2011 International Symposium on Nonlinear Theory and its Applications
NOLTA2011, Kobe, Japan, September 4-7, 2011

Sheaf-theoretic structural analysis of computational networks

Michael Robinson†

†Mathematics Department
University of Pennsylvania
209 S. 33rd Street
Philadelphia, PA 19104
Email: robim@math.upenn.edu

Abstract—
We show how logic circuits can be encoded into a cellular sheaf, and how traditional sheaf theoretic invariants can be interpreted semantically. In this setting, there exists strictly more information available to a circuit designer than in static truth tables, but less than in event-level simulation. This information is related to the timing behavior of the logic circuits, and thereby provides a "bridge" between static logic analysis and detailed simulation. Future directions are also outlined.

1. Introduction

Verification of asynchronous computational networks usually requires extensive simulation and appropriate test coverage. This article presents a novel technique for detecting certain behavioral properties of a network using a less exhaustive structural analysis. In this analysis, delays are unknown and finite, but unlike the work of others in this situation, the delays are implicit. We do not need to assume that they have a fixed value over time, and we never specify them even as variables. We show how potentially hazardous race conditions (which often cause transients or unwanted latching) correspond to nontrivial first cohomology classes of a particular sheaf that encodes this implicit timing model of the network.

1.1. Historical perspective

The design of asynchronous networks is an old subject, having been of interest at least into the early 1950s during the development of asynchronous controllers for digital computers. [9] Although the benefits of using asynchronous hardware instead of synchronous hardware are substantial (better composibility of modules, lower power usage, lower electromagnetic interference, faster speed), its challenges have generally precluded its widespread acceptance. Most of the difficulty of asynchronous design involves careful control of delays within the circuit, and the avoidance of race conditions. [11] Asynchronous software has recently become more important as multicore processors become the norm. [17]

The challenges of asynchronous design in both hardware and software revolve around timing instabilities and sensitivities, which usually mean that verification requires exhaustive simulation. As a result of both the benefits and the challenges, a lively literature has grown up around asynchronous design and verification. There are essentially three major threads of inquiry:

1. Specification of a semantic or behavioral model of the network, [20, 12, 11, 14, 7]

2. Synthesis of the network from this specification, [5, 4, 15, 18] and

3. Simulation of the network to verify its correct performance. [2, 3, 10, 13, 22]

Although there are numerous design rules and specification languages that have been devised for reliable operation of asynchronous networks (for instance [6] [16] [8]), these cut a narrow path through the design space. If a network that does not follow these design rules is to be analyzed or verified, one needs to perform essentially exhaustive analysis. We submit that in this situation, there is another option: that of a coarser structural analysis of the network. One should look for invariants of the network that are robust to changes in the network that preserve its semantics. Such an anaylsis cannot completely characterize network semantics, but should be able to detect behaviors of interest.

2. A structural approach

Our approach is motivated by the insight that local network analysis is often easy, even in a fully asynchronous setting. We would like our analysis to have the following three properties:

1. That it collates the local analysis into a global picture of the semanics,

2. That it facilitates basic network transformations, such as joining two subnetworks or ignoring a subnetwork, and

3. That it manages uncertainty as a proxy for an exhaustive temporal analysis.

As it happens, these requirements are naturally met by a family of mathematical tools called sheaf theory. Specifically, *cohomology* facilitates local-to-global inferences, the *Mayer-Vietoris* principle relates subcircuits to their union, and a particular *categorification* permits metered amounts of uncertainty to be examined.

2.1. Cellular sheaves

A sheaf is a mathematical tool for storing local information over a domain. We are interested in cellular sheaves [21], which assign some algebraic object to each cell of a cellular space. Global information is extracted from this structure using compatability conditions of two kinds: (1) those that pertain to restricting the information from a lower- to a higher-dimensional cell, and (2) those that pertain to assembling information on small sets of cells into information on larger ones. What is of particular interest is the relationship of the global information, which is valid over the entire space, to the topology of that space. This is captured by the cohomology of the sheaf, in the way we summarize here.

Consider on X, an oriented CW complex, the *face category* in which the objects are the cells of X and the morphisms $c \to d$ connect a cell d to another cell c in its boundary. A *cellular sheaf* $\mathcal{F}$ taking values in a category C is a covariant functor from the face category of X to C. Specifically, $\mathcal{F}$ assigns an object $\mathcal{F}(c)$ of C to each cell c of X, and a morphism ρ_c^d in C for each cell-boundary pair $c \in \partial d$. Borrowing from standard sheaf theory terminology [1], the objects $\mathcal{F}(c)$ are called *stalks* and the maps ρ_c^d are called *corestriction maps*. Likewise, we define a *section* of $\mathcal{F}$ to be a function f on the cells of X that satisfies $f(c) \in \mathcal{F}(c)$, and $f(c) \in (\rho_c^d)^{-1}(f(d))$. If we relax the latter condition, f is called a *serration*.

When a cellular sheaf takes values in a category of abelian groups, we can define its *cohomology*, which is a disciplined way of extracting global information from the local information stored in the sheaf. Define the *cellular cochains* for a cellular sheaf $\mathcal{F}$ to be the groups $C^k(X;\mathcal{F}) = \prod_{\dim c = k} \mathcal{F}(c)$, where the c range over all k-dimensional cells of X. The cellular cochains form a chain complex with the coboundary map $d^k : C^k(X;\mathcal{F}) \to C^{k+1}(X;\mathcal{F})$ coming from the corestrictions in the following way:

$$d^k(f)(c) = \sum_{\substack{d \in \partial c, \\ \dim c = \dim d + 1}} [c:d] \rho_c^d(f(c)), \tag{1}$$

where f is a serration of $\mathcal{F}$ and $[c:d] = 1$ if the orientations of c and d agree and -1 otherwise. The usual algebraic manipulation shows that $d^{k+1}d^k = 0$. If f is a section (not just a serration), then $d^k(f) = 0$, so we define the *cohomology* of $\mathcal{F}$ to be

$$H^k(X;\mathcal{F}) = \ker\ d^k / \text{image } \mathrm{d}^{k-1}. \tag{2}$$

$H^0(X;\mathcal{F})$ can be shown to be isomorphic to the group of sections of $\mathcal{F}$, and $H^k(X;\mathcal{F}) = 0$ when $k > \dim X$.

3. Switching sheaves

Suppose that X is an oriented CW complex. If c, d are cells of X such that $c \in \partial d$ and $\dim\ d = \dim\ c + 1$, then c is called a *face* of d. We call d a *coface* of c. If c and d agree about orientation, we call them *co-oriented*.

Suppose Q is a cellular sheaf on an oriented cell complex X. We will say that Q is a *quiescent switching sheaf* based on a set A if for all cells $c \in X$,

1. the stalk $Q(c) = A$ if c has no co-oriented cofaces,
2. otherwise, the stalk $Q(c) = \prod_d Q(d)$ where d ranges over the co-oriented cofaces of c, and
3. the corestriction $Q(c) \to Q(e)$ function is the projection onto the factor in the product $Q(c) = \prod_d Q(d)$ associated to e.

Unfortunately, quiescent switching sheaves are not generally sheaves of abelian groups. As a result, we cannot compute their cohomology. We correct for this problem by encoding the values of A in an $\mathbb{F}$-vector space. Consider the function $T : A \to \mathbb{F} \otimes A$, given by the inclusion $x \mapsto 1 \otimes x$. This T lifts the corestrictions to $\mathbb{F}$-linear maps. Applying this idea in our definition of quiescent switching sheaves corresponds to a particular *categorification*.

Casual examination suggests that very little has changed, except the algebraic structure has been slightly enhanced. However, two new things have occured:

1. Problems of logic can now be addressed computationally using the framework of linear algebra. This can result in gains in asymptotic computational complexity. Rather than being forced to enumerate states over the set A, one may instead perform standard polynomial-time linear algebra (over the vector space $\mathbb{F} \otimes A$).
2. It is possible to superpose two logic states, and thereby study certain kinds of transitions between logic states. This is subtle and somewhat surprising: we have not explicitly described anything about time evolution of circuits, and indeed the usual way of examining the states of a logic circuit does not concern itself with time. However, by permitting superposed states, we are able to study the circuit's response to both *simultaneously* and thereby discern the way that one might transition to the other. Any section of a switching sheaf that vanishes anywhere must be the linear superposition of two or more sections of the underlying quiescent switching sheaf, and therefore describes uncertainty or transient states.

We therefore define a *switching sheaf* to be a cellular sheaf that is the categorification via T of a quiescent switching sheaf. This essentially amounts to rewriting the definition of quiescent switching sheaves by replacing the Cartesian product with a tensor product, and the corestriction maps become contractions instead of projections.

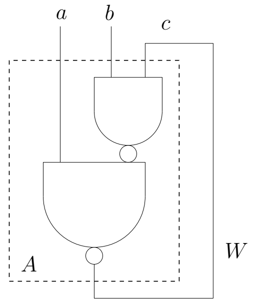

Figure 1: An R-S flip-flop circuit

3.1. An example

Consider the circuit X shown in Figure 1, which is a basic memory element. We split it into two pieces: a combinational circuit A with a 3-input gate, and a feedback wire W. The logic states for this circuit [20] are summarized in the following table:

a	b	c	q	Description
0	0	1	1	Danger
0	1	1	1	Set
1	0	0	0	Reset
1	1	0	0	Hold zero
1	1	1	1	Hold one

We construct a 1-dimensional cellular space that represents the connections in the circuit, called the connection graph. This particular directed graph X has one vertex, representing the three-input gate, one directed self-loop, and two additional directed edges incident on the vertex and directed toward it. On this directed graph, we construct a quiescent switching sheaf Q based on the finite set $\{0,1\}$. From the definition, it is clear that the stalk over each edge is the set $\{0,1\}$ and the stalk over the vertex is the set of ordered binary triples.

The categorification $T : \{0,1\} \to \mathbb{F}_2 \otimes \{0,1\}$ of Q yields a switching sheaf $\mathcal{S}$. Since the stalk of $\mathcal{S}$ over the vertex is the tensor product of 3 copies of $\{0,1\}$, it is an $\mathbb{F}_2$-vector space of dimension 8. The corestrictions from the vertex to each input edge are given by the following matrices:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \tag{3}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{4}$$

and

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{5}$$

The corestriction from the vertex to its output edge (which is a self loop) is given by

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

Computation of the cohomology shows that $H^0(X;\mathcal{S})$ has dimension 7 and $H^1(X;\mathcal{S})$ has dimension 1. Here is a basis for $H^0(X;\mathcal{S})$:

Element of $H^0(X;\mathcal{S})$	Description
$\overline{a} \otimes \overline{b} \otimes c$	Danger
$\overline{a} \otimes b \otimes c$	Set
$a \otimes \overline{b} \otimes \overline{c}$	Reset
$a \otimes b \otimes \overline{c}$	Hold zero
$a \otimes b \otimes c$	Hold one
$\overline{a} \otimes \overline{b} \otimes \overline{c} + a \otimes \overline{b} \otimes c$	Transition Danger to Reset
$\overline{a} \otimes \overline{b} \otimes \overline{c} + \overline{a} \otimes b \otimes \overline{c}$	Transition Danger to Set

Of most interest are the last two basis elements. These are linear combinations of two terms, neither of which is a T-lift of a section of Q. The most suggestive interpretation is that they imply an uncertainty when exiting the Danger state. As the inputs a and b transition from both logic 0 to both logic 1, there is a race condition. Only one of them transitions first, so there is a brief transition into the Set or Reset states before entering a Hold state. If we add the last two basis elements, we obtain $a \otimes \overline{b} \otimes c + \overline{a} \otimes b \otimes \overline{c}$ which indicates that an uncertainty about which of a or b transitions has occured results in uncertainty in the signal c.

3.2. Invariants and semantics

It is clear from the example that switching sheaves contain strictly more information than simply the logic states of a circuit, which are encoded as sections of Q. Using the Mayer-Vietoris principle [1] for cellular sheaves, it is possible to relate the cohomology of a circuit composed of simpler circuits to the cohomology of each of these subcircuits.

Theorem: [19] The effect of attaching a wire is best described by the following slogan:

- Attaching a wire that does not participate in feedback suppresses logic states and leaves H^1 unchanged.
- Attaching a wire that participates in feedback leaves logic states unchanged and adds to the dimension of H^1.

As an immediate corollary, *nontrivial H^1 of a switching sheaf detects race conditions.*

4. Future work

The most evident area for development involves the interpretation of higher dimensional switching sheaves and their associated cohomology. It seems likely that a thorough analysis of the higher cohomology will permit semantic *comparison* between circuits, much as cartesian products of finite state machines are crucial in their comparison. To this end, it will be interesting to determine the relationship between semantic equivalence and switching sheaves with isomorphic cohomology. It is likely that the two concepts are intimately connected, but that one does not imply the other.

It also remains to quantify precisely what features of circuit behavior are captured by categorification. For instance, it is immediate that complicated sequential semantics will not be captured by the categorified model. However, it is possible that a filtration structure may permit both the cohomological outlook as well as the manipulation of timeseries. In doing so, we will likely figure out exactly how glitch and hazard transitions of logic circuits are represented (or not) by the cohomology of switching sheaves.

Finally, it is straightforward to compute the cohomology of a switching sheaf automatically in software. Our initial experiments with this software model indicate that there is a delicate relationship between the combinatorial structure of a cellular space and the cohomology of switching sheaves over that space.

Acknowledgments

This work was supported under AFOSR FA9550-09-1-0643. The author would additionally like to thank Dr. Yasu Hiraoka for insightful discussion about this work and the invitation to the NOLTA conference.

References

[1] Glen Bredon. *Sheaf theory*. Springer, 1997.

[2] Michael Browne, Edmund Clarke, David Dill, and Bud Mishra. Automatic verification of sequential circuits using temporal logic. *IEEE Transactions on Computers*, C-35(12), December 1986.

[3] E. M. Clarke and O. Grümberg. Avoiding the state explosion problem in temporal logic model checking algorithms. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 294–303, 1987.

[4] David Cox. Complete synthesis method for d flip-flops with set and reset inputs. In *9th NASA Symposium on VLSI Design*, November 2000.

[5] Al Davis and Steven Nowick. An introduction to asynchronous circuit design. Technical Report UUCS-97-013, University of Utah Computer Science Department, September 1997.

[6] Jordi Cortadella *et al.* *Logic synthesis for asynchronous controllers and interfaces*. Springer, 2002.

[7] John Harrison. Formal proof - theory and practice. *Notices of the AMS*, 55(11):1395–1406, December 2008.

[8] C. A. R. Hoare. *Communicating Sequential Processes*. 2004.

[9] D. A. Huffman. The synthesis of sequential switching circuits. Technical Report 274, Research Laboratory of Electronics, MIT, 1954.

[10] Nagisa Ishiura, Mizuki Takahashi, and Shuzo Yajima. Time-symbolic simulation for accurate timing verification of asynchronous behavior of logic circuits. In *26th ACM/IEEE Design Automation Conference*, 1989.

[11] Rajit Manohar and Alain Martin. Quasi-delay-insensitive circuits are turing-complete. In *Second International Symposium on Advanced Research in Asynchronous Circuits and Systems*, 1995.

[12] Alain Martin. The limitations to delay-insensitivity in asynchronous circuits. Technical Report Caltech-CS-TR-90-02, Computer Science Department, California Institute of Technology, 1990.

[13] Gerd Meister. A survey on parallel logic simulation. Technical report, University of Saarland, Department of Computer Science, 1993.

[14] L. G. Meredith and D. Snyder. Knots as processes. In *Traced Monoidal Categories Workshop*, 2007.

[15] Yannick Monnet, Marc Renaudin, and Régis Leveugle. Designing resistant circuits against malicious faults injection using asynchronous logic. *IEEE Transactions on Computers*, 55(9), September 2006.

[16] Joachim Parrow. An introduction to the π-calculus. In Bergstra *et al.*, editor, *Handbook of Process algebra*. Elsevier, 2001.

[17] D. Patterson. The trouble with multicore. *IEEE Spectrum*, July 2010.

[18] Dumitru Potop-Butucaru. Correct-by-construction asynchronous implementation of modular synchronous specifications. *Fundamenta Informaticae*, 78:131159, 2007.

[19] M. Robinson. Asynchronous logic circuits and sheaf obstructions, `arxiv:1008.2729` (submitted). In *GETCO*, 2010.

[20] Claude Shannon. A symbolic analysis of relay and switching circuits. Master's thesis, MIT, 1940.

[21] A. Shepard. *A cellular description of the derived category of a stratified space*. PhD thesis, Brown University, 1985.

[22] Vida Vakilotojar. *Induced Hierarchical verification of asynchronous circuits using a partial order technique*. PhD thesis, University of Southern California, 2000.