Algebras for Hazard Detection

Janusz Brzozowski[1], Zoltán Ésik[2], and Yaacov Iland[1]

[1] Department of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
{brzozo,yiland}@uwaterloo.ca
http://maveric.uwaterloo.ca

[2] Department of Computer Science, University of Szeged, Árpád tér 2, 6720 Szeged, Hungary
esik@inf.u-szeged.hu
http://www.inf.u-szeged.hu/~esik

Abstract. Hazards pulses are undesirable short pulses caused by stray delays in digital circuits. Such pulses not only may cause errors in the circuit operation, but also consume energy, and add to the computation time. It is therefore very important to detect hazards in circuit designs. Two-valued Boolean algebra, which is commonly used for the analysis and synthesis of digital circuits, cannot detect hazard conditions directly. To overcome this limitation several multi-valued algebras have been proposed for hazard detection. This paper surveys these algebras, and studies their mathematical properties. Also, some recent results unifying most of the multi-valued algebras presented in the literature are described. Our attention in this paper is restricted to the study of static and dynamic hazards in gate circuits.

1 Introduction

The two-element Boolean algebra, $\mathbf{A_2}$, has been the standard algebra for circuit analysis and design, since Shannon's pioneering work in 1938 [40]. The problem of hazards has been recognized very early; hazards were already discussed in 1951 in a book by Keister, Ritchie and Washburn [28]. In 1957 Huffman [27] proposed informal definitions of static and dynamic hazards, and provided some characterizations of these hazards. Roughly speaking, a static hazard is one or more pulses occurring in a signal which should be constant, and a dynamic hazard is one or more pulses in a signal that is changing from one binary value to the other. A more formal treatment of hazards was given by McCluskey [33] in 1962, and further results were obtained by Unger [42] in 1969. These early works used the two-valued Boolean algebra for gate circuits.

Although the above-mentioned binary methods were successfully applied to a number of problems related to hazards, they are indirect, and require the use of such tools as Karnaugh maps, or detailed knowledge of the circuit structure including wire delays. Consequently, researchers turned their attention to algebras using more than two values, with the hope of finding a more direct method for hazard detection.

The first nonbinary hazard algebra was the three-valued ternary algebra, which we call $\mathbf{A_3}$, introduced by Goto [22] in 1948. In 1953 a four-valued algebra, $\mathbf{A_4}$, was described by Metze [34]. A five-valued algebra, $\mathbf{A_5}$, was presented by Lewis [32] in 1972. A six-valued algebra, $\mathbf{A_6}$, was derived by Hayes [25] in 1986 for the detection of static hazards. An eight-valued algebra, $\mathbf{A_8}$, is implicit in the 1974 work of Breuer and Harrison [5]. The same algebra is also implicit in the 1974 work of Fantauzzi [20], who presented a nine-valued algebra, $\mathbf{A_9}$. In 1986 Hayes [25] studied the then-known multi-valued algebras. He presented general methods for constructing new algebras. Using these methods he obtained a thirteen-valued algebra, $\mathbf{A_{13}}$. Breuer and Harrison [5] proposed a 27-valued algebra, $\mathbf{A_{27}}$, for eliminating hazards in test generation.

In several papers mentioned above, little attention has been paid to the mathematical structure of hazard algebras. This paper summarizes the mathematical properties of hazard algebras, and examines their completeness and usefulness for hazard detection.

Some very recent results by Brzozowski and Ésik [8, 9] generalize and unify most of the previous work on hazard algebras. An infinite algebra, $\mathbf{C}$, and an infinite number of finite algebras, $\mathbf{C_k}$, were introduced for counting signal transitions and hazard pulses. A characterization of the results of simulation in algebra $\mathbf{C}$ for a restricted class of feedback-free circuits has been given by Gheorghiu [21].

The remainder of the paper is structured as follows. Section 2 introduces the algebraic laws which occur in many of the hazard algebras. Sections 3–10 examine the mathematical and hazard-detecting properties of the hazard algebras with three, four, five, six, eight, nine, thirteen and 27 elements. Section 11 summarizes the recent work by Brzozowski and Ésik on the infinite hazard algebra, and its application to the classification of hazard algebras.

2 Laws of hazard algebras

Logic circuits are often viewed as being constructed with OR gates, AND gates and inverters. Thus, algebras for such circuits naturally have three operations, $+, *$, and $^-$, corresponding to these gates. The normal logic values are denoted by 0 and 1, and an unknown value (if present in the algebra) is represented by Φ. The values 0, 1, and Φ are constants in the algebras. For these reasons, we are dealing with general algebraic systems having the form $P = \langle \mathcal{A}, +, *, ^-, 0, 1 \rangle$, where $\mathcal{A}$ is a set of elements, $+$ and $*$ are binary operations on $\mathcal{A}$, $^-$ is a unary operation on $\mathcal{A}$, and 0 and 1 are constants in $\mathcal{A}$. We take the liberty of considering Φ as a constant of the algebra whenever appropriate.

Various subsets of the following set of laws have been used to describe algebraic structures defined for the study of hazard algebras. We first recall the definitions of several algebraic systems. A *semigroup* is a system $S = \langle \mathcal{A}, + \rangle$

Table 1. Laws pertaining to hazard algebras

For all $a, b, c \in \mathcal{A}$:

Idempotence	L1 $a + a = a$	L1′ $a * a = a$
Commutativity	L2 $a + b = b + a$	L2′ $a * b = b * a$
Associativity	L3 $a + (b + c) = (a + b) + c$	L3′ $a * (b * c) = (a * b) * c$
Absorption	L4 $a + (a * b) = a$	L4′ $a * (a + b) = a$
Identity	L5 $a + 0 = a$	L5′ $a * 1 = a$
Bounding	L6 $a + 1 = 1$	L6′ $a * 0 = 0$
Distributivity	L7 $a + (b * c) = (a + b) * (a + c)$	L7′ $a * (b + c) = (a * b) + (a * c)$
Involution	L8 $\overline{(\overline{a})} = a$	
De Morgan's laws	L9 $\overline{(a + b)} = \overline{a} * \overline{b}$	L9′ $\overline{(a * b)} = \overline{a} + \overline{b}$
Complement laws	L10 $a + \overline{a} = 1$	L10′ $a * \overline{a} = 0$
Ternary laws	L11 $(a + \overline{a}) + \Phi = a + \overline{a}$	L11′ $(a * \overline{a}) * \Phi = a * \overline{a}$
Self-complement	L12 $\overline{\Phi} = \Phi$	

satisfying L3. A *bisemigroup* [8,9] is a pair of semigroups, $S_+ = \langle \mathcal{A}, + \rangle$ and $S_* = \langle \mathcal{A}, * \rangle$, with the same underlying set. A bisemigroup is *commutative* if both of its operations are commutative, *i.e.,* if it satisfies L2 and L2′. A bisemigroup is *bounded* if it has two constants 0 and 1 satisfying L5, L5′, L6, and L6′. A commutative bisemigroup is *de Morgan* [8,9,19] if it is bounded and has a unary operation $^-$ satisfying L8, L9 and L9′. A *semilattice* is a semigroup satisfying L1 and L2. A *bisemilattice* [6] is a pair of semilattices $S_+ = \langle \mathcal{A}, + \rangle$ and $S_* = \langle \mathcal{A}, * \rangle$ which have the same underlying set. A bisemilattice is *de Morgan* [6] if it is bounded and satisfies L8, L9 and L9′. A *lattice* is a bisemilattice satisfying L4 and L4′. A lattice is said to be *bounded* if L5, L5′, L6 and L6′ hold. A lattice is said to be *distributive* if L7 and L7′ hold. A *de Morgan algebra* is a bounded, distributive lattice equipped with a $^-$ operation satisfying L8, L9 and L9′. A de Morgan algebra is a *Boolean algebra* if L10 and L10′ hold. A de Morgan algebra is a *ternary algebra* [13] if it contains Φ, and L11, L11′ and L12 hold.

A binary relation on $\mathcal{A}$ which is reflexive, antisymetric and transitive is called a partial order on $\mathcal{A}$. For $a, b, c \in \mathcal{A}$, c is the least upper bound (*lub*) of a and b if $a \le c$, $b \le c$, and for any d such that $a \le d$ and $b \le d$, we have $c \le d$. If the *lub* exists for every pair of elements in a partially ordered set (poset), let $a + b = lub\{a, b\}$. Then $\langle \mathcal{A}, + \rangle$ is a semilattice. Conversely, given a semilattice operation $+$, define $a \le b$ iff $a + b = b$. Then $\le$ is a partial order such that $a + b$ is in fact the least upper bound of a and b, for any $a, b \in \mathcal{A}$. A Hasse diagram representing such a partial order is then a convenient way of defining the operation $+$.

In a bisemilattice [6] $\langle \mathcal{A}, +, * \rangle$ there are two partial orders:

$$a \le b \text{ iff } a + b = b,$$

and

$$a \sqsubseteq b \text{ iff } a * b = a.$$

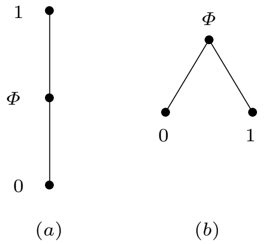

Fig. 1. Partial orders for $\mathbf{A_3}$: (a) $\leq$ and $\sqsubseteq$; (b) uncertainty order

In case the bisemilattice is a lattice, these partial orders coincide.

3 The three-valued algebra

Three-valued algebra has been introduced by Goto [22, 23] (1948). Other early work using three-valued algebra for hazard detection includes the papers by Moisil [35] (1956), Roginskii [38] (1959), Muller [36] (1959), Yoeli and Rinon [43] (1964), and Eichelberger [17] (1965).

In three-valued algebra, 0 and 1 are logic 0 and logic 1 respectively, while Φ represents an unknown value. The operations of the three-valued algebra are shown in Table 2. It is easily verified that the three-valued algebra is a ternary algebra (defined above). It is not a Boolean algebra, since $\Phi + \overline{\Phi} = \Phi + \Phi = \Phi \neq 1$, which violates the complement law. The partial order corresponding to the $+$ and $*$ operations in the three-valued algebra is given in Fig. 1(a). We can read off the result of either binary operation from this figure. For example, $\Phi + 1 = 1$ since the least upper bound of $\{\Phi, 1\}$ is 1, and $\Phi * 0 = 0$ since the greatest lower bound of $\{0, \Phi\}$ is 0.

Table 2. Operations of three-valued algebra

$+$	0	Φ	1
0	0	Φ	1
Φ	Φ	Φ	1
1	1	1	1

$*$	0	Φ	1
0	0	0	0
Φ	0	Φ	Φ
1	0	Φ	1

a	$\overline{a}$
1	0
Φ	Φ
0	1

At this point it is convenient to describe some of the results in the 1986 paper of Hayes [25], in which the author surveys the algebras that have been used for hazard detection and other simulation tasks. He also clarifies a number of issues, introduces a unified notation, and describes two methods for generating new hazard algebras from smaller ones. Here we digress to describe the first method of Hayes, which sheds further light on algebra $\mathbf{A_3}$.

The first method constructs a new algebra $\mathbf{A}'$ from a given algebra $\mathbf{A}$. The underlying set $\mathcal{A}'$ of $\mathbf{A}'$ is a subset of the power set of the underlying set $\mathcal{A}$ of $\mathbf{A}$. The operations are extended from $\mathbf{A}$ to $\mathbf{A}'$ as follows. If $X_1, \ldots, X_n$ are in $2^{\mathcal{A}}$ and $f(x_1, \ldots, x_n)$ is an n-ary operation in $\mathbf{A}$, then f is extended to $\mathbf{A}'$ by defining $f(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) \mid x_i \in X_i, i = 1, \ldots, n\}$.

For example, let the given algebra be $\mathbf{A_2}$. We can choose the following subset of $2^{\{0,1\}}$: $\{\{0\}, \{1\}, \{0, 1\}\}$ as the underlying set of a new algebra, $\mathbf{A_2'}$. Hayes observes that $\mathbf{A_2'}$ is isomorphic to $\mathbf{A_3}$, as the reader can easily verify. The correspondence is $\{0\} \mapsto 0$, $\{1\} \mapsto 1$, and $\{0, 1\} \mapsto \mathit{\Phi}$.

Eichelberger [17] introduces an efficient method for hazard and race detection. His ternary simulation consists of two algorithms, A and B, which use the uncertainty partial order shown in Fig. 1(b). This partial order puts values with more uncertainty higher in the order. We want to analyze what happens when some inputs change from their initial binary values to their final binary values. We first apply Algorithm A, in which the changing inputs are set to the uncertain value $\mathit{\Phi}$. We then simulate the circuit using the three-valued algebra to determine whether the uncertainty introduced by the changing inputs can spread to other gates. All unstable variables are changed at the same time. It can be shown that in Algorithm A a variable can only change from a binary value to $\mathit{\Phi}$, or it can remain unchanged. Thus, Algorithm A stops in at most n steps, if there are n gates in the circuit.

Algorithm B uses the output of Algorithm A as the initial state, and now simulates the circuit using the final values of the changing inputs. This time a variable can only change from $\mathit{\Phi}$ to a binary value, or it can remain constant. Algorithm B also terminates in at most n steps.

A static hazard is detected if an output has the same value in the initial state of Algorithm A and in the final state of Algorithm B, but has value $\mathit{\Phi}$ in the final state of Algorithm A. Ternary simulation cannot detect dynamic hazards, but it can detect oscillations in sequential circuits. The method is described in detail in [13].

Ternary simulation was shown to be correct with respect to binary analysis by Brzozowski and Seger [11–13]. A more general version, which does not require that the initial state of the circuit be stable, was described in [39]. Brzozowski, Lou and Negulescu [10] gave a set-theoretic characterization of finite ternary algebras, and this result was generalized to the infinite case by Esik [18]. Free ternary algebras were recently characterized by Balbes [2].

The three-valued algebra has been used in circuit simulators, for example, in the TEGAS simulator of Thompson and Szygenda [41]. Three-valued simulation is also described in a 1976 book by Breuer and Friedman [4].

Summary: Simulation in the three-valued algebra, $\mathbf{A_3}$, is an efficient method for detecting static hazards, but cannot handle dynamic hazards. Its correctness with respect to binary analysis has been proven. Algebra $\mathbf{A_3}$ is the smallest example of ternary algebra; the mathematical properties of ternary algebra are well understood. We return to $\mathbf{A_3}$ in Section 11.

4 The four-valued algebra

The four-valued algebra of Metze has the form $P = \langle \mathcal{A}, +, *, ^-, 0, 1\rangle$, where $\mathcal{A}$ = {0, 1/0, 0/1, 1}, + and $*$ are defined in Table 3, and $\{0, 1\}$ and $\{0/1, 1/0\}$ are complementary pairs. The values 0 and 1 are the usual logic values, and 0/1 and 1/0 represent transitions from 0 to 1 and 1 to 0, respectively. The same partial order corresponds to both operations + and $*$; it is in fact a total order: $0 \leq 1/0 \leq 0/1 \leq 1$.

Table 3. Operations + and $*$ in $\mathbf{A_4}$

+	0	1/0	0/1	1
0	0	1/0	0/1	1
1/0	1/0	1/0	0/1	1
0/1	0/1	0/1	0/1	1
1	1	1	1	1

$*$	0	1/0	0/1	1
0	0	0	0	0
1/0	0	1/0	1/0	1/0
0/1	0	1/0	0/1	0/1
1	0	1/0	0/1	1

Metze observes that addition and multiplication in $\mathbf{A_4}$ are idempotent, commutative and associative, and that the distributive, involution, and de Morgan's laws hold.

From Table 3 it is clear that Metze's algebra satisfies also the absorption, identity, and bounding laws. Hence $\mathbf{A_4}$ is a de Morgan algebra. It is not a Boolean algebra, since

$$0/1 + \overline{0/1} = 0/1 + 1/0 = 0/1 \neq 1.$$

Since $\mathbf{A_4}$ does not have a self-complementary element, it is not a ternary algebra.

Metze's algebra is at least as powerful as $\mathbf{A_3}$, since the mapping $h : \mathbf{A_4} \to \mathbf{A_3}$, defined below is a surjective homomorphism:

$$h(1) = 1, \quad h(0) = 0, \quad h(0/1) = h(1/0) = \Phi.$$

Thus, if we perform a simulation in $\mathbf{A_4}$ and apply this homomorphism to the results, we obtain the results of ternary simulation.

Metze presents some examples in which one can detect hazards by examining sequences of values that can occur in a circuit output. For example, if an output sequence is $0, 1/0, 1$, then there is dynamic hazard. However, these methods are rather involved, and no general algorithm is described.

Summary: From the hazard point of view, $\mathbf{A_4}$ is a step in the right direction, but has a flaw, as we discuss in the next section. Algebra $\mathbf{A_4}$ is a de Morgan algebra. A set-theoretic characterization of de Morgan algebras was recently discovered by Brzozowski [7].

5 The five-valued algebra

In 1972 Lewis presents a successful generalization of $\mathbf{A_3}$ [32], and corrects the flaw in $\mathbf{A_4}$. He notes that $\mathbf{A_4}$ has no element to represent a signal for which neither the actual value (0 or 1) nor the direction of a transition (if there is one) is known. Also, since $\mathbf{A_3}$ has no values to represent transitions without hazards, it cannot differentiate between such transitions and the corresponding dynamic hazards. Thus, Lewis adds Φ to Metze's algebra, and modifies the operations so that 0/1 and 1/0 now represent transitions without hazards. He defines a five-valued algebra, $\mathbf{A_5}$, with values (in our notation) 0, 1, 01 (representing a 0-to-1 transition without hazards), 10 (a 1-to-0 transition without hazards), and Φ (an unknown value). Table 4 shows the operations $+$ and $*$. The pairs $\{0, 1\}$ and $\{01, 10\}$ are complementary, and $\overline{\Phi} = \Phi$. Five-valued simulation is described by Breuer and Friedman [4].

Table 4. $+$ and $*$ operations for $\mathbf{A_5}$

$+$	0	01	Φ	10	1
0	0	01	Φ	10	1
01	01	01	Φ	Φ	1
Φ	Φ	Φ	Φ	Φ	1
10	10	Φ	Φ	10	1
1	1	1	1	1	1

$*$	0	01	Φ	10	1
0	0	0	0	0	0
01	0	01	Φ	Φ	01
Φ	0	Φ	Φ	Φ	Φ
10	0	Φ	Φ	10	10
1	0	01	Φ	10	1

Algebra $\mathbf{A_5}$ is not a lattice, since the absorption law fails:

$$01 * (01 + 10) = 01 * \Phi = \Phi \neq 01.$$

Distributivity does not hold because

$$01 * (10 + 1) = 01 \neq \Phi = \Phi + 01 = (01 * 10) + (01 * 1).$$

The five-valued algebra satisfies the laws of a de Morgan bisemilattice. Details of the verification are given in [3]. De Morgan bisemilattices were studied by Brzozowski [6], who showed that $\mathbf{A_5}$ is even more special, since it is a locally distributive de Morgan bilattice. He also provided a set-theoretic characterization for locally distributive de Morgan bilattices.

The partial orders for the five-valued algebra are given in Fig. 2 (a) and (b).

Hayes [25] uses $\mathbf{A_5}$ to give a second example of the application of his powerset method. Algebra $\mathbf{A_5}$ can be constructed using five elements of $2^{\mathcal{A}_2 \times \mathcal{A}_2}$, with $\mathcal{A}_2 = \{0, 1\}$, and with the following correspondence:

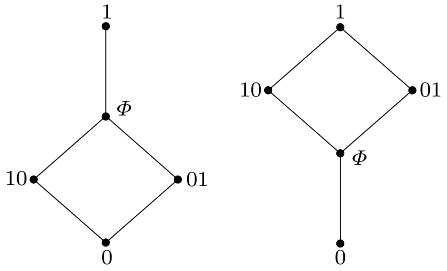

Fig. 2. Partial orders for $\mathbf{A_5}$: (a) $\leq$; (b) $\sqsubseteq$

$$\begin{aligned}
0 &\mapsto \{(0,0)\}\\
1 &\mapsto \{(1,1)\}\\
01 &\mapsto \{(0,0),(0,1),(1,1)\}\\
10 &\mapsto \{(1,1),(1,0),(0,0)\}\\
\varPhi &\mapsto \{(0,0),(0,1),(1,0),(1,1)\}
\end{aligned}$$

Summary: Algebra $\mathbf{A_5}$ is capable of detecting both static and dynamic hazards. It is a de Morgan bisemilattice and a locally distributive de Morgan bilattice [6], but not a de Morgan algebra. We return to $\mathbf{A_5}$ in Section 11.

6 The six-valued algebra

In this section it is convenient to discuss the second method of Hayes since the six-valued algebra is generated by this technique. This is the subdirect product [24] method. We obtain the underlying set $\mathcal{A}$ of a new algebra by taking a subset $\mathcal{S}$ of the direct product $\mathcal{P}$ of the underlying sets $\mathcal{A}_1, \ldots, \mathcal{A}_n$ of some n algebras, such that $\mathcal{S}$ determines a subalgebra of $\mathcal{P}$, and each element of each $\mathcal{A}_i$ appears as the ith component of some element of $\mathcal{S}$. We define the operations on $\mathcal{A}$ componentwise, using the operations on the n algebras. For example, algebra $\mathbf{A_6}$ is defined using $\mathbf{A_3}$ and two copies of $\mathbf{A_2}$. More specifically, $\mathbf{A_6} = \langle \mathcal{A}_6, +, *, ^-, 0, 1\rangle$, where

$$\mathcal{A}_6 = \{(0,0,0),(0,\varPhi,0),(0,\varPhi,1),(1,\varPhi,0),(1,\varPhi,1),(1,1,1)\}$$

is a subset of $\mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_2$, $\mathcal{A}_2 = \{0,1\}$, and $\mathcal{A}_3 = \{0,\varPhi,1\}$. For $+$, the first and third components are added in $\mathbf{A_2}$, and the second component, in $\mathbf{A_3}$; the other two operations are handled similarly. For example,

$$(0,0,0)+(1,\varPhi,0) = (0+1, 0+\varPhi, 0+0) = (1,\varPhi,0),$$

and

$$\overline{(0,\varPhi,1)} = (\overline{0},\overline{\varPhi},\overline{1}) = (1,\varPhi,0).$$

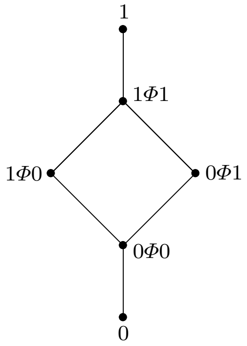

Fig. 3. Partial order for $\mathbf{A_6}$

In an algebra constructed by the subdirect product method, a law holds iff it holds in all the component algebras. Thus, since both $\mathbf{A_2}$ and $\mathbf{A_3}$ are de Morgan algebras, we know that $\mathbf{A_6}$ is also a de Morgan algebra. It is not a Boolean algebra since the complement laws do not hold in $\mathbf{A_3}$, and it is not a ternary algebra, since there is no self-complemented element in $\mathbf{A_2}$.

Hayes interprets the ordered triples (a, b, c) in $\mathcal{A}_6$ as follows: a is the initial binary value of a signal, b, the transient value, and c, the final binary value. Thus, $(0, 0, 0)$ represents a constant signal with value 0, and $(0, \Phi, 0)$ represents a static hazard, since the transient component has the unknown value. The value $(0, \Phi, 1)$ represents a 0-to-1 transition, with or without hazards. The remaining values are similarly interpreted. It is clear that $\mathbf{A_6}$ is capable of representing static hazards, but not dynamic hazards.

In our notation we represent the elements of $\mathbf{A_6}$ as 0, $0\Phi0$, 1, $1\Phi1$, $0\Phi1$, and $1\Phi0$. The partial order corresponding to both binary operations in $\mathbf{A_6}$ is given in Fig. 3.

Summary: Algebra $\mathbf{A_6}$ is capable of representing static, but not dynamic hazards. It is a de Morgan algebra, but not a Boolean or ternary algebra. We return to $\mathbf{A_6}$ in Section 11.

7 The eight-valued algebra

The recent work of Brzozowski and Ésik discussed in Section 11 shows that there is a natural seven-valued algebra, but this algebra has not been considered before. Thus, the next value used in the past is eight.

An eight-valued algebra, $\mathbf{A_8}$, is discussed by Hayes [25] in 1986 for representing static and dynamic hazards. Hayes points out that this algebra appears already in the 1974 paper by Breuer and Harrison [5]. In that work, however, this algebra is not explicitly used, but is part of a 27-element algebra, to which we return later. In [5] the values consist of the pairs $(0, 0), (0, 1), (1, 0), (1, 1)$,

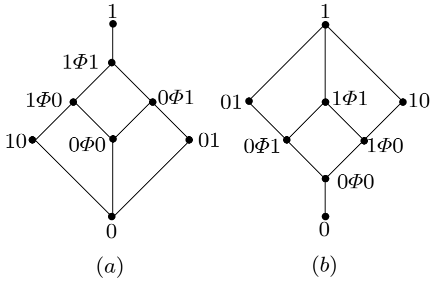

Fig. 4. Partial orders for $\mathbf{A_8}$: (a) $\leq$; (b) $\sqsubseteq$

which represent the levels of a signal before and after a change, and each pair is accompanied by a hazard status bit indicating whether or not a hazard occurs during the transition, for a total of eight values. Hayes also points out that the eight-valued algebra appears implicitly in the 1974 work of Fantauzzi [20], but there it is a part of a nine-valued algebra. Fantauzzi uses a different set of symbols, but the basic idea is the same. We return to his algebra in Section 8. Eight-valued simulation is described by Breuer and Friedman [4]. Algebra $\mathbf{A_8}$ is also used in [1, 26].

Using the subdirect product construction, Hayes generates the underlying set $\mathcal{A}_8$ of $\mathbf{A_8}$ as a subset of $\mathcal{A}_2 \times \mathcal{A}_5 \times \mathcal{A}_2$, or equivalently of $\mathcal{A}_2 \times (2^{\mathcal{A}_2 \times \mathcal{A}_2}) \times \mathcal{A}_2$. Algebra $\mathbf{A_2}$ is a Boolean algebra, and thus a de Morgan bisemilattice. Algebra $\mathbf{A_5}$ is a de Morgan bisemilattice, but is not a lattice. It is therefore immediate that $\mathbf{A_8}$ is a de Morgan bisemilattice. The partial orders for $\mathbf{A_8}$ are given, in our notation, in Fig. 4.

Summary: Algebra $\mathbf{A_8}$ is capable of representing both static and dynamic hazards. It is a de Morgan bisemilattice, but not a lattice. We return to $\mathbf{A_8}$ in Section 11.

8 The nine-valued algebra

As we have mentioned above, a nine-valued algebra, $\mathbf{A_9}$, was introduced by Fantauzzi [20] in 1974. The values are those of $\mathbf{A_8}$ together with a value called "ambiguous" by the author; we represent this value by Φ.

Like all hazard algebras, $\mathbf{A_9}$ has two binary operations, $+$ and $*$, and a unary operation $^-$. Hayes [25] argues that the set $\mathcal{A}_9 = \mathcal{A}_8 \cup \{\Phi\}$ is not closed under the operations $*$ and $+$. When $\mathcal{A}_9$ is viewed as consisting of ordered triples, it is easy to see that this is indeed the case. While $(1, \Phi, 0) * (\Phi, \Phi, \Phi)$ should give $(\Phi, \Phi, 0)$, this last value is not in $\mathcal{A}_9$. Fantauzzi defines $(1, \Phi, 0) * (\Phi, \Phi, \Phi)$ to be (Φ, Φ, Φ), and this results in operations $*$ and $+$ which are not associative.

Algebra $\mathbf{A_9}$ is also used in [31].

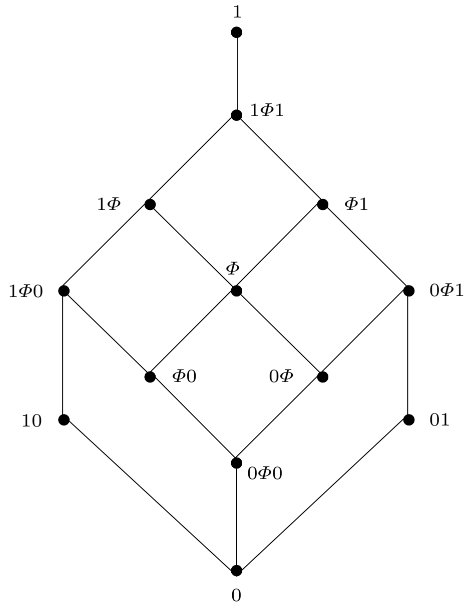

Fig. 5. Partial order $\leq$ for $\mathbf{A_{13}}$

Summary: Algebra $\mathbf{A_9}$ is flawed, since the binary operations are ill defined. The subalgebra $\mathbf{A_8}$ of $\mathbf{A_9}$ is well defined, as we have stated above. There does exist a useful hazard algebra with nine-elements, as we show in Section 11.[1]

9 The 13-valued algebra

As shown in Section 11, there is a hazard algebra with eleven elements, but the next value considered in the literature is thirteen.

Algebra $\mathbf{A_{13}}$ is due to Hayes [25]. It is generated by the subdirect product construction, and its underlying set is a subset of $\mathcal{A}_3 \times \mathcal{A}_5 \times \mathcal{A}_3$. The laws of a ternary algebra are a superset of those of a de Morgan bisemilattice. Algebra $\mathcal{A}_3$ is a ternary algebra, and $\mathcal{A}_5$ is a de Morgan bisemilattice. Hence, $\mathbf{A_{13}}$ is a de Morgan bisemilattice. This was first observed by Beare and Brzozowski [3]. The partial order $\leq$ corresponding to $+$ for $\mathbf{A_{13}}$ is shown in Fig. 5 in our notation.

[1] A nine valued algebra is used by Muth [37] to generalize the D-algorithm for test generation, but has not been applied to hazard detection. It is the direct product $\mathbf{A_3} \times \mathbf{A_3}$, and so is a ternary algebra. A nine-valued simulator is also used by Knudsen for NMOS circuits [29, 30]. The nine values are 0, 1, three rising values (below threshold, critically near threshold, and above threshold), three falling values, and the unknown value. The simulator has some hazard-detecting capabilities, but NMOS circuits are outside the scope of this paper.

The thirteen-valued algebra can represent all of the states used in the eight-valued algebra. In addition, there are values to represent a completely unknown signal, (Φ, Φ, Φ) (which we denote simply by Φ), signals $(0, \Phi, \Phi)$ and $(1, \Phi, \Phi)$ starting at 0 or 1 and becoming unknown (we denote them by 0Φ and 1Φ), and signals $(\Phi, \Phi, 0)$ and $(\Phi, \Phi, 1)$ (denoted by $\Phi 0$ and $\Phi 1$), which begin in an unknown state and change to 0 or 1. In our notation each element is a word over the alphabet $\{0, 1, \Phi\}$, as is further explained in Section 11.

Chakraborty, Agrawal and Bushnell [14] rediscovered the thirteen-valued algebra independently in 1992. They give an alternate method of constructing $\mathbf{A_{13}}$. They examine all triples in $\mathbf{A_3} \times \mathbf{A_2} \times \mathbf{A_3}$. The first coordinate is the initial value, the third coordinate is the final value and the second coordinate indicates whether or not a hazard occurs in the transition. The eighteen elements so obtained are reduced to thirteen by noting that for signals with unknown initial or final states, it is not useful to specify the presence or absence of hazards. Hence, any two signals $(a, b, 0)$ and $(a, b, 1)$ are identified, if either a or b or both are Φ. Algebra $\mathbf{A_{13}}$ is also used in [15, 16]. Precise simulation algorithms (like Algorithms A and B for ternary simulation) for general sequential circuits have not been studied for $\mathbf{A_{13}}$.

Summary: Algebra $\mathbf{A_{13}}$ is the most complete of the algebras discussed above for analyzing circuits with respect to hazards and unknown signals, provided that counting hazard pulses (see Section 11) is not important. Algebra $\mathbf{A_{13}}$ is capable of representing static and dynamic hazards and unknown values. It is de Morgan bisemilattice, but not a lattice.

10 The 27-valued algebra

Next, we briefly consider the 27-valued algebra introduced by Breuer and Harrison [5] for eliminating static and dynamic hazards in test generation.[2] This algebra is the direct product $\mathbf{A_3} \times \mathbf{A_3}$, with an additional three-valued component, and the following interpretation. Each element of the algebra is an ordered triple (a, b, c), where a and b are the initial and final values of a transition, each from the set $\{0, \Phi, 1\}$ (in our notation), and c is the hazard status component taking its values from $\{hf, hsu, hp\}$ representing "hazard-free," "hazard status unknown," and "hazard present." The algebra is flawed, as $(0, 1, hp) * (1, 1, hsu)$ should give $(0, 1, hp)$, but instead is defined as $(0, 1, hsu)$. This results in a non-associative algebra. For example,

$$((0, 1, hp) * (1, 1, hsu)) * (0, 1, hsu) \neq (0, 1, hp) * ((1, 1, hsu) * (0, 1, hsu)).$$

[2] Hławiczka and Badura [26] describe a 16-valued algebra for analyzing flow tables of asynchronous circuits in order to detect various types of conditions, such as critical races, essential hazards, and D-trios [42]. A discussion of their work is outside the scope of this paper. The authors observe that their algebra satisfies laws L1-L3, L5, L8, L9, and their primed versions. Since this algebra also satisfies L6, it is, in fact, a de Morgan bisemilattice.

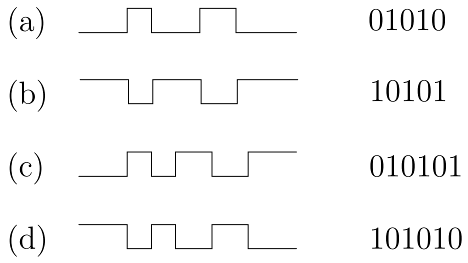

Fig. 6. Transients: (a) constant 0 with static hazards; (b) constant 1 with static hazards; (c) change from 0 to 1 with dynamic hazards; (d) change from 1 to 0 with dynamic hazards

Summary: The 27-valued algebra is flawed, since it labels an output as "hazard status unknown" when there should be a hazard. Consequently, it is not an associative algebra. As we have mentioned above, the eight-valued subalgebra $\mathbf{A_8}$ of $\mathbf{A_{27}}$ is well defined and useful.

11 Change-counting algebras

This section is a summary of the recent work of Brzozowski and Ésik [8, 9] on an infinite hazard algebra.

11.1 The infinite algebra C

Consider waveforms with a constant initial value, a transient period involving a finite number of changes, and a constant final value. Waveforms of this type are called *transients*. Figure 6 gives four examples of transients. With each such transient is associated a binary word, *i.e.,* a sequence of 0s and 1s, in a natural way. In this binary word a 0 (1) represents a maximal interval during which the signal has the value 0 (1). Such an interval is called a *0-interval* (*1-interval*).

In this section we use $\vee$, $\wedge$, and $^-$ for the Boolean OR, AND, and NOT operations, respectively. For the present, assume that circuits are constructed with 2-input OR gates, 2-input AND gates and inverters. We examine how transients are processed by such gates. The case of the inverter is the easiest one. If $t = a_1 \ldots a_j$ is the binary word of a transient at the input of an inverter, then its output has the transient $\overline{t} = \overline{a_1} \ldots \overline{a_j}$. For example, in Fig. 6, the first two transients are complementary, as are the last two. The following propositions show the largest number of changes possible at the output of an OR gate and an AND gate.

Proposition 1. *If the inputs of an* OR *gate have m and n 0-intervals respectively, then the maximum number of 0-intervals in the output signal is 0 if $m = 0$ or $n = 0$, and is $m + n - 1$, otherwise.*

Proposition 2. *If the inputs of an* AND *gate have m and n 1-intervals respectively, then the maximum number of 1-intervals in the output signal is 0 if $m = 0$ or $n = 0$, and is $m + n - 1$, otherwise.*

Let

$$T = \{0, 1, 01, 10, 010, 101, 0101, 1010, 01010, \ldots\}$$

be the set of all binary words of alternating 0s and 1s. In regular expression notation,

$$T = 0(10)^* \cup 1(01)^* \cup 0(10)^*1 \cup 1(01)^*0.$$

This is the set of all transients. The *change-counting algebra* is defined as $\mathbf{C} = (T, +, *, ^-, 0, 1)$, where $+$, $*$ and $^-$ are defined below. For any $t \in T$ define $z(t)$ (z for zeros) and $u(t)$ (u for units) to be the number of 0s in t and the number of 1s in t, respectively. Let $\alpha(t)$ and $\omega(t)$ be the first and last letters of t, and let $l(t)$ denote the length of t. For example, if $t = 10101$, then $z(t) = 2$, $u(t) = 3$, $\alpha(t) = \omega(t) = 1$, and $l(t) = 5$.

Operations $+$ and $*$ are binary operations on T intended to represent the worst-case OR-ing and AND-ing of two transients at the inputs of a gate. We assume that input changes specified by the transients occurring at the inputs of a gate can take place at any time. For example, consider an OR gate with inputs X_1 and X_2 and transients 01 and 010, respectively. The input changes can occur in any one of the following orders: $X_1X_2X_2$, $X_2X_1X_2$, or $X_2X_2X_1$. One verifies that, in the first two cases, the output transient is 01, whereas in the last case, that transient is 0101. Therefore, we define $01 + 010 = 0101$. In general, the $+$ and $*$ operations are defined as follows:

$$t + 0 = 0 + t = t, \quad t + 1 = 1 + t = 1,$$

for any $t \in T$. If w and w' are words in T of length > 1, and their *sum* is denoted by $t = w + w'$, then t is that word in T that begins with $\alpha(w) \vee \alpha(w')$, ends with $\omega(w) \vee \omega(w')$, and has $z(t) = z(w) + z(w') - 1$, by Proposition 1. For example, $010 + 1010 = 101010$.

Next, define

$$t * 1 = 1 * t = t, \quad t * 0 = 0 * t = 0,$$

for any $t \in T$. Consider now the *product* of two words $w, w' \in T$ of length > 1, and denote it by $t = w * w'$. Then t is that word in T that begins with $\alpha(w) \wedge \alpha(w')$, ends with $\omega(w) \wedge \omega(w')$, and has $u(t) = u(w) + u(w') - 1$, by Proposition 2. For example, $0101 * 10101 = 01010101$.

The *(quasi-)complement* $\overline{t}$ of a word $t \in T$ is obtained by complementing each letter in t. For example, $\overline{1010} = 0101$. The constants 0 and 1 are the words 0 and 1 of length 1.

Proposition 3. *Algebra* $\mathbf{C} = (T, +, *, ^-, 0, 1)$*, is a commutative de Morgan bisemigroup.*

11.2 Counting changes to a threshold

Since the underlying set T of algebra $\mathbf{C}$ is infinite, an arbitrary number of changes can be counted. An alternative is to count only up to some threshold $k \geq 1$, and consider all transients with length k or more as equivalent.

Relation $\sim_k$ in algebra $\mathbf{C} = (T, +, *, ^-, 0, 1)$ is defined as follows: For $t, s \in T$, $t \sim_k s$ if either $t = s$ or t and s are both of length $\geq k$.

Proposition 4. *Relation $\sim_k$ is a congruence relation on $\mathbf{C}$, by which we mean that it is an equivalence relation on T such that for all $t, s, w \in T$, $t \sim_k s$ implies $(w+t) \sim_k (w+s)$, and $\overline{t} \sim_k \overline{s}$; this then implies that $(w*t) \sim_k (w*s)$ whenever $t \sim_k s$.*

The equivalence classes of the quotient algebra $\mathbf{C_k} = \mathbf{C}/{\sim_k}$ are of two types. Each transient t with $l(t) < k$ is in a class by itself, and all the words of length $\geq k$ constitute a class, which is denoted by Φ. The operations on equivalence classes are as follows. The complement of the class containing t is the class containing $\overline{t}$. The sum (product) of the class containing t and the class containing t' is the class containing $t + t'$ $(t * t')$. Thus, the quotient algebra $\mathbf{C_k}$ is a commutative de Morgan bisemigroup with $2k - 1$ elements.

Example 1. The following are examples of quotient algebras:

- For $k = 2$, the operations $+$, $*$, and $^-$ are those of the 3-element ternary algebra $\mathbf{A_3}$. Hence $\mathbf{A_3}$ is isomorphic to $\mathbf{C_2}$.
- For $k = 3$, the operations are those of the 5-element ternary algebra $\mathbf{A_5}$. Hence $\mathbf{A_5}$ is isomorphic to $\mathbf{C_3}$.
- In general, for each $k \geq 2$ there is an algebra $\mathbf{C_k}$ with $2k - 1$ elements. □

11.3 Circuit Simulation in algebras C and $\mathbf{C_k}$

The following example illustrates how static hazards are detected by ternary simulation.

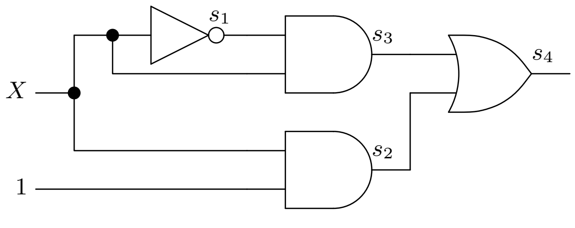

Fig. 7. Circuit with hazards

Table 5. Ternary simulation

	X	s_1	s_2	s_3	s_4
initial state	0	1	0	0	0
	Φ	1	0	0	0
	Φ	Φ	Φ	Φ	0
result A	Φ	Φ	Φ	Φ	Φ
	1	Φ	Φ	Φ	Φ
	1	0	1	Φ	Φ
result B	1	0	1	0	1

Example 2. Consider the circuit shown in Fig. 5. Refer now to Table 5. The values of the input variable X and the state variables $s_1, \ldots, s_4$ are shown in rows as the simulation progresses. We begin in the initial state 01000, which is stable. We wish to study the behavior of the circuit when the input changes from 0 to 1 and is kept constant at 1.

Ternary simulation consists of two algorithms, A and B. In Algorithm A, the input changes to the uncertain or unknown value Φ. Instead of Boolean functions, we now use the ternary functions as defined in Section 3. Thus we use the excitation equations

$$S_1 = \overline{X},\ S_2 = 1 * X,\ S_3 = X * s_1,\ S_4 = s_2 + s_3.$$

After X changes to Φ, Gates 1, 2, and 3, become unstable in the ternary model. All unstable variables are changed at the same time. This results in the second row of Algorithm A. Now Gate 4 becomes unstable and changes, to yield the third row of Algorithm A.

In Algorithm A we introduce uncertainty in the circuit inputs and we see how this uncertainty spreads throughout the circuit. In Algorithm B we start in the state produced by Algorithm A, but now we set the changing inputs from Φ to their final values, thus reducing uncertainty. In our example, X becomes 1. We again use the ternary excitation functions to see whether the uncertainty will be removed from any gates. The final result is the vector 10101, showing that each gate reaches a binary value after the transient is over.

It is clear from the circuit diagram that s_1 changes from 1 to 0 and s_2 changes from 0 to 1 without any hazards. By Boolean analysis, it can be verified [13] that a dynamic hazard is present in s_4.

This example illustrates that ternary simulation is capable of detecting static hazards. Gate s_3 is 0 at the beginning and 0 at the end, but it is Φ at the end of Algorithm A, and this indicates a static hazard [13]. Our example also clearly shows that ternary simulation is not capable of detecting dynamic hazards. Gates s_2 and s_4 both change from 0 to Φ to 1, yet s_2 has no dynamic hazard, while s_4 has one. □

In the examples that follow, we show how the accuracy of the simulation improves when we use algebra $\mathbf{C_k}$ as k increases.

Example 3. In Table 6 we repeat the simulation, this time using five-valued gate functions as defined in $\mathbf{C_3}$. Instead to changing X to Φ, we change it to 01 in Algorithm A, since this is the change we wish to study. From the result of Algorithm A we see that s_1 changes from 1 to 0 and s_2 changes from 0 to 1, both without hazards. The static hazard in s_3 is detected as before, as is the dynamic hazard in s_4. This example shows that quinary simulation is capable of detecting both static and dynamic hazards. □

Table 6. Quinary simulation

	X	s_1	s_2	s_3	s_4
initial state	0	1	0	0	0
	01	1	0	0	0
	01	10	01	01	0
	01	10	01	Φ	01
result A	01	10	01	Φ	Φ
	1	10	01	Φ	Φ
	1	0	1	10	Φ
result B	1	0	1	0	1

Table 7. Septenary simulation

	X	s_1	s_2	s_3	s_4
initial state	0	1	0	0	0
	01	1	0	0	0
	01	10	01	01	0
	01	10	01	010	01
result A	01	10	01	010	Φ
	1	10	01	010	Φ
	1	0	1	10	Φ
result B	1	0	1	0	1

Example 4. We repeat the simulation now using algebra $\mathbf{C_4}$ with seven values. Refer to Table 7. This time Algorithm A not only reveals that there is a static hazard in s_3, but also identifies it as 010. Note, however, that the dynamic hazard is still not identified. □

Example 5. We repeat the simulation using algebra $\mathbf{C_5}$ with nine values. Refer to Table 8. This time Algorithm A identifies the dynamic hazard as 0101.

Table 8. Nonary simulation

	X	s_1	s_2	s_3	s_4
initial state	0	1	0	0	0
	01	1	0	0	0
	01	10	01	01	0
	01	10	01	010	01
result A	01	10	01	010	0101
	1	10	01	010	0101
	1	0	1	10	0101
result B	1	0	1	0	1

Observe that the same table results if we simulate our circuit in algebra $\mathbf{C_k}$ with any $k \geq 5$, or in algebra $\mathbf{C}$. Note also, that for $k \geq 4$ Algorithm B is no longer necessary, since the entire history of worst-case signal changes is recorded on each wire. □

A characterization of the results of simulation in algebra $\mathbf{C}$ has been recently obtained by Gheorghiu [21] for feedback-free circuits consisting of 2-input AND gates, 2-input OR gates and inverters.

11.4 Simulation with initial, transient, and final values

In a number of simulators [14, 25], the signal values are ordered triples containing the initial, transient, and final values of a signal. We now show how such algebras can be described in our framework.

Relation $\approx_k$ in the algebra $\mathbf{C} = (T, +, *, ^-, 0, 1)$ is defined as follows. For $t, s \in T$, $t \approx_k s$ if either $t = s$ or $\alpha(t) = \alpha(s)$, $\omega(t) = \omega(s)$, and t and s are both of length $\geq k$. Denote by λ (for left) and ρ (for right) the congruences defined by $t\,\lambda\,s$ iff $\alpha(t) = \alpha(s)$, $t\,\rho\,s$ iff $\omega(t) = \omega(s)$. Then $\approx_k = \lambda \cap \sim_k \cap \rho$.

Proposition 5. *Relation $\approx_k$ is a congruence relation on $\mathbf{C}$.*

The quotient algebra $\mathbf{C'_k} = \mathbf{C}/{\approx_k}$ is a commutative de Morgan bisemigroup with $2(k-1)+4 = 2k+2$ elements. Each word $t \in T$ with $l(t) < k$ determines a singleton congruence class. In addition, for any $b_1, b_2 \in \{0, 1\}$, the words $t \in T$ with $l(t) \geq k$, $\alpha(t) = b_1$, and $\omega(t) = b_2$ determine a congruence class that we denote by $b_1\Phi b_2$. Since $\approx_k \subseteq \sim_k$, $\mathbf{C_k}$ is a quotient of $\mathbf{C'_k}$. It can be constructed from $\mathbf{C'_k}$ by identifying the four elements $0\Phi0$, $0\Phi1$, $1\Phi0$, and $1\Phi1$. Also, if $k \leq m$, then $\mathbf{C'_k}$ is a quotient of $\mathbf{C'_m}$.

Proposition 6. $\mathbf{C'_k}$ *is isomorphic to a subdirect product of two copies of the 2-element Boolean algebra* $\mathbf{A_2}$ *and algebra* $\mathbf{C_k}$*.*

Example 6. The following are examples of algebras $\mathbf{C'_k}$:

- For $k = 2$, there is a six-element algebra $\mathbf{C'_2}$. It is isomorphic to $\mathbf{A_6}$.
- For $k = 3$, there is a eight-element algebra $\mathbf{C'_3}$. It is isomorphic to $\mathbf{A_8}$.
- In general, for any $k \geq 2$ there is a $(2k + 2)$-element algebra $\mathbf{C'_k}$. □

It is shown in [9] that simulation in $\mathbf{C'_k}$ may not terminate for circuits with feedback; hence this approach is not suitable for such circuits.

11.5 Simulation with unknown values

As Hayes [25] points out, most digital simulators include an unknown value. Algebra $\mathbf{A_{13}}$ is an example of an algebra that allows us to represent a value that is completely unknown, as well as a value that is unknown initially, but becomes known at the end, and one that is known initially, but unknown at the end. This example can be generalized as follows. Consider $\mathbf{C'_k} = (T'_k, +, *, ^-, 0, 1)$, where

$$T'_k = T_k \setminus \{\Phi\} \cup \{0\Phi 0, 0\Phi 1, 1\Phi 0, 1\Phi 1\},$$

and the operations of $\mathbf{C'_k}$ are defined as above. We define a new algebra, $\mathbf{C''_k}$, by adjoining five elements to $\mathbf{C'_k}$. Let

$$T''_k = T'_k \cup \{0\Phi, 1\Phi, \Phi 0, \Phi 1, \Phi\}.$$

We consider all the elements of T''_k to be words over the alphabet $\{0, 1, \Phi\}$. Complementation in $\mathbf{C''_k}$ is letter by letter complementation in the three-valued algebra $\mathbf{A_3}$. For example, $\overline{0\Phi 1} = 1\Phi 0$. Addition is defined as follows. First,

$$t + 0 = 0 + t = t, \quad t + 1 = 1 + t = 1,$$

for any $t \in T''_k$. Next, if w and w' are words in T''_k of length > 1, then their *sum* is denoted by $t = w + w'$, and is a word in T''_k that begins with $\alpha(t) = \alpha(w) \vee \alpha(w')$, and ends with $\omega(t) = \omega(w) \vee \omega(w')$. It remains to compute the middle portion of t, if any. We have $t = \Phi$, if $\alpha(t) = \omega(t) = \Phi$, $t = \Phi a$, if $\alpha(t) = \Phi, \omega(t) = a \in \{0, 1\}$, and $t = a\Phi$, if $\alpha(t) = a \in \{0, 1\}, \omega(t) = \Phi$.

There remain only the cases where both $\alpha(t)$ and $\omega(t)$ are in $\{0, 1\}$, *i.e.*, where both w and w' are in T'_k. Here the rules of $\mathbf{C'_k}$ apply.

Example 7. The following are examples of algebras $\mathbf{C''_k}$:

- For $k = 3$, there is a thirteen-element algebra $\mathbf{C''_3}$. It is isomorphic to $\mathbf{A_{13}}$.
- In general, for any $k \geq 2$ there is a $2k + 7$-element algebra $\mathbf{C''_k}$. □

Summary: Algebra $\mathbf{C}$ leads to a general theory of simulation of gate circuits for the purpose of hazard detection, identification, and counting. The same simulation algorithms can be used to count the number of signal changes during a given input change of a circuit. This provides an estimate of the worst-case energy consumption of that input change. If a circuit has m inputs and n gates, the simulation algorithms run in $O(m + n^2)$ time. By choosing the value of the threshold k one can count signal changes and hazards to any degree of accuracy. For further properties of these algebras see [8, 9].

12 Conclusions

We have presented a survey of the algebras that have been designed for the detection of static and dynamic hazards. We have generalized the simulation algorithms, previously used only in the three-valued algebra, to the change-counting algebra $\mathbf{C}$ and its related algebras $\mathbf{C_k}$. We have provided a single framework which includes all the successful hazard algebras as special cases.

Acknowledgments: This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871, and Grant No. T30511 from the National Foundation of Hungary for Scientific Research.

References

1. R. Andrew, "An Algorithm for Eight-Valued Simulation and Hazard Detection in Gate Networks," *Proc. 16th Int. Symp. on Multiple-Valued Logic*, pp. 273–280, IEEE, 1986.
2. R. Balbes, "Free Ternary Algebras," *Int. J. Algebra and Computation*, vol. 10, no. 6, pp. 739–749, 2000.
3. B. D. Beare and J. A. Brzozowski, "An Exploration of the Properties of a Thirteen-Valued Algebra," unpublished, 1999.
4. M. A. Breuer and A. D. Friedman, *Diagnosis & Reliable Design of Digital Systems,* Computer Science Press, 1976.
5. M. A. Breuer and R. L. Harrison, "Procedures for Eliminating Static and Dynamic Hazards in Test Generation," *IEEE Trans. Comput.*, vol. C-23, no. 10, pp. 1069–1078, Oct. 1974.
6. J. A. Brzozowski, "De Morgan Bisemilattices," *Int. Symp. on Multiple-Valued Logic*, pp. 173–178, IEEE, 2000.
7. J. A. Brzozowski, "A Characterization of de Morgan Algebras," *Int. Journal of Algebra and Computation*. Vol. 11, No. 5, pp. 525–527, Oct. 2001.
8. J. A. Brzozowski and Z. Ésik, "Hazard Algebras" (Extended Abstract), *A Half-Century of Automata Theory*, A. Salomaa, D. Wood, and S. Yu, eds., pp. 1–19, World Scientific, Singapore, 2001.
9. J. A. Brzozowski and Z. Ésik, "Hazard Algebras," Maveric Report 00-2, University of Waterloo, Waterloo, ON, Canada, 27 pp., July 2000; revised December, 2001. `http://maveric.uwaterloo.ca/publication.html`
10. J. A. Brzozowski, J. J. Lou, and R. Negulescu, "A Characterization of Finite Ternary Algebras," *Int. J. Algebra and Computation*, vol. 7, no. 6, pp. 713–721, 1997.
11. J. A. Brzozowski and C-J. Seger, "Correspondence between Ternary Simulation and Binary Race Analysis in Gate Networks," *Proc. Coll. Automata, Languages and Programming*, L. Kott, ed., Springer-Verlag, Berlin, pp. 69–78, July 1986.
12. J. A. Brzozowski and C-J. Seger, "A Characterization of Ternary Simulation of Gate Networks," *IEEE Trans. Computers*, vol. C-36, no. 11, pp. 1318–1327, November 1987.
13. J. A. Brzozowski and C-J. Seger, *Asynchronous Circuits*, Springer-Verlag, Berlin, 1995.

14. T. Chakraborty, V. Agrawal and M. Bushnell, "Delay Fault Models and Test Generation for Random Logic Sequential Circuits," *Proc. Design Automation Conf.*, pp. 165–172, IEEE, June 1992.
15. S. Chakraborty and D. L. Dill, "More Accurate Polynomial-Time Min-Max Timing Simulation," *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, pp. 112–123, April 1997.
16. S. Chakraborty, D. L. Dill, and K. Y. Yun, "Min-Max Timing Analysis and An Application to Asynchronous Circuits," *Proc. IEEE*, pp. 1–13, 1999.
17. E. B. Eichelberger, "Hazard Detection in Combinational and Sequential Switching Circuits," *IBM J. Res. and Dev.*, vol. 9, pp. 90–99, 1965.
18. Z. Ésik, "A Cayley Theorem for Ternary Algebras," *Int. J. Algebra and Computation* vol. 8, no. 3, pp. 311–316, 1998.
19. Z. Ésik, "Free De Morgan bisemingroups and bisemilattices," Algebra Colloquium, to appear.
20. G. Fantauzzi, "An Algebraic Model for the Analysis of Logical Circuits," *IEEE Trans. Computers*, vol. C–23, no. 6, pp. 576–581, June 1974.
21. M. Gheorghiu, *Circuit Simulation Using a Hazard Algebra*, MMath Thesis, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, December 2001. `http://maveric.uwaterloo.ca/publication.html`
22. M. Goto, "Application of Three-Valued Logic to Construct the Theory of Relay Networks" (in Japanese), *Proc. Joint Meeting IEE, IECE, and I. of Illum. E. of Japan,* 1948.
23. M. Goto, "Application of Logical Mathematics to the Theory of Relay Networks" (in Japanese), *J. Inst. Elec. Eng. of Japan,* vol. 69, no. 729, 1949.
24. G. Grätzer, *Universal Algebra*, Second Edition, Springer-Verlag, 1979.
25. J. P. Hayes, "Digital Simulation with Multiple Logic Values," *IEEE Trans. CAD*, vol. CAD–5, no. 2, pp. 274–283, 1986.
26. A. Hławiczka and D. Badura, "The Method of Recognition of Critical Hazards, Critical Races, Essential Hazards and D-Trio," *Proc. Int. Symp. Multiple-Valued Logic*, pp. 298–311, IEEE, 1982.
27. D. A. Huffman, "The Design and Use of Hazard-Free Switching Circuits," *J. ACM*, vol. 4, pp. 47–62, 1957.
28. W. Keister, A. E. Ritchie, and S. H. Washburn, *The Design of Switching Circuits*, D. Van Nostrand, New York, 1951.
29. M. S. Knudsen, "A Compact Nine-Valued Logic Simulation Algorithm," *Proc. Int. Symp. Circuits and Systems*, vol. 3, pp. 1190–1193, IEEE, 1982.
30. M. S. Knudsen, "A Nine-Valued Logic Simulator for Digital N-MOS Circuits," *Proc. Int. Symp. Multiple-Valued Logic*, pp. 293–297, IEEE, 1982.
31. D. S. Kung, "Hazard-Non-Increasing Gate-Level Optimization Algorithms," *Proc. Int. Conf. Computer-Aided Design*, pp. 631–634, IEEE, 1992.
32. D. W. Lewis, *Hazard Detection by a Quinary Simulation of Logic Devices with Bounded Propagation Delays, MSc Thesis*, University of Syracuse, 1972.
33. E. J. McCluskey, "Transient Behavior of Combinational Logic Circuits," in *Redundancy Techniques for Computing Systems*, R. H. Wilcox and W. C. Mann, eds., Spartan Books, Washington, DC, pp. 9–46, 1962.
34. G. A. Metze, *Many-Valued Logic and the Design of Switching Circuits, MSc Thesis*, University of Illinois, Urbana, 1953.
35. Gr. C. Moisil, "Sur l'application des logiques à trois valeurs à l'étude des schémas à contacts et relais," *Actes proc. congr. intern. automatique,* p. 48, 1956.

36. D. E. Muller, “Treatment of Transition Signals in Electronic Switching Circuits by Algebraic Methods,” *IRE Trans. Electronic Computers,* vol. EC–8, no. 3, p. 401, September, 1959.
37. P. Muth, “A Nine-Valued Circuit Model for Test Generation,” *IEEE Trans. Computers*, vol. C–25, no. 6, June 1976.
38. V. N. Roginskii, “The Operation of Relay Networks in Transitional Periods,” *Avtomatika i Telemekhanika,* vol. 20, no. 10, pp. 1408–1416, October 1959.
39. C-J. Seger and J. A. Brzozowski, “Generalized Ternary Simulation of Sequential Circuits,” *Theoretical Informatics and Applications*, vol. 28, No. 3–4, pp. 159–186, 1994.
40. C. E. Shannon, “A Symbolic Analysis of Relay and Switching Circuits,” *Trans. AIEE,* vol. 57, pp. 713–723, 1938.
41. E. W. Thompson and S. A. Szygenda, “Three Levels of Accuracy for the Simulation of Different Fault Types in Digital Systems,” Proc. Design Automation Conference pp. 105-113, IEEE, 1975.
42. S. H. Unger, *Asynchronous Sequential Switching Circuits*, Wiley-Interscience, New York, 1969.
43. M. Yoeli and S. Rinon, “Application of Ternary Algebra to the Study of Static Hazards,” *J. ACM,* vol. 11, no. 1, pp. 84–97, January 1964.