 175) (Lambek and Scott 1986, 41).
4. finally, the step from cartesian closed category to topos is a small one, but one which introduces the notion of 'truth object', and hence which provides the natural logic to go with the algebra.

The domain equation is read "let δ be an arbitrary element chosen from the model named $DICT$, the structure of which is given by $\mathcal{P}\,WORD$." Since we are moving away from membership based expressions (in the short term) we will need to find another way of expressing the fact that δ is a typical object in the structure under consideration.

Of immediate importance here, $\mathbb{B}$ denotes the usual two-valued logic and specifically denotes the set of two 'truth values', $\{0, 1\}$ where 1 denotes true and 0 denotes false. The corresponding set of many truth values will be denoted by $\mathbb{H}$. It is immediately to be noticed that always $\mathbb{B} \subseteq \mathbb{H}$.

We shall focus only on the usual write, read, and remove operations, here called enter, lookup, and remove. In the remainder of this section we give the classical form of the specification we have hitherto used.

The write operation: enter the entering of a *new* word into an existing dictionary is captured by

$$Ent\colon WORD \longrightarrow DICT \longrightarrow DICT \tag{2}$$

$$Ent[w]\delta := \{w\} \cup \delta \tag{3}$$

The expression $\{w\} \cup \delta$ is well-defined within the Heyting algebra and, therefore, does not have to be replaced. This operation is subject to the pre-condition or guard which captures the idea that the word w is *new*:

$$\mathsf{pre}\text{–}\, Ent\colon WORD \longrightarrow DICT \longrightarrow \mathbb{B} \tag{4}$$

$$\mathsf{pre}\text{–}\, Ent[w]\delta := w \notin \delta \tag{5}$$

The expression $w \notin \delta$ is taken to be equivalent to the predicate $\neg(w \in \delta)$. There are two ideas that need to be examined. The first is that of *set membership*. In a topos-theoretic foundation set membership is not a primary concept. Therefore, we must find a suitable alternative here. The second idea is, of course, that of *negation*. We need to deal with that also.

The read operation: lookup to look up a word in the dictionary is to ask whether or not it is present in that dictionary. There is no pre-condition.

$$Lkp\colon WORD \longrightarrow DICT \longrightarrow \mathbb{B} \tag{6}$$

$$Lkp[w]\delta := w \in \delta \tag{7}$$

We expect the result of the lookup to be either true or false. The expression $w \in \delta$ is not an appropriate expression within the Heyting algebra. We will have to find an acceptable alternative.

The remove operation the removal of an *existing* word from the dictionary is usually specified by

$$Rem\colon WORD \longrightarrow DICT \longrightarrow DICT \tag{8}$$

$$Rem[w]\delta := \delta \backslash \{w\} \tag{9}$$

Set removal $\delta \backslash \{w\}$ is not an appropriate expression. We will provide an alternative. This operation is subject to the pre-condition

$$\mathsf{pre}\text{–}\, Rem\colon WORD \longrightarrow DICT \longrightarrow \mathbb{B} \tag{10}$$

$$\mathsf{pre}\text{–}\, Rem[w]\delta := w \in \delta \tag{11}$$

Let us now examine each operation in turn and consider whether the specification is fully constructive. In order to be complete and, therefore, comprehensive, we also need to say something about the form of the signatures. Finally, we recall the fundamental philosophical distinction between the concept of proof in classical and constructive mathematics.

> "In classical mathematics, a proposition is thought of as being true or false independently of whether we can prove or disprove it. On the other hand, a proposition is constructively true only if we have a method of proving it." (Nordström, Petersson, and Smith 1990, 11)

3 Intuitionistic spelling checker dictionary

> " The Boolean term $S \vee \bar{L}$ is often written as an implication (e.g., $L \supset S$); indeed, the above law,
>
> $$\frac{P \Rightarrow S \vee \bar{L}}{P \wedge L \Rightarrow S} \quad ,$$
>
> together with the inference in the opposite direction, is used in intuitionistic logic to define implication [... which] is always a predicate [and being] antimonotonic in its first argument, it will rarely be a program " (Hoare 1999, 8).

Who could possibly resist exploring the consequences of such a statement? What is the nature of implication and its role in intuitionistic logic which would render it almost useless as a program?

As a first step towards the construction of an intuitionistic spelling checker dictionary we shall introduce the mathematical structures of Heyting algebra $\mathcal{H}$ and cartesian closed category. Then we shall recast each of the operations on the spelling checker dictionary in terms of these structures. It will be assumed that the reader is already familiar with the elements of Category Theory. A highly recommended introductory text is *Conceptual Mathematics, A First Introduction to Category Theory* (Lawvere and Schanuel 1997).

For a *working* definition of Heyting algebra we follow (Fitting 1969, 23). Note that Fitting uses an "older name" for the Heyting algebra: the pseudo-boolean algebra. We consider the name Heyting algebra more appropriate. The definition is cast within traditional Set Theory. Note in particular that we have deliberately 'mapped' the algebra to the logic. This is in conformance with an old mathematical tradition. Strict formalists (and logicians) prefer separation. See (Fitting 1969) for details.

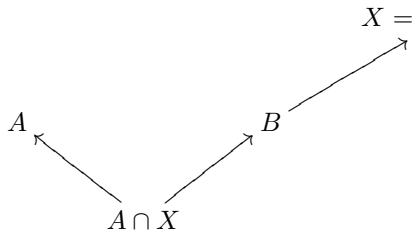
Definition 1 (Heyting algebra) A Heyting algebra is a pair $\langle \mathcal{H}, \leq \rangle$ where $\mathcal{H}$ is a non-empty set and $\leq$ is a partial ordering relation on $\mathcal{H}$ such that for any two elements A and B of $\mathcal{H}$:

1. the least upper bound $A \vee B$ exists [to correspond with *logical or* or disjunction];

2. the greatest lower bound $A \wedge B$ exists [to correspond with *logical and* or conjunction];
3. the pseudo complement of A relative to B denoted $A \Rightarrow B$, defined to be the largest $X \in \mathcal{H}$ such that $A \wedge X \leq B$, exists [to correspond with *logical implication*];
4. a least element called bottom, denoted $\bot$, exists [to correspond with false].

Let the complement of A, denoted $\neg A$, be $A \Rightarrow \bot$ [to correspond with *logical negation*]. Note that the complement of A is the pseudo complement of A relative to $\bot$. Let the top element, denoted $\top$ be $\neg\bot$ [to correspond with true]. Clearly, then $\top = (\bot \Rightarrow \bot)$. In fact, in general $(A \Rightarrow A) = \top$. Singleton sets of the algebra are called atoms. It is to be noted that there exist Boolean algebras and hence Heyting algebras which do not have atoms. See (Stoll 1974, 211). However, this will not affect our presentation.

In the case of the spelling checker dictionary we take $\mathcal{H} = \mathcal{P}\,WORD, \subseteq$ for the partial ordering relation, $\bot = \emptyset$, $\top = WORD$, the least upper bound of A and B in $\mathcal{H}$ is given by $A \cup B$, and the greatest lower bound of A and B in $\mathcal{H}$ is given by $A \cap B$. Note that the pseudo complement operator $\Rightarrow$ is distinctively new! Its role may be exemplified by the following diagram where the arrow $A \longrightarrow B$ denotes $A \subseteq B$.



(12)

Definition 2 (Boolean algebra) A Boolean algebra is a Heyting algebra with the special property that for every A in $\mathcal{H}$, $\neg A \vee A = \top$.

It will be demonstrated that, with appropriate modifications, the spelling checker dictionary model, *is* a Heyting algebra.

Before we proceed to that demonstration let us look at the categorical foundations. First, it is well-known that a Heyting algebra is a cartesian closed category. Following (Mac Lane and Moerdijk 1992, 20) we define a cartesian closed category.

Definition 3 (Cartesian closed category) A category $\mathcal{C}$ is called *cartesian closed* if it has finite products (i.e., a terminal object and binary products) and if all objects of $\mathcal{C}$ are exponentiable.

It is a simple exercise to demonstrate that the Heyting algebra is a cartestian closed category. Let A and B be two objects in the category. Then, if $A \leq B$ we have the map $A \longrightarrow B$. In this view of the Heyting algebra, the terminal object

is $\mathbf{1} = \top$. Binary products are given by $\langle A, B \rangle \mapsto A \wedge B$. The exponentiable objects are $B^A = (A \Rightarrow B)$.

Since we will demonstrate that the spelling checker dictionary is a Heyting algebra then it is also a cartesian closed category.

DEFINITION 4 (ELEMENTARY TOPOS) An elementary topos $\mathcal{T}$ is a *cartesian closed category* which has a truth value object Ω.

Essentially, this means that such a topos provides us with an intrinsic logic. That logic is generally intuitionistic ((Mac Lane and Moerdijk 1992, 268)). According to Szabo (1978, 190) the internal logic of an elementary topos is "strictly weaker than intuitionistic logic ... certain intuitionistically valid formulas such as $(\forall\xi)\phi(\xi) \Rightarrow (\exists\xi)\phi(\xi)$ no longer hold." Consequently, one needs to be very careful in scoping out a topos of the right shape to accommodate the constructions we are interested in.

3.1 Enter

Recall that the classical specification is

$$Ent\colon WORD \longrightarrow DICT \longrightarrow DICT \tag{13}$$
$$Ent[w]\delta := \{w\} \cup \delta \tag{14}$$

In the context of the Heyting algebra we recognize the dictionary δ as a Heyting subalgebra. The entering of a new word w is the extension of the existing set of atoms of δ by $\{w\}$. Clearly, therefore, the specification may be interpreted properly as the extension of *structure* denoted by

$$\delta = \{\{a\}, \{b\}, \{c\}, \dots\} \mapsto \{\{a\}, \{b\}, \{c\}, \dots, \{w\}\} \tag{15}$$

The Heyting algebra can be recovered by applying the operations $\cup$, $\cap$, and $\Rightarrow$. For example, $\emptyset$ is recovered from $\{a\} \cap \{b\}$, say, and complement is then determined by $\neg\{w\} = (\{w\} \Rightarrow \emptyset)$. In practice, we can use a sequence of words, canonically ordered lexicographically, to represent the set of atoms. This gives us a simple but direct structural implementation of the Heyting algebra and hence of the corresponding spelling-checker dictionary. There is a direct relationship between structure and function which is now becoming evident.

We can extend the original specification to allow for this extra structure:

$$Ent\colon WORD \longrightarrow DICT \times WORD_{\leq}^{\star} \longrightarrow DICT \times WORD_{\leq}^{\star} \tag{16}$$
$$Ent[w]\langle \delta, \alpha \rangle := \langle \{w\} \cup \delta, \sigma(\langle w \rangle \cdot \alpha) \rangle \tag{17}$$

where α is the ordered sequence of atoms of δ and σ is a sorting morphism on sequences.

There is still the usual need for an invariant here to guarantee that the words in the dictionary $\{w\} \cup \delta$ correspond exactly to the set of atoms in $\sigma(\langle w \rangle \cdot \alpha)$. If

we introduce a primitive function atoms on a Heyting algebra $\mathcal{H}$ which returns its set of atoms then we may write the appropriate invariant as

$$\mathsf{inv}\text{–}Ent\colon WORD \longrightarrow DICT \times WORD_{\leq}{}^{\star} \longrightarrow \mathbb{H} \tag{18}$$

$$\mathsf{inv}\text{–}Ent[w]\langle\, \delta, \alpha \,\rangle := \mathsf{atoms}(\{w\} \cup \delta) = \mathsf{elems}(\sigma(\langle\, w \,\rangle \cdot \alpha)) \tag{19}$$

Now let us take a closer look at the pre-condition for the enter operation. We propose to reject the particular expression $w \notin \delta$ in favour of $\{w\} \cap \delta = \emptyset$. The reasoning for the change is as follows.

The expression $w \notin \delta$ is read as "the word w is not an element of the set δ", which expression we abbreviate as $\neg(w \in \delta)$. Let us first look at the simpler form $w \in \delta$. This expression is interpreted in the context of a membership based set theory such as Zermelo-Fraenkel (ZF) set theory. However, there is a central difference between sets in a *well-pointed topos* and sets approached via membership (McLarty 1992, 215). To paraphrase McLarty, in the topos of Sets $\mathcal{S}$ we can take a set $WORD$ and ask whether a given element w of $WORD$ is a member of a given subset δ of $WORD$, but it is pointless to ask whether an element of $WORD$ is also an element of some other set, $DUCK$, say.

To comprehend this radically different view of reality one needs to understand how points and elements are defined and used. In general, a point x in a topos $\mathcal{T}$ is a map $\mathbf{1} \xrightarrow{x} A$ from the terminal object $\mathbf{1}$ to the object A. Objects need not have any points whatsoever. In the category of Sets $\mathcal{S}$ the points of an object A correspond exactly to the elements of the set A. In a general topos $\mathcal{T}$ such points are called global elements. If in a general topos a pair of maps $A \underset{g}{\overset{f}{\rightrightarrows}} B$ are equal, $fx = gx$, for every general element x then the topos $\mathcal{T}$ is said to be well-pointed (Mac Lane and Moerdijk 1992, 236).

A Heyting algebra is not in itself a topos. It seems to fail by a very slight margin. As a cartesian closed category, the only point is $\mathbf{1} \longrightarrow \mathbf{1}$. This prevents us from having a non-trivial truth object. It appears at first glance to be a strange and worrisome result. However, once one becomes accustomed to the view that a space might reasonably be considered to be composed of parts rather than points(Lawvere 1975, 32), then one is freed from a certain blinkered view. Therefore, we need to embed the Heyting algebra in a suitable topos in order to achieve the desired goal. On the other hand, Heyting algebras are plentiful in any topos. Specifically, for any object A in a topos, the power object $\mathcal{P}A$ is a(n internal) Heyting algebra and, as a special case, so is the truth value object $\Omega = \mathcal{P}\mathbf{1}$ (Mac Lane and Moerdijk 1992, 201).

For the present section we content ourselves to the transformation of the dictionary in a Heyting algebra compatible form.The choice of suitable topoi, compatible with the VDM, is still under active investigation.

First we observe that $\neg(w \in \delta)$ can be written in terms of the Heyting algebra operations as $\neg(\{w\} \subseteq \delta)$, read as "the atom $\{w\}$ does not belong to the subalgebra δ." If the atom $\{w\}$ does **not** belong to δ then it must belong somewhere and that somewhere is the complement of the subalgebra, denoted

$\neg\delta$. In other words we have the **fundamental equivalence**

$$\neg(\{w\} \subseteq \delta) \text{ iff } \{w\} \subseteq \neg\delta \tag{20}$$

But by the definition of complement in a Heyting algebra $\neg\delta$ is the exponential $\delta \Rightarrow \emptyset$. Hence we have

$$\neg(\{w\} \subseteq \delta) \text{ iff } \{w\} \subseteq (\delta \Rightarrow \emptyset) \tag{21}$$

Now we focus on the espression $\{w\} \subseteq (\delta \Rightarrow \emptyset)$. Since a Heyting algebra is a cartesian closed category then from the basic adjunction relating products and exponentials

$$\frac{Z \longrightarrow Y^X}{Z \times X \longrightarrow Y} \tag{22}$$

we make the obvious substitutions to obtain (Mac Lane and Moerdijk 1992, 50):

$$\frac{z \leq (x \Rightarrow y)}{(z \wedge x) \leq y} \tag{23}$$

Now substituting $z \mapsto \{w\}$, $x \mapsto \delta$, $y \mapsto \emptyset$, $\leq \ \mapsto \ \subseteq$, and $\wedge \ \mapsto \ \cap$ we obtain

$$\frac{\{w\} \subseteq (\delta \Rightarrow \emptyset)}{(\{w\} \cap \delta) \subseteq \emptyset} \tag{24}$$

This gives us

$$\neg(\{w\} \subseteq \delta) \text{ iff } (\{w\} \cap \delta) \subseteq \emptyset \tag{25}$$

Since $\emptyset$ is bottom then we also have the *fact* that

$$\emptyset \subseteq (\{w\} \cap \delta) \tag{26}$$

Hence

$$\neg(\{w\} \subseteq \delta) \text{ iff } (\{w\} \cap \delta) \subseteq \emptyset \wedge \emptyset \subseteq (\{w\} \cap \delta) \tag{27}$$

$$(\{w\} \cap \delta) = \emptyset \tag{28}$$

and this is our desired pre-condition in the Heyting algebra. We summarise this derivation as follows:

$$\neg(w \in \delta) \tag{29}$$

$$\neg(\{w\} \subseteq \delta) \tag{30}$$

$$\neg(\{w\} \subseteq \delta) \text{ iff } \{w\} \subseteq \neg\delta \tag{31}$$

$$\{w\} \subseteq \neg\delta \text{ iff } \{w\} \subseteq (\delta \Rightarrow \emptyset), \quad \text{by definition} \tag{32}$$

$$\{w\} \subseteq (\delta \Rightarrow \emptyset) \text{ iff } \{w\} \cap \delta \subseteq \emptyset, \quad \text{by adjunction} \tag{33}$$

$$\{w\} \cap \delta = \emptyset \tag{34}$$

Hence the pre-condition or guard for the "enter a new word" operation may be written in the form

$$\mathsf{pre}\text{–}\, Ent\colon\ WORD \longrightarrow DICT \longrightarrow \mathbb{H} \tag{35}$$

$$\mathsf{pre}\text{–}\, Ent[w]\delta := \{w\} \cap \delta = \emptyset \tag{36}$$

This is a sensible specification from the point of view of the end-user. This pre-condition may also be expressed in the more 'exotic' forms of

$$\mathsf{pre}\text{–}\, Ent\colon\ WORD \longrightarrow DICT \longrightarrow \mathbb{H} \tag{37}$$

$$\mathsf{pre}\text{–}\, Ent[w]\delta := \{w\} \cap \delta \subseteq \emptyset \tag{38}$$

or

$$\mathsf{pre}\text{–}\, Ent\colon\ WORD \longrightarrow DICT \longrightarrow \mathbb{H} \tag{39}$$

$$\mathsf{pre}\text{–}\, Ent[w]\delta := \{w\} \subseteq \neg\delta \tag{40}$$

3.2 Remove

Consider the meaning of set difference $A - B$ where A and B are subsets of some ambient or universal set U. We may write $A - B$ in the form $A \cap (-B)$ where $-B$ is the complement of B with respect to U. Now in the Heyting algebra $-B$ is defined to be $B \Rightarrow \emptyset$. Hence we have the definition

$$A - B := A \cap (B \Rightarrow \emptyset) \tag{41}$$

This leads directly to an intuitionistic definition of the removal operation.

$$Rem\colon\ WORD \longrightarrow DICT \longrightarrow DICT \tag{42}$$

$$Rem[w]\delta := \delta \cap (\{w\} \Rightarrow \emptyset) \tag{43}$$

subject to the pre-condition that w is in the dictionary:

$$\mathsf{pre}\text{–}\, Rem\colon\ WORD \to DICT \to \mathbb{H} \tag{44}$$

$$\mathsf{pre}\text{–}\, Rem[w]\delta := (\delta \cap \{w\}) = \{w\} \tag{45}$$

Let us consider the definition of the remove operation first. The expression $\delta \cap (\{w\} \Rightarrow \emptyset)$ can hardly be considered intuitive to the end-user *at the present time*. Nor does it seem very constructive. A more end-user friendly form might be $\delta \cap \neg\{w\}$ for which we might agree to use the abbreviation $\delta - \{w\}$ or $\delta \backslash \{w\}$, to get back to where we started.

Turning now to the pre-condition which classically was $w \in \delta$ and which is given here as $(\delta \cap \{w\}) = \{w\}$. Again it is intuitively clear that the new definition is correct. However, it is of interest to attempt to derive this from the classical expression. Already we have agreed above that $\{w\} \subseteq \delta$ is the equivalent to the membership expression. Hence we have a first reasonable and directly accessible specification of a pre-condition:

$$\mathsf{pre}\text{–}\, Rem\colon\ WORD \to DICT \to \mathbb{H} \tag{46}$$

$$\mathsf{pre}\text{–}\, Rem[w]\delta := \{w\} \subseteq \delta \tag{47}$$

3.3 The pre-conditions

Let us now take a closer look at the intuitionistic pre-conditions which we have already specified. In the case of pre– $Ent[w]$ we replaced $w \notin \delta$ by $\{w\} \cap \delta = \emptyset$ and in the case of pre– $Rem[w]$ we replaced $w \in \delta$ by $(\delta \cap \{w\}) = \{w\}$. But we noticed that there were other possibilities. For example, in the case of pre– $Rem[w]$ above we suggested the use of $\{w\} \subseteq \delta$. Let us demonstrate formally, that from this expression we can derive algebraically, $\{w\} \subseteq \mathbf{1}$, in much the same way that from $w \in \delta$ one deduces $w \in WORD$ from the containment $\delta \subseteq WORD$.

$$w \in \delta \tag{48}$$

$$\{w\} \subseteq \delta \tag{49}$$

$$\delta \cap \{w\} \subseteq \delta \cap \delta \text{ implies } \delta \cap \{w\} \subseteq \delta \tag{50}$$

$$\delta \cap \{w\} \subseteq \delta \text{ implies } \{w\} \cap \delta \subseteq \delta, \quad \text{by commutativity of } \cap \tag{51}$$

$$\{w\} \cap \delta \subseteq \delta \text{ iff } \{w\} \subseteq (\delta \Rightarrow \delta), \quad \text{by adjunction} \tag{52}$$

$$\{w\} \subseteq \mathbf{1} \tag{53}$$

$$\text{i.e., } w \in WORD \tag{54}$$

Perhaps we need to comment upon $(\delta \Rightarrow \delta) = \mathbf{1}$. By definition, $(\delta \Rightarrow \delta)$ is the largest X in $\mathcal{P}\,WORD$ such that $\delta \cap X \subseteq \delta$. Such an X is clearly $WORD$ and $WORD = \mathbf{1}$ in this Heyting algebra.

3.4 Test

In reflecting upon the structural forms of the intuitionistic pre-conditions for both the enter operation

$$\text{pre–}\, Ent[w]\delta := (\{w\} \cap \delta) = \emptyset \tag{55}$$

and the remove operation

$$\text{pre–}\, Rem[w]\delta := (\{w\} \cap \delta) = \{w\} \tag{56}$$

it is clear that they both have the general form $A \cap B = C$. Therefore, it seems appropriate to consider a *new* operation on the dictionary that generalises these expressions. For historical reasons we call this the *test* operation. The formal definition is

$$Tst\colon \mathcal{P}\,WORD \to DICT \to DICT \tag{57}$$

$$Tst[S]\delta := S \cap \delta \tag{58}$$

Using the Tst operation then the pre-conditions for Ent and Rem become,

$$\text{pre–}\, Ent[w]\delta := Tst[\{w\}]\delta = \emptyset \tag{59}$$

and

$$\text{pre–}\, Rem[w]\delta := Tst[\{w\}]\delta = \{w\} \tag{60}$$

respectively. Knowing that, in practice, one pre-condition is the *opposite* of the other, i.e., that **not** pre– $Rem[w]\delta$ = pre– $Ent[w]\delta$ and assigning "true" to pre– $Rem[w]\delta$ entails assigning "false" to pre– $Ent[w]\delta$. Clearly, we can generalise this to give the truth assignments

$$Tst[S]\delta = \emptyset \mapsto \mathsf{false}, \text{ if } S \cap \delta = \emptyset \tag{61}$$

$$Tst[S]\delta = S' \mapsto \text{ the degree of truth measured by } S' \subseteq S \tag{62}$$

$$Tst[S]\delta = S \mapsto \mathsf{true}, \text{ if } S \cap \delta = S \tag{63}$$

Hence, we do have a natural underlying multi-valued logic. Note in particular that since $Tst[S]\delta = S \cap \delta = S$ then the last equation is equivalent to

$$Tst[S]\delta = \mathbf{1} \mapsto \mathsf{true} \tag{64}$$

in the Heyting subalgebra δ of $\mathcal{P}\,WORD$.

3.5 A simple proof

To conclude this section we present a simple proof in the new style.

Consider the proof of the assertion that *if one enters a new word w into a dictionary δ and then removes that word the result is the original dictionary δ that one started with.* Constructively, we have

$$(Rem[w] \circ Ent[w])\delta \tag{65}$$

$$= Rem[w](Ent[w]\delta) \tag{66}$$

$$= Rem[w](\{w\} \cup \delta) \tag{67}$$

$$= (\{w\} \cup \delta) \backslash \{w\} \tag{68}$$

$$= (\{w\} \cup \delta) \cap (\{w\} \Rightarrow \emptyset) \tag{69}$$

$$= (\{w\} \cap (\{w\} \Rightarrow \emptyset)) \cup (\delta \cap (\{w\} \Rightarrow \emptyset)) \tag{70}$$

$$= \emptyset \cup \delta \tag{71}$$

$$= \delta \tag{72}$$

The noteworthy aspects of the proof are at (70) where the reduction of $\{w\} \cap (\{w\} \Rightarrow \emptyset)$ to $\emptyset$ may be regarded either as *modus ponens* or as a simple map evaluation in the cartesion closed category (recall that $\{w\} \Rightarrow \emptyset$ is an exponential), and the reduction $\delta \cap (\{w\} \Rightarrow \emptyset)$ to δ is justified by the pre-condition pre– $Ent[w]\delta := \{w\} \cap \delta = \emptyset$.

4 Klinik of doctors and their patients

The usual model of doctors (DOC) and their patients (PAT) that we have become accustomed to use is that which associates with each doctor d in the klinik κ her/his set of current patients S. This model is captured by

$$\kappa \in KLINIK = DOC \longrightarrow \mathcal{P}PAT \tag{73}$$

and a typical klinik κ might have the form

$$\kappa = \begin{bmatrix} c \mapsto \{p, q, r\} \\ d \mapsto \{p, s\} \\ e \mapsto \emptyset \end{bmatrix} \tag{74}$$

It will be noticed that in this model the same patient p might be shared between two doctors c and d, and there is a doctor e with no patients.

This model of a klinik is the most general abstract model of the doctor-patient relation. It is a *directed* model in the sense that the relation is "the doctor d has the set of patients S".

From the perspective of the intuitionistc logic that we are developing it is clear that the codomain may be given the usual structure of a Heyting algebra.

Were one to exclude the possibility of null sets of patients, i.e., maplets of the form $d \mapsto \emptyset$ then one has the classical relational model of doctors and patients which we denote by

$$\kappa' \in KLINIK' = DOC \longrightarrow \mathcal{P}'PAT \tag{75}$$

where $\mathcal{P}'PAT = \mathcal{P}PAT \backslash \{\emptyset\}$.

Being a classical relation, models κ' are invertible. Thus we are led to introduce

$$\nu \in CLINIQUE = PAT \longrightarrow \mathcal{P}'DOC \tag{76}$$

where to each κ' in $KLINIK'$ there corresponds its inverse $(\kappa')^{-1} = \nu$.

Being accustomed to working with set-valued maps such as κ in the belief that these were the most interesting and practical models in practice we eschewed the more restricted model domains such as those of the form

$$\mu \in CLINIC = PAT \longrightarrow DOC \tag{77}$$

Our attention was drawn to their significance in a completely round-about manner. Specifically, in the abstract modelling of a hash table, we discovered that it might be cast completely in terms of a fibre bundle (Mac an Airchinnigh and Hughes 1997). This particular work was a very successful adventure into a geometry of formal methods. From fibre bundles we were led to the more general theory of sheaves and topoi. A good account of the relevance of such theories for our purposes may be gleaned from (Mac Lane and Moerdijk 1992).

It is clear to us that all these models of a klinik belong together. It is also clear that the natural framework is a topos. The basic recasting of all VDM map constructors and operators is the subject matter of a doctoral thesis just being completed (Hughes 2000) and we will report on this outcome at a later stage.

4.1 Klinik as fibred space

> " Logicians have long thought that the essence of existential quantification is projection; however, this is merely a special case of the *actual* essence, which is the taking of images. This is why we have adopted the notation $\exists_f(S) = f(S)$" (Lawvere 1975, 23).

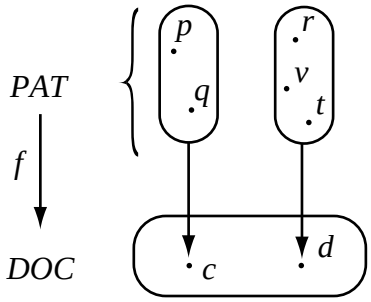
To complete this secion we now explain how quantifiers are introduced. In general, for a total map $f\colon X \longrightarrow T$, we may consider f as inducing structure on the domain X. In particular, for any t in the codomain T, the inverse image $f^{-1}(t)$ is called the fibre over t. (See Lawvere and Schanuel (1997, 81–5) for a brief account of the perspective that a map produces structure in its domain or in its codomain, depending upon the desired model.) Let us consider the model of doctors and patients given by the space of *total* maps

$$f \in CLINIC = PAT \longrightarrow DOC \tag{78}$$

subject to the constraint that f is surjective, i.e., that $\mathsf{rng}\, f = \mathsf{codom}\, f$. This condition will guarantee that no fibre is empty. In this highly desirable case one can then taken a (cross-)*section* through the fibres. Such sections provide further modelling concepts.

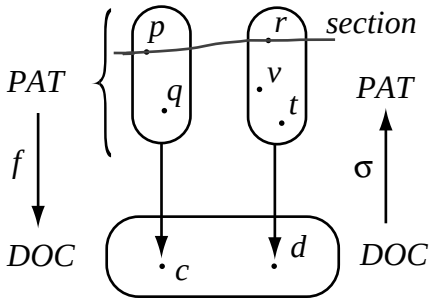
Consider the typical map

$$f = \begin{bmatrix} p \mapsto c \\ q \mapsto c \\ r \mapsto d \\ v \mapsto d \\ t \mapsto d \end{bmatrix}$$

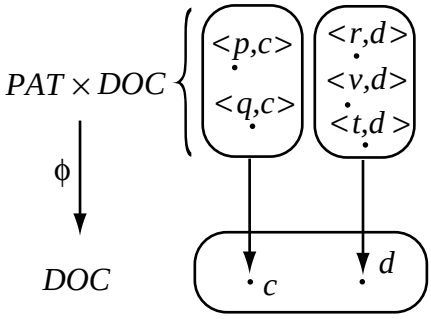


It may be represented as the fibred space shown where there are exactly two fibres each of which corresponds to a doctor. It is quite clear that such fibres capture a particular view of a doctor-patient relationship.

Now let us consider a section through the fibres. In general, a total map $f\colon X \longrightarrow T$ which is surjective has a section $s\colon T \longrightarrow X$ such that $s \circ f = 1_T$. A section may be considered to be a right-inverse for the map f. Shown here is a typical section σ through the given clinic map f. It is denoted $\sigma = [c \mapsto p, d \mapsto r]$. One interpretation of a section is the scheduling of doctors to patients concurrently within the same time period. There are clearly six possible schedules.

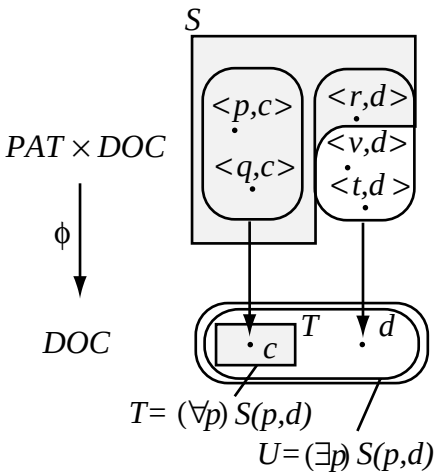


The fibring constructed above is not the only one. By considering the map f as a relation, i.e., a set of pairs of the form $\langle\, p, c \,\rangle$ each of which corresponds to $p \mapsto c$, one may produce an isomorphic fibring. Here the map ϕ is considered to extend f, where $\phi\langle\, p, c \,\rangle = f(p) = c$. It is this fibring which permits us to introduce universal and existential quantifiers as *constructions* into the VDM, following (Mac Lane and Moerdijk 1992, 57).



Consider the predicate $S(p, d)$ read "p is a patient of doctor d". Let $S \subseteq PAT \times DOC$ be the set of pairs $\langle\, p, d \,\rangle$ for which $S(p, q)$ is true.
Given S we **define** the universal quantifer $(\forall p)S(p, d)$ to be the subset $T \subseteq DOC$ which consists of all those d with $\langle\, p, d \,\rangle \in S$. The relationship between S and T is shown by the shaded areas of the diagram.
Similarly, **given** S we **define** the existential quantifier $(\exists p)S(p, d)$ to be the subset $U \subseteq DOC$ for which there exists a d with $\langle\, p, d \,\rangle \in S$. By construction, it is always the case that

$$(\forall p)S(p, d) \subseteq (\exists p)S(p, d).$$



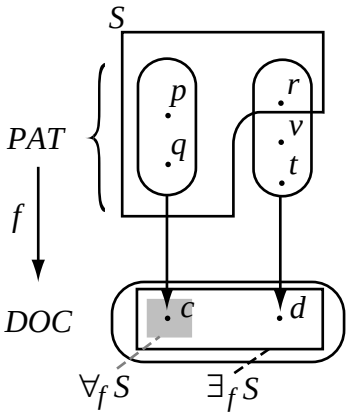
By the isomorphism observed above between the two different fibrings we can generalise the definitions of the universal and existential quantifiers to an arbitrary map f. Again, from (Mac Lane and Moerdijk 1992, 58), we have

$$\forall_f S := \{d \mid \text{for all } p, \text{ if } f(p) = d, \text{ then } p \in S\}$$

and

$$\exists_f S := \{d \mid \text{there exists a } p, \text{ with } f(p) = d, \text{ and } p \in S\}$$

We observe that $\forall_f S \subseteq \exists_f S$.



We have referred above to the definition of the quantifiers given by construction. We illustrate this for $\forall_f S$ and $\exists_f S$. Given f and S.

1. Compute the direct image $\exists_f S = f_*(S) = U$.
   [This is guaranteed to be constructive *in practice* since all our structures are finite. For model-theoretic methods such as VDM, Z, B, we may interchange $\mathsf{rng}\, f$, $f(S) = f_*(S)$, and $\exists_f$ where $S \subseteq \mathsf{dom}\, f$.]
2. Take the inverse image $X = f^*U = f^*f(S)$.
3. Since $S \subseteq f^*f(S)$ then let $Y = X \backslash S$.
4. Compute the direct image $f_*(Y)$.
5. Then $\forall_f S = \exists_f S \backslash f_*(Y)$.

Elimination of Y gives one (pleasing) form of the result:

$$\forall_f S = \exists_f S \backslash f_*(f^* \exists_f S \backslash S) \tag{79}$$

5 Epilogue

5.1 Related Work

In our School we have always taken the view that functions and maps are primary and that relations are secondary. This was and still is the primary focus. Consequently, the move to categories and topoi is straightforward. There is however a completely opposite well-known and well-established view that relations are primary. The categorical companion to the relation is an allegory. It is not surprising, therefore, to discover that what is here treated in terms of Heyting algebras and topoi is also covered within the chapter on relations and allegories by Bird and de Moor (1997, 81–110).

Similarly, in their work on the unified theories of programming, Hoare and He (1998, 86–112) have provided a chapter on linking theories wherein comparable material is handled in terms of lattices and Galois connections.

5.2 Acknowledgements

The members of the Foundations and Methods Group (FMG) at the Department of Computer Science in the University of Dublin, Trinity College, reviewed an earlier draft of this paper and made many valuable suggestions. Particular thanks are due to Andrew Butterfield for pointing out the usual need for a VDM invariant in the refined enter operation of the intuitionistic spelling checker dictionary to guarantee that the words in the dictionary $\{w\} \cup \delta$ correspond exactly to the set of atoms in $\sigma(\langle\, w \,\rangle \cdot \alpha)$. Any errors which remain in the paper are the sole responsibility of the author.

A particular debt of gratitude is owed especially to Arthur Hughes who helped me to struggle with what seemed to me to be very strange, difficult and irrelevant concepts (irrelevant that is from the perceived need of real computer scientists and software engineers) in both Category Theory and Topos Theory.

The paper was typeset on a Power Macintosh using BlueSky Textures 2.1.2 and the LaTeX2ϵ format. The style sheet used is Springer-Verlag's *llncs.cls* for the Lecture Notes in Computer Science. Vince Darley's BibTex 1.1.7 was used for the references.

References

[Barr and Wells (1995] Barr, M. and C. Wells (1995). *Category Theory for Computing Science* (Second ed.). London: Prentice Hall.

[Bird and de Moor (1997] Bird, R. and O. de Moor (1997). *Algebra of Programming*. London: Prentice Hall.

[Fitting (1969] Fitting, M. C. (1969). *Intuitionistic Logic, Model Theory and Forcing*. Studies in Logic and The Foundations of Mathematics. Amsterdam: North-Holland Publishing Company.

[Goguen (1999] Goguen, J. (1999). Tossing algebraic flowers down the great divide. In C. S. Calude (Ed.), *People & Ideas in Theoretical Computer Science*, pp. 93–129. Singapore: Springer-Verlag Singapore Pte. Ltd. [ISBN 981-4021-13-X].

[Hoare (1999] Hoare, C. A. R. (1999). Theories of Programming: Top-Down and Bottom-Up and Meeting in the Middle. In J. W. Jeannette Wing and J. Davies (Eds.), *FM'99 World Congress on Formal Methods*, Volume 1708 of *Lecture Notes in Computer Science*, pp. 1–27. Berlin: Springer-Verlag.

[Hoare and He (1998] Hoare, C. A. R. and J. He (1998). *Unifying Theories of Programming*. London: Prentice Hall. [ISBN 0-13-458761-8].

[Jones (1999] Jones, C. B. (1999). Scientific Decisions which Characterize VDM. In J. W. Jeannette Wing and J. Davies (Eds.), *FM'99 World Congress on Formal Methods*, Volume 1708 of *Lecture Notes in Computer Science*, pp. 28–47. Berlin: Springer-Verlag.

[Körner (1960] Körner, S. (1960). *The Philosophy of Mathematics, an Introductory Essay*. London: Hutchinson and Company, Limited. [ISBN 0-486-25048-2], The Dover edition is cited; Dover Publications, Inc., New York, 1986.

[Lambek and Scott (1986] Lambek, J. and P. J. Scott (1986). *Introduction to Higher Order Categorical Logic*. Cambridge: Cambridge University Press.

[Lawvere (1975] Lawvere, F. (1975). Variable sets etendu and variable structure in topoi. Technical report, University of Chicago. Notes by Steven Landsburg of Lectures and Conversations.

[Lawvere and Schanuel (1997] Lawvere, F. and S. Schanuel (1997). *Conceptual Mathematics, A first introduction to categories*. Cambridge: Cambridge University Press. [ISBN 0-521-47817-0].

NOTE: 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 (1990] 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 (1991] 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: Tu-*

torials, Lecture Notes in Computer Science ***552***, pp. 141–237. Berlin: Springer-Verlag.

[Mac an Airchinnigh and Hughes (1997] 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/.

[Mac Lane and Moerdijk (1992] Mac Lane, S. and I. Moerdijk (1992). *Sheaves in Geometry and Logic, A First Introduction to Topos Theory*. New York: Springer-Verlag. [ISBN 0-387-97710-4].

[McLarty (1992] McLarty, C. (1992). *Elementary Categories, Elementary Toposes*. Oxford: Clarendon Press. [ISBN 0 19 851473 5].

[Nordström, Petersson, and Smith (1990] Nordström, B., K. Petersson, and J. M. Smith (1990). *Programming in Martin-Löf's Type Theory, an Introduction*. Number 7 in The International Series of Monographs on Computer Science. Oxford: Clarendon Press. [ISBN 0-19-853814-6].

[Reilly (1995] Reilly, C. (1995). Exploring Specifications with Mathematica. In J. P. Bowen and M. G. Hinchey (Eds.), *ZUM'95: The Z Formal Specification Notation, Lecture Notes in Computer Science* ***967***, pp. 408–20. Berlin: Springer-Verlag.

[Selchow & Righter Company (1978] Selchow & Righter Company (1978). *The Official SCRABBLE® Players Dictionary*. Springfield, Massachusetts: Merriam-Webster Inc. [ISBN 0-87779-020-5].

[Stoll (1974] Stoll, R. R. ([1961] 1974). *Sets, Logic, and Axiomatic Theories* (Second ed.). San Francisco: W. H. Freeman and Company. [ISBN 0-7167-0457-9].

[Szabo (1978] Szabo, M. E. (1978). *Algebra of Proofs*. Number 88 in Studies in Logic and the Foundations of Mathematics. Amsterdam: North-Holland Publishing Company. [ISBN 0-7204-2286-8].

[Tyrrell, Butterfield, and Donnelly (2000] Tyrrell, M., A. Butterfield, and A. Donnelly (2000, January). Oo-motivated process algebra: A calculus for corba-like systems. In *To appear in the Third Workshop in Rigorous Object-Oriented Methods, York, England*.