The Logic Of A Tos

Ryan Shelswell and Michael Johnson
School of Mathematics, Physics, Computing and Electronics
Macquarie University
Australia

15th April, 1998

Abstract

In a categorical model of an information system, the internal logic of the category determines the query language of the information system. It has been argued that this logic should be determined by the categorical operations which can be performed within the system, rather than by post hoc addition of categorical structure. This paper presents the internal logic of the categorical model known as the tos.

1 Introduction

Information systems are of great interest to application developers and theoreticians alike due to their utility and elegance. Recently there have been efforts to produce categorical models of information systems [1,2,3]. A major component of any information system is the query language used for retrieving information from the system, and this has a deep correspondence with a particular aspect of a categorical model - the internal logic of the category. While some researchers have ignored this or simply imported sufficient categorical structure (typically that of a topos) to ensure the required logic is present, one of the authors has argued [1] that the categorical model should be developed operationally and the internal logic of the resulting category should be examined, perhaps obviating the need for the addition of extra structure. In [4] it was speculated that for the categorical model of a tos, this logic would be a restricted form of first-order intuitionistic predicate calculus. We now confirm this speculation and further argue that in typical application to information systems, the restrictions of the logic are not felt, so that such models effectively enjoy a standard first-order intuitionistic predicate calculus induced naturally from the operations of the categorical model.

The rest of this paper is organised as follows. Section 2 states the formal definition of a tos and provides some examples and basic results. Section 3 develops the construction of the various logical operations within a tos and shows how special kinds of toses are related to other well-known categories, and Section 4 concludes the paper, examining the implications of the results for the modelling of information systems.

2 The definition of a tos and basic results

A tos is defined to be a category which is lextensive, has a subobject classifier and a stable proper factorisation system.

Examples of toses include all topoi [7] and some logoi [8], although toses are distinct from both of these well-known categories. A specific example is the full subcategory of the topos $\mathbf{Set^2}$ which consists of finitely-fibred functions between sets. This category is a tos but not a topos.

Toses have been used to model information systems [1,5,6] and in fact this application has driven the definition of the category. Toses enjoy many nice technical properties relevant to information systems modelling, including the following: toses are regular categories, factorisation in a tos is canonical and is image factorisation, toses are closed under slicing, toses are not necessarily finitely cocomplete and toses are cartesian closed if and only if they are topoi.

3 The logic of a tos

3.1 The categorical interpretation of logic

The calculus of statements provided by logic can also be interpreted formally as an algebra of subsets. The basic idea is that a predicate is identified with a subset p of some set X. Then we can define logical operations as set-theoretic operations: for example, $p \wedge q$ is the set-theoretic intersection of p and q. If we want to model intuitionistic logic, we define a topology on our subsets and take just the open sets as predicates. We can then define all the propositional operators $\wedge$, $\vee$, $\rightarrow$ and $\neg$, as well as the quantifiers $\exists$ and $\forall$ in terms of set-theoretic operations. There is a well-known generalisation of this interpretation of logic into categorical terms but often the standard constructions [7,9] require more categorical structure than is present in a tos. In the developing the following results, we assume the reader is familiar with the standard constructions.
In particular we will assume the reader is familiar with the concepts of subobject and subobject classifier, which are used to generalise the notion of subset. In the following discussion, we will use the following (non-standard) definition of finiteness: an object B is called *finite* when the category $\mathbf{Sub}(\mathbf{B})$ of subobjects of B has a finite number of objects. Clearly in the case of **Set**, this notion corresponds to the standard set-theoretic notion of finiteness.

3.2 Propositional operators

We define our basic logical operators in the following way: we define a construction on subobjects which we will refer to as the *external* operation, a map into the codomain Ω of the subobject classifier which we will call the *internal* operation, and we show that pulling the internal operation back along any subobject results in a subobject isomorphic to that produced by the external operation.
We define the intersection operator $\wedge$ in the standard way as the pullback of subobjects p and q of an object B, and show that this is the greatest lower bound p and q. The union operator $\vee$ is also defined in the standard way, as the image of the coproduct, but in proving the isomorphism between the internal and external operations we need to use the fact that toses are locally distributive; this follows from lextensivity [10]. This definition of union makes $p \vee q$ the lowest upper bound of p and q.
Since for any pair of subobjects p and q of an object B in a tos we have defined a greatest lower bound and lowest upper bound, we have that $\mathbf{Sub}(\mathbf{B})$ is a lattice [11] for any object B in a tos. By defining $\bigwedge$ for any finite set of subobjects $P = \{p_1, p_2, ..., p_n\}$ as $\bigwedge P = p_1 \wedge p_2 \wedge ... \wedge p_n$ (and $\bigvee$ similarly), we have that $\mathbf{Sub}(\mathbf{B})$ is a finitely complete lattice.
The definition of the implication operator follows in the standard treatment from the cartesian closure of the base category; since toses are not necessarily cartesian closed we cannot define implication in this way. Instead we define internal implication $\rightarrow$ as the classifying map of the equaliser of internal intersection and the first projection and show that for any subobject p of an object B, $(p \rightarrow \_)$ is the right adjoint of $(p \wedge \_)$, and so the category $\mathbf{Sub}(\mathbf{B})$ is cartesian closed for any B.
The negation operator $\neg$ is defined in the standard way for any subobject p as $p \rightarrow 0$, where 0 is the bottom element of the lattice of subobjects $\mathbf{Sub}(\mathbf{B})$.
A partially ordered set is said to have a Heyting Algebra structure when it is a cartesian closed, finitely complete lattice [7]. The above definition of implication, together with union and intersection, show that for any object B in a tos, $\mathbf{Sub}(\mathbf{B})$ has a Heyting Algebra structure.

3.3 Quantification

In this section we recall some standard results and develop alternative constructions where necessary.
In a category with pullbacks, a morphism $f : A \rightarrow B$ induces a functor $f^* : \mathbf{C}/B \rightarrow \mathbf{C}/A$ called the change of base functor. If the category has a lattice of subobjects for each object, this change of base functor determines a functor $f^{-1} : \mathbf{Sub}(\mathbf{B}) \rightarrow \mathbf{Sub}(\mathbf{A})$ called the inverse image functor. So, a tos has an inverse image functor for each morphism $f : A \rightarrow B$.
It is not difficult to show that, in a tos, the Heyting Algebra structure of the subobject lattice $\mathbf{Sub}(\mathbf{B})$ of any object B is preserved by the inverse image functor associated with any morphism $f : A \rightarrow B$.
In a category with pullbacks, the change of base functor has a left adjoint denoted $\Sigma_f : \mathbf{C}/A \rightarrow \mathbf{C}/B$, which in turn determines a functor on the subobject lattices $\exists_f : \mathbf{Sub}(\mathbf{A}) \rightarrow \mathbf{Sub}(\mathbf{B})$ called the existential quantification of f. This functor is left adjoint to the inverse image functor.

In a category with pullbacks, if the category is cartesian closed then it is well-known that there exists a right adjoint to the change of base functor denoted $\Pi_f : \mathbf{C}/A \to \mathbf{C}/B$, which similarly determines a functor on subobjects $\forall_f : \mathbf{Sub(A)} \to \mathbf{Sub(B)}$ called the universal quantification of f. However, toses are not in general cartesian closed and furthermore the existence of Π_f forces a tos to be a topos.
To construct a universal quantification functor associated with a morphism $f : A \to B$ in a tos, we instead adapt another well-known construction used for example in [8]. We define universal quantification $\forall_f : \mathbf{Sub(A)} \to \mathbf{Sub(B)}$ in the following way. For any subobject a' of A, take the collection of subobjects b'_i of B for which $f^{-1}(b'_i) \leq a'$. Define $\forall_f(a') = \bigcup b'_i$. This map is order-preserving and so defines a functor between the subobject lattices which is the right adjoint of the inverse image functor $f^{-1} : \mathbf{Sub(B)} \to \mathbf{Sub(A)}$.
This universal quantification functor may not be definable in this way for every morphism $f : A \to B$ in a tos, because it requires the construction of a possibly infinite union of subobjects. Union of subobjects is defined in terms of coproduct, and while finite coproducts exist in a tos by virtue of lextensivity [10], infinite coproducts may not. We can characterise the restriction of universal quantification in a tos in the following way: if the codomain B of a morphism $f : A \to B$ is a finite object then the universal quantification functor $\forall_f$ is constructible in the way just described. If the codomain B is not a finite object then the above construction of $\forall_f$ does require an infinite union.
From these results we can investigate the relationship of a tos to another well-known category, a logos, defined to model first-order intuitionistic predicate calculus. As mentioned in Section 2, toses are not necessarily logoi, but it can be shown that a tos for which every object is finite is a logos, and therefore such a tos provides an internal logic equivalent to a standard first-order predicate calculus.

4 Conclusion

We have described the internal logic of a tos, a category used to model the operation of information systems. The categorical structure of a tos has resulted from examination of the operation of information systems; so the internal logic of the category exists naturally in the operational model, rather than by addition of categorical structure to ensure a certain logic is present.
The main result of the paper is that the logic of a tos which has only finite objects is a full first-order intuitionistic predicate calculus. Information systems can typically be modelled with such finite domains of variation and so the query language of information systems modelled in a tos is found to be equivalent to a standard first-order intuitionistic predicate calculus.

5 References

[1] C.N.G. Dampney, M. Johnson and G.P. Monro, An illustrated mathematical foundation for ERA, in C.M.I. Rattray and R.G. Clarke (eds) *The unified computation laboratory*, 77-84. Oxford University Press (1992).

[2] R. Rosebrugh and R.J. Wood, Relational databases and indexed categories, in *Canadian Mathematical Society Conference Proceedings*, **13** 391-407 (1992).

[3] A. Islam and W. Phoa, Categorical models of relational databases I: Fibrational formulation, schema integration, preprint (1993).

[4] M. Johnson, A categorical logic for information systems, *Journal of the Interest Group in Pure and Applied Logic*, **4** 502-504. Oxford University Press (1996).

[5] M. Johnson, R. Rosebrugh and R.J. Wood, Entity-Relationship Models and Sketches, preprint (1996).

[6] M.Johnson and C.N.G. Dampney, On the value of commutative diagrams in information modelling, *Springer Workshops in Computing*, 47-60 (1994).

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

[8] P.J. Freyd and A. Scedrov, *Categories, Allegories.* Amsterdam: Elsevier Science Publishers B.V. (1990).

[9] F. Borceux, Fasci, Logica e topoi, *Quderni dell'Unione Matematica Italiana*, **34**. Bologna: Pitagora Editrice (1989).

[10] A. Carboni, S. Lack and R.F.C. Walters, Introduction to Extensive and Distributive Categories, *Journal of Pure and Applied Algebra*, **84** 145-158.

[11] B.A. Davey and H.A. Priestly, *Introduction to Lattices and Order.* Cambridge University Press (1990).