

Dynamic Algebras: Examples, Constructions, Applications

Vaughan Pratt*

Dept. of Computer Science, Stanford, CA 94305

Abstract

Dynamic algebras combine the classes of Boolean ($B \vee ' 0$) and regular ($R \cup ; *$) algebras into a single finitely axiomatized variety ($\mathcal{B} \mathcal{R} \diamond$) resembling an R -module with “scalar” multiplication \diamond . The basic result is that $*$ is reflexive transitive closure, contrary to the intuition that this concept should require quantifiers for its definition. Using this result we give several examples of dynamic algebras arising naturally in connection with additive functions, binary relations, state trajectories, languages, and flowcharts. The main result is that free dynamic algebras are residually finite (i.e. factor as a subdirect product of finite dynamic algebras), important because finite separable dynamic algebras are isomorphic to Kripke structures. Applications include a new completeness proof for the Segerberg axiomatization of propositional dynamic logic, and yet another notion of regular algebra.

Key words: Dynamic algebra, logic, program verification, regular algebra.

This paper originally appeared as Technical Memo #138, Laboratory for Computer Science, MIT, July 1979, and was subsequently revised, shortened, retitled, and published as [Pra80b]. After more than a decade of armtwisting I. Németi and H. Andréka finally persuaded the author to publish TM#138 itself on the ground that it contained a substantial body of interesting material that for the sake of brevity had been deleted from the STOC-80 version. The principal changes here to TM#138 are the addition of footnotes and the sections at the end on retrospective citations and reflections.

*This research was supported by the National Science Foundation under NSF grant no. MCS78-04338. Preparation of the present version was supported by the National Science Foundation under NSF grant number CCR-8814921.

1 Introduction

We propose, with Kozen [Koz79c], the notion of a *dynamic algebra*, which integrates an abstract notion of *proposition* with an equally abstract notion of *action*. Just as propositions tend to band together to form *Boolean algebras*, as Halmos [Hal62] puts it, with operations $p \vee q$ and p' , so do actions organize themselves into *regular algebras*, with operations $a \cup b$, $a; b$, and a^* . Analogously to the proposition $p \vee q$ being the *disjunction* of propositions p and q , and p' the *complement* of p , the action $a \cup b$ is the *choice* of actions a or b , $a; b$, or just ab , is the *sequence* a followed by b , and a^* is the *iteration* of a indefinitely often.

Just as $p \vee q$ and p' have natural set theoretic interpretations, namely union and complement, so do $a \cup b$, $a; b$, and a^* have natural interpretations on such concrete kinds of actions as additive functions, binary relations, trajectory sets, languages, and matrices over regular algebras, to name those regular algebras that are suited to dynamic algebra. The section below on examples illustrates this.

It is natural to think of an action as being able to bring about a proposition. We write $\langle a \rangle p$, or just ap , pronounced “ a enables p ,” as the proposition that action a can bring about proposition p . A *dynamic algebra* then is a Boolean algebra $(B, \vee, \wedge, ')$, a regular algebra $(R, \cup, ;, *)$, and the *enables* operation $\diamond : R \times B \rightarrow B$.

Motivation. An important problem in computer science is how to reason about computer programs. The proposals of [Flo67, Hoa69, Sal70, Dij76, Con77, Pra76] are representative of a class of methods (by no means the only class) that may be exemplified by the following.

Let $x:=5$ denote the program for setting the value of the program variable x to 5. Then $\langle x:=5 \rangle x=5$ asserts that setting x to 5 can make x equal to 5, which is necessarily true and so is the same (abstract) proposition as the top element 1 of the Boolean algebra of all such propositions. Thus $\langle x:=5 \rangle x=5$ simplifies to that 1. On the other hand $\langle x:=x+1 \rangle x=5$, again viewed abstractly, is not 1 but rather is the same abstract proposition as $x=4$. (The “1” in “ $x+1$ ” is numeric 1, distinguished by context from Boolean 1.) And $\langle (x:=x-1)^* \rangle x=0$ must be $x \geq 0$, as it is not possible to make an initially negative variable zero by decrementing it indefinitely.

These observations about $x:=5$, $x:=x+1$, etc., depend on the nature of program variables, numbers, arithmetic, assignment, and so on. However there are also more universal observations one can make, at a level that knows nothing about programs that manipulate variables and numbers. For example no program can bring about the truth of *false*; that is, $a0 = 0$. Moreover, a program can bring about $p \vee q$ just when it can bring about p or it can bring about q , i.e.

$a(p \vee q) = ap \vee aq$. One of a or b can bring about p just when either a can bring about p or b can, i.e. $(a \cup b)p = ap \vee bp$. And ab can bring about p just when a can bring about a situation from which b can bring about p , i.e. $(ab)p = a(bp)$. (In full, $\langle a; b \rangle p = \langle a \rangle (\langle b \rangle p)$. We rely on the sorts to indicate whether $;$ or \diamond is intended.)

Suppose now that either p holds, or a can bring about a situation from which a can *eventually* (by being iterated) bring about p . Then a can eventually bring about p . That is, $p \vee aa^*p \leq a^*p$. (We write $p \leq q$ to indicate that p implies q , defined as $p \vee q = q$.) In turn, if a can eventually bring about p , then either p is already the case or a can eventually bring about a situation in which p is not the case but one further iteration of a will bring about p . That is, $a^*p \leq p \vee a^*(p' \wedge ap)$. This is the principle of induction, in a simple Boolean disguise as can be seen by forming the contrapositive and replacing p' by q to yield $q \wedge [a^*](q \supset [a]q) \leq [a^*]q$, where $[a]p, p \supset q$ abbreviate $(ap)'$, $p' \vee q$ respectively. $[a]$ is the *dual* of $\langle a \rangle$, and $[a]p$ asserts that whatever a does, p will hold if and when a halts.

The notion of a *test* program $p?$ is also useful. A test cannot bring about a different situation; moreover $p?$ cannot bring about even the present situation unless p already holds. Thus $p?q = p \wedge q$. Tests are of use in defining certain well-known programming constructs such as **if** p **then** a **else** $b = (p?a) \cup (p'?b)$, and **while** p **do** $a = (p?a)^*p'?$. We will have little to say about tests in this paper.

Outline. The remainder of the paper is as follows. Section 2 supplies the main definitions. Section 3 gives the basic result, that the regular operation $*$ of a dynamic algebra is reflexive transitive closure, or quasiclosure to be precise¹. Section 4 illustrates the abstract concept of dynamic algebra with five concrete examples of dynamic algebras that arise in practice, and also gives some counterexamples. Section 5 gives the main results, that free dynamic algebras are residually Boolean-finite (isomorphic to a subdirect product of dynamic algebras with finite Boolean component), and that free dynamic algebras are residually finite.

Section 6 gives several applications of these results. Using the fact that every finite separable dynamic algebra is isomorphic to a Kripke structure we apply the first part of the main result to show the completeness of the Segerberg axiomatization of propositional dynamic logic [Seg77], which we state in algebraic form as the equality of two varieties. Using the second part of the main result we show that separable dynamic algebras and Kripke structures generate the same variety and so have the same equational theory, both Boolean and regular.

¹Alternatively one may restrict attention to separable dynamic algebras when characterizing the definitional content of the Segerberg axioms, avoiding the awkwardness of “quasi” [Pra80b, Pra90].

Finally we explore a new definition of the notion of regular algebra, a surprisingly controversial class given the importance of some of its instances. Section 7 tidies up some loose ends. Section 8 supplies some algebraic prerequisites as a convenient reference for the reader who may need prompting on some of the definitions; in the text we refer thus⁸ to this section.

For us the most interesting parts of the paper are Theorems 3.4 (* is reflexive transitive closure) and 5.3 (existence of filtrations). We find it surprising that an equational system can define reflexive transitive quasiclosure exactly, that is, with no nonstandard models. While it may not come as a surprise to those familiar with the Fischer-Ladner filtration result [FL79] that it can be obtained in an algebraic form, those familiar with the various efforts to obtain the completeness result may find it of interest that the filtration result itself need be the only subtle part. Lemma 3.1 is the key to the rest of the paper, filtration included, but it is not a difficult lemma.

2. DEFINITIONS

Syntax. We define the following classes of algebras.

<i>Class</i>	<i>Symbol</i>	<i>Similarity Type</i>
Boolean algebras	\mathcal{B}	$(B \vee ' 0)$
Regular algebras	\mathcal{R}	$(R \cup ; *)$
Dynamic algebras	\mathcal{D}	$(\mathcal{B} \mathcal{R} \diamond)$
Test algebras	\mathcal{E}	$(\mathcal{D} ?)$

The types are:

$$\begin{array}{lll} \vee : B \times B \rightarrow B & ' : B \rightarrow B & 0 : B \\ \cup : R \times R \rightarrow R & ; : R \times R \rightarrow R & * : R \rightarrow R \\ \diamond : R \times B \rightarrow B & ? : B \rightarrow R & \end{array}$$

We write $p \wedge q$ for $(p' \vee q')'$, 1 for $0'$, $[a]p$ for $(ap)'$, $p \leq q$ for $p \vee q = q$, $a \leq b$ for $\forall p(ap \leq bp)$.

Semantics. A Boolean algebra is a complemented distributive lattice, properties all definable equationally. A dynamic algebra $(\mathcal{B} \mathcal{R} \diamond)$ satisfies the following equations.

1. \mathcal{B} is a Boolean algebra.
- 2a. $a0 = 0$
- 2b. $a(p \vee q) = ap \vee aq$
3. $(a \cup b)p = ap \vee bp$
4. $(ab)p = a(bp)$
- 5a,b. $p \vee aa^*p \leq a^*p \leq p \vee a^*(p' \wedge ap)$

These axioms are obtained from Segerberg's Hilbert-style axioms [Seg77] for propositional dynamic logic [FL79] in the same way one may obtain Boolean identities from a Hilbert-style axiomatization of propositional calculus. Note that axiom 2a is obtained from the modal logic inference rule of Necessitation, namely from p infer $[a]p$. We have discussed the motivation for all of these axioms in the previous section.

A test algebra satisfies:

$$6. \quad p?q = p \wedge q.$$

We will not prove anything about test algebras in this paper. Although they introduce a little complication into some of the arguments, the reader should find it straightforward to extend most of the results below about dynamic algebras to test algebras². We mention *converse* among the open problems of Section 7.

Predynamic Algebras. A *predynamic* algebra is any algebra similar to (having the same similarity type as, but not necessarily satisfying the equations for) a dynamic algebra. Free predynamic algebras, i.e. term or word algebras, are used in this paper in the definition of FL-set.

Separability. If $ap = bp$ for all p we call a and b *inseparable* and write $a \cong b$, an equivalence relation which we shall later show to be a congruence relation on dynamic algebras. Following Kozen [Koz79c] we call *separable* any dynamic algebra in which inseparability is the identity relation. We let SDA denote the class of separable dynamic algebras.

Separability can be expressed with the first-order sentence $\forall a \forall b \exists p (ap = bp \rightarrow a = b)$ (i.e. $\forall a \forall b (\forall p [ap = bp] \rightarrow a = b)$). This being a Horn sentence [Coh65, p235], it follows that SDA is closed under direct products [Hor51]. Example 7 shows that SDA is closed under neither homomorphisms nor subalgebras.

3. BASIC RESULT

In this section we prove the fundamental theorem of dynamic algebra. We present this result before giving the examples, partly because it helps somewhat in understanding the examples, partly because its proof does not deserve to be buried deeper in the paper.

Actions as Functions. Although we have taken the type of \diamond to be $\diamond : R \times B \rightarrow B$ we could equivalently have taken it as $\diamond : R \rightarrow (B \rightarrow B)$; the reason for the former was so that it would be clear that we were defining an ordinary algebra. The latter type is consistent with our use of the notation

²A subsequent detailed treatment of test by Némethi showed that this assessment was naively optimistic

$\langle a \rangle p$ —we may think of $\langle a \rangle$ as a function on B . Our use of the alternative notation ap is to suggest that we may think of a itself as a function on B . All that is lacking is *extensionality*; that is, we may have $ap = bp$ for all $p \in B$ yet not have $a = b$. The lack of extensionality does not prevent us from appearing to be able to define operations such as composition by saying that $(ab)p = a(bp)$ for all p .

Accordingly we shall think of the elements of R as *quasifunctions*, having all the attributes of functions save extensionality. In a separable dynamic algebra they become functions.

Recall that $a \leq b$ means that $ap \leq bp$ for all p . It follows that \leq on quasifunctions is reflexive and transitive but not antisymmetric, and so is a quasiorder. In a separable dynamic algebra it becomes a partial order.

The content of axioms 2a,b is now clear. Axiom 2a says that all quasifunctions are *strict* (0-preserving), and axiom 2b that they are *finitely additive* (preserving joins of finite non-empty sets).

Continuing in this vein, axioms 3 and 4 leave no doubt that \cup is pointwise disjunction and $;$ is composition. In the absence of extensionality we must consider these apparent functionals to be *quasifunctionals*, which are operations on quasifunctions with which the relation of inseparability is compatible ($a \cong a'$ and $b \cong b'$ implies $a \cup b \cong a' \cup b'$ etc.), easily seen from axioms 3 and 4 to be the case for \cup and $;$.

Axiom 5 however is nothing short of inscrutable. It may be made a little more symmetric by rephrasing it as $p \vee (p' \wedge aa^*p) \leq a^*p \leq p \vee a^*(p' \wedge ap)$, using Boolean manipulation on 5a. This can then be broken up into $p \leq a^*p$ together with the even more symmetric $p' \wedge aa^*p \leq p' \wedge a^*p \leq a^*(p' \wedge ap)$. The lower and upper bounds on $p' \wedge a^*p$ seem to be referring to the start and end of the “interval” during which p remains false, an interval which must exist when $p' \wedge a^*p$ holds. This intuitive analysis, while suggestive, is however not a characterization of $*$. The following supplies a more satisfactory formal analysis.

Let $a!p = \{q \mid p \vee aq \leq q\}$. Let $\min S$ be the least element of the partially ordered set S when it exists, and undefined otherwise. (This is in contrast to the infimum $\bigwedge S$ of S , which may exist but not be in S .)

We propose the following alternative to 5a,b.

$$5'. \quad a^*p = \min(a!p).$$

Axiom 5' is not an acceptable equational identity for the purpose of defining a variety because of its use of \min and $!$. Nevertheless it exactly characterizes $*$, as the following lemma shows.

Lemma 3.1. 5a,b and 5' are interchangeable as axioms.

Proof: (\rightarrow). Assume 5a,b. 5a asserts that $a^*p \in a!p$. Now consider arbitrary $q \in a!p$. We show that $a^*p \leq q$. We have:

$$\begin{aligned} p &\leq q && (p \vee aq \leq q) \\ \text{Hence } a^*p &\leq a^*q && (\text{isotonicity, from 2b}) \\ &\leq q \vee a^*(q' \wedge aq) && (5b) \\ &= q \vee a^*0 && (p \vee aq \leq q) \\ &= q. && (2a, 1) \end{aligned}$$

(\leftarrow). Assume $a^*p = \min(a!p)$. Then $a^*p \in a!p$, so 5a holds. For 5b it suffices to show that $p \vee a^*(p' \wedge ap) \in a!p$, since $a^*p \leq q$ for any $q \in a!p$.

$$\begin{aligned} p \vee a(p \vee a^*(p' \wedge ap)) &= p \vee (p' \wedge a(p \vee a^*(p' \wedge ap))) && (1) \\ &= p \vee (p' \wedge (ap \vee aa^*(p' \wedge ap))) && (2b) \\ &\leq p \vee (p' \wedge ap \vee aa^*(p' \wedge ap)) && (1) \\ &\leq p \vee a^*(p' \wedge ap) && (5' \rightarrow 5a) \end{aligned}$$

■

Notice that we used only isotonicity of a , namely $p \leq q$ implies $ap \leq aq$, in the \rightarrow direction, not additivity. If we relax axiom 2b to require only isotonicity we get *isodynamic algebras*, which we shall consider in Section 6 in discussing regular algebras. In Section 6 we show that isotonicity is inadequate for the \leftarrow direction.

From Lemma 3.1 we infer that $*$ is a quasifunctional, since if $a \cong a'$ then $a^*p = \min(a!p) \cong \min(a'!p) = a'^*p$, so $a^* \cong a'^*$. Thus \cong is a congruence relation. We now address the question of characterizing which quasifunctional $*$ is. We define a quasiclosure operator to be as for a closure operator⁸, except that idempotence is replaced by *quasi-idempotence*, $fx \leq ffx$ and $ffx \leq fx$, which for regular elements means $fa \cong ffa$ where \cong is inseparability.

Lemma 3.2. $*$ is a quasiclosure operator.

Proof:

(Isotonicity.) If $a \leq b$ then for all p , $b!p \subseteq a!p$, whence $\min(a!p) \leq \min(b!p)$, thus $a^*p \leq b^*p$, whence $a^* \leq b^*$.

(Reflexivity.) $p \leq a^*p$, so $ap \leq aa^*p \leq a^*p$, for all p , whence $a \leq a^*$.

(Quasi-idempotence.) $a^*p = \min(a!a^*p) = a^*a^*p$, so $a^*p \in a^*!p$. But if $q \in a^*!p$ then $p \leq q$, so $a^*p \leq a^*q \leq q$, whence $a^*p = \min(a^*!p) = a^{**}p$. ■

We call the quasiclosure system associated with $*$ the *system of asterates*, the word “asterate” being Conway’s [Con71, p25]. By the definition of a closure system⁸, an asterate is a fixed point of $*$.

There are of course many quasiclosure operators, and merely being one is not a remarkable thing in a variety. So which quasiclosure operator is this? We say that the quasifunction a is *reflexive* when $p \leq ap$ for all p , and *transitive* when $aa \leq a$. Thus a is reflexive and transitive when for all p , $p \vee aap \leq ap$, i.e. $ap \in a!p$, the characterization we use in the next proof.

Lemma 3.3. a is an asterate iff a is reflexive and transitive.

Proof: $(\rightarrow) ap = a^*p \in a!p$.

$(\leftarrow) a^*p = \min(a!p) \leq ap$, and $ap \leq a^*p$, so $a^*p = ap$. ■

Thus the system of asterates coincides with the set of reflexive transitive quasifunctions, making $*$ reflexive transitive quasiclosure. From all this we infer the following “representation theorem” for dynamic algebras.

Theorem 3.4. Every dynamic algebra is a Boolean algebra \mathcal{B} together with a set of strict finitely additive quasifunctions on \mathcal{B} closed under the quasifunctionals of pointwise disjunction, composition, and reflexive transitive quasiclosure.

Note that when a is reflexive, $aap = ap$ iff $a^*p = ap$, for all p .

This theorem is very helpful in reasoning about dynamic algebras. It does not however make as satisfactory a connection as does Theorem 6.2 with the intuitions of computer scientists, which tend to be oriented towards the notion of *state* as providing a “place” for predicates to hold in, and for programs to travel between. Example 2 below, Kripke structures, amplifies this intuition.

4. EXAMPLES

This section is meant to be suggestive rather than encyclopedic, and is kept short by omitting proofs and lengthy explanations³. The reader will however find the following lemmas helpful in understanding Example 1.

Lemma 4.1. The set $a!p$ is closed under arbitrary meets when they exist.

Proof: Let $S \subseteq a!p$, and suppose $\bigwedge S$ exists. Then for any $r \in S$, $\bigwedge S \leq r$ so $p \vee a(\bigwedge S) \leq p \vee ar \leq r$. Hence $p \vee a(\bigwedge S) \leq \bigwedge S$ ($\bigwedge S$ being the *greatest* lower

³The equational theories of most of these examples are the same [Pra79a].

bound on S), so $\bigwedge S \in a!p$. ■

Corollary 4.2. In a complete lattice (hence in a complete Boolean algebra) $\min(a!p)$ always exists and is $\bigwedge(a!p)$.

Let X be a set of subsets of a lattice \mathcal{B} , and write aS for $\{as \mid s \in S\}$ for any S in X . We call the quasifunction a *X-additive* when $a(\bigvee S) = \bigvee(aS)$ for all S in X for which $\bigvee S$ exists. We call a respectively *strict*; *isotone*; *finitely additive*; *continuous*; and *completely additive*; when a is X -additive for X respectively $\{\emptyset\}$; the set of nonempty finite chains of \mathcal{B} ; the set of nonempty finite subsets of \mathcal{B} ; the set of directed sets of \mathcal{B} ; and the power set $2^{\mathcal{B}}$ of all subsets of \mathcal{B} .

Lemma 4.3. The three quasifunctionals \cup ; $;$ $*$ on quasifunctions on a complete lattice preserve X -additivity for any X .

Proof: Let $S \in X$. For \cup we have $(a \cup b)(\bigvee S) = a(\bigvee S) \vee b(\bigvee S) = \bigvee(aS) \vee \bigvee(bS) = \bigvee((a \cup b)S)$. For $;$ we have $(ab)(\bigvee S) = a(b(\bigvee S)) = a(\bigvee(bS)) = \bigvee(a(bS)) = \bigvee((ab)S)$. For $*$ we have $(\bigvee S) \vee a(\bigvee(a^*S)) = (\bigvee S) \vee (\bigvee(aa^*S)) = \bigvee(S \cup aa^*S) \leq \bigvee(a^*S)$, so $\bigvee(a^*S) \in a!(\bigvee S)$. For any $r \in a!(\bigvee S)$ we argue as follows. Let $s \in S$, so $s \leq \bigvee S$, so $a!(\bigvee S) \subseteq a!s$, whence $a^*s = \min(a!s) \leq r$ ($\min(a!s)$ exists by Corollary 4.2). So $\bigvee(a^*S) \leq r$. Hence $\bigvee(a^*S) = \min(a!(\bigvee S)) = a^*(\bigvee S)$. ■

Thus if a and b are strict, so are $a \cup b$, ab , and a^* . The same holds with “strict” replaced by “isotone,” “finitely additive,” “continuous,” or “completely additive.”

Example 1: Full dynamic algebras. Given a complete Boolean algebra $\mathcal{B} = (B, \vee, \wedge, ' , 0)$, let R be the set of all strict finitely (resp. completely) additive functions on B and let $\diamond : R \times B \rightarrow B$ be application of elements of R to elements of B . By Lemma 4.3, R is closed under \cup , $;$, and $*$ when assigned the interpretations of Theorem 3.4. Hence by that theorem $(\mathcal{B} (R \cup ; *) \diamond)$ is a dynamic algebra. We call it the full (resp. completely full) dynamic algebra on \mathcal{B} . Note that the completely full algebra has fewer functions than the full one. Both are of course separable.

The class of full dynamic algebras contains some pathological cases, as Example 6 will show. Examples 2-5 however give algebras encountered in ordinary practice.

Example 2: Kripke structures. Given a set W (the set of *possible worlds*, or *states*), let \mathcal{B} be the power set algebra on W . Then the *full Kripke structure on W* is the completely full dynamic algebra on \mathcal{B} , a separable dynamic algebra. A *Kripke structure on W* is a subalgebra of the full Kripke structure on W .

The class KRI consists of all Kripke structures. The class of dynamic algebras, being a variety, is closed under taking subalgebras, whence Kripke structures are dynamic algebras. They are not however necessarily separable ones since SDA is not closed under subalgebras, as Example 7 will show.

Completely additive functions on the complete Boolean algebra of subsets of W correspond to binary relations on W . This is because the set of functions $f : W \rightarrow 2^W$ (binary relations) naturally corresponds to the completely additive subset of the set of functions $f : 2^W \rightarrow 2^W$. The correspondence is as follows. If $f : W \rightarrow 2^W$ is a binary relation, the corresponding $g : 2^W \rightarrow 2^W$ satisfies $g(U) = \cup\{f(u) \mid u \in U\}$. Conversely, any completely additive $g : 2^W \rightarrow 2^W$ must satisfy $g(U) = \cup\{g(u) \mid u \in U\}$, by complete additivity, so the restriction of g to the atoms of 2^W corresponds naturally to a function $f : W \rightarrow 2^W$.

From this it should be clear that the operations of pointwise disjunction and composition correspond exactly to those of union and composition for binary relations. Reflexive transitive closure for functions on lattices similarly corresponds to reflexive transitive closure for binary relations since the definition of reflexive transitive closure depends only on \cup and $;$ and these have already been shown to correspond. Note that we need to start from the full power set algebra to avoid omitting any closures and getting the wrong reflexive transitive closure; “smaller” Kripke structures are obtained as subalgebras *after* $*$ is defined.

Because of this correspondence we will *define* binary relations on W to be completely additive functions on 2^W .

Kripke structures supply quite satisfactory models of programs. The elements of the sets constituting the Boolean part of a Kripke structure model the states of one or more computers. Each Boolean element is then a predicate on states; for example the formula $x \geq 0$ denotes the set of states in which the variable x is non-negative. The regular elements correspond to binary relations on the set of states. The relations in turn correspond to the edges of a graph whose vertices are the states. Each edge is labelled with a program. The whole graph then presents a picture of all the possible states of a system together with all the possible state transitions, each labelled with its agent.

The graph enjoys certain closure properties. For example there exists between two states a transition labelled either a or b just when there exists between those states a transition labelled $a \cup b$. There exists a path between two states consisting of transition a (to some state) followed by transition b just when there exists a transition ab between those two states. There exists a (possibly trivial) finite path of a 's between two states just when there exists a transition a^* between those states (whence for every state and every program a there is a transition a^* leaving and returning to that state).

Because of the importance of Kripke structures in computer science, the

question arises as to whether the equational theory of dynamic algebras does justice to Kripke structures. The next section lays the groundwork for showing in the section after it that the classes KRI and SDA (Kripke structures and separable dynamic algebras) generate the same variety and so have the same equational theory. Both the Boolean and regular theory are of considerable importance in reasoning about programs, the former dealing with program correctness and termination, the latter with program equivalence.

Example 3: Trajectory algebras. These constitute a variation on Kripke structures in which \mathcal{B} is as before, while the completely additive functions in R are replaced by sets of non-empty strings over the set W of states. The regular operations are as follows. \cup is set union, $;$ is “fusion product,” in which $ab = \{u \dots v \dots w \mid u \dots v \in a \text{ and } v \dots w \in b\}$, and $a^* = W \cup a \cup aa \cup aaa \cup \dots$ (Fusion product differs from concatenation in that there is a requirement of “compatibility” between the strings being “fused.” Thus the fusion product of $\{ab, cd\}$ with $\{de, bc\}$ is $\{abc, cde\}$, while the concatenation is $\{abde, abbc, cdde, cdbc\}$.) Finally $ap = \{u \mid u \dots v \in a \text{ and } v \in p\}$.

In a Kripke structure a program can be considered a set of pairs of states, each pair having an initial and a final state. In a trajectory algebra a program is a set of state trajectories, each trajectory having an initial state, intermediate states, and a final state.

Trajectory algebras supply a natural example of a nonseparable dynamic algebra. This is because two sets of trajectories may differ only in their intermediate states, and hence exhibit the same functional behavior on \mathcal{B} .

Other modalities besides \diamond suggest themselves for trajectory algebras, such as “throughout” and “sometime during.” An axiomatization of such modalities appears in [Pra79b].

Example 4: Linguistic Algebras. Let Σ^* denote the set of finite strings over some alphabet Σ . A *language* is a subset of Σ^* . Recall that a field of sets is a set of sets closed under union and complementation relative to its union. We define a *regular algebra of languages* to be a set of languages closed under union, concatenation, and reflexive transitive closure. A *finitary-linguistic dynamic algebra* over a given alphabet is an algebra consisting of a field of languages (with complementation being relative to Σ^*), a regular algebra of languages, and the operation of concatenation, all languages being over the given alphabet. Every finitary-linguistic dynamic algebra is a dynamic algebra.

In connection with programs Σ may be considered to be the vocabulary of commands the computer can issue. A program is then a set of command sequences. There seems to be no natural interpretation of propositions in this setting.

Σ^ω denotes the set of infinite-to-the-right strings over Σ . A *linguistic dynamic algebra* differs from a finitary linguistic dynamic algebra in that \mathcal{B} is a set of *infinitary* languages (subsets of $\Sigma^* \cup \Sigma^\omega$). Every linguistic dynamic algebra is a dynamic algebra.

With some thought the reader may verify that $a^*(ap \wedge ap')' = 1$ is an identity of finitary-linguistic dynamic algebras but not of linguistic dynamic algebras, and hence not of dynamic algebras.

Every finite Kripke structure is a homomorphic image of a linguistic dynamic algebra (see [Pra79a] for a proof). It follows from this and the results below that the Boolean equational theory of linguistic dynamic algebras coincides with that of dynamic algebras.

An application of linguistic dynamic algebras is to Pnueli's tense-logic treatment of non-terminating processes [Pnu77]. Though no semantics is proposed in [Pnu77], the subsequent development of temporal logic has been based substantially on propositions as a set of sequences.

Example 5: Flowchart Algebras. Let $\mathcal{D} = (\mathcal{B} \ \mathcal{R} \ \diamond)$ be a separable dynamic algebra, and let V be a finite set of *dimensions*. Take \mathcal{B}' to be the direct power \mathcal{B}^V , whose elements are Boolean vectors that combine pointwise under \vee and $'$, using the corresponding operations of \mathcal{B} .

A *regular matrix* is a function $a : V^2 \rightarrow R$. Regular matrices combine pointwise under \cup using the corresponding operation of \mathcal{R} . They are multiplied in the usual way for matrices, taking \cup and $;$ from \mathcal{R} as “addition” and “multiplication” respectively. When a is a 1×1 matrix ($|V| = 1$) with sole element $b \in R$, a^* is the 1×1 matrix with sole element b^* . When V can be partitioned into two non-empty subsets V_1, V_2 , inducing a corresponding partition of a as the block matrix

$$\begin{pmatrix} A & B \\ D & C \end{pmatrix},$$

then, following Conway [Con71], a^* is taken to be the matrix

$$\begin{pmatrix} (A^*BC^*D)^*A^* & (A^*BC^*D)^*A^*BC^* \\ (C^*DA^*B)^*C^*DA^* & (C^*DA^*B)^*C^* \end{pmatrix}.$$

Separability is used to ensure that this definition of a^* does not depend on the choice of partition of V .

A set of such matrices closed under these three operations forms an algebra \mathcal{R}' . The operation \diamond' multiplies matrices by vectors in the usual way, with the

\vee of \mathcal{B} as “addition” and the \diamond of \mathcal{D} as “multiplication.” The algebra $(\mathcal{B}' \mathcal{R}' \diamond')$ we call a *flowchart algebra*.

Every flowchart algebra is a dynamic algebra.

Flowchart algebras supply a natural solution to the problem of treating programs algebraically. The traditional method is via “elimination of goto’s,” using various program transformations that non-trivially manipulate the structure of the original program. Flowchart algebras permit an algebraic treatment of the original “goto-laden” program. The set V supplies vertices for a graph; each Boolean vector gives a possible labelling of the vertices with facts from \mathcal{B} , while each matrix defines a labelling of the edges of the graph with programs from \mathcal{R} .

In this formulation the vertex labels are the complements of the formulas used in the programming milieu to annotate flowcharts as described in [Flo67], the complement being attributable to our use of \diamond as “possible” where programming custom calls for the dual “necessary.” Axiom 5b in this setting amounts to the so-called “Floyd induction” principle; the reader familiar with [Flo67] will find the axiom more recognizable if the contrapositive is taken and p replaced by p' (as just mentioned) to yield $p \wedge [a^*](p \supset [a]p) \leq [a^*]p$, where $[a]p$ abbreviates $(ap)'$ and $p \supset q$ abbreviates $p' \vee q$.

Example 6: A non-standard dynamic algebra. This example shows that there exist dynamic algebras in which $a^*p = \bigvee\{a^i p \mid i \geq 0\}$ fails. Let \mathcal{B} be the power set algebra of $\mathbf{N} \cup \{\infty\} = \{0, 1, 2, \dots, \infty\}$ and let \mathcal{D} be the full dynamic algebra on \mathcal{B} , a specialization of Example 1. Now let a be the function mapping $p \subseteq \mathbf{N} \cup \{\infty\}$ to $\{n + 1 \mid n \in p\} \cup \{\infty \mid p \text{ is infinite}\}$. That is, $a(p)$ forms the set of successors of the elements of p (with $\infty + 1$ defined as ∞) together with ∞ when p is infinite. Observe that a is strict and finitely additive but not continuous and so not completely additive. Then $\bigvee\{a^i 0 \mid i \geq 0\} = \mathbf{N}$, while $\min(a!\{0\}) = \mathbf{N} \cup \{\infty\}$.

Kozen [Koz79c] calls a dynamic algebra **-continuous* when it satisfies $a^*p = \bigvee\{a^i p \mid i \geq 0\}$. His definition of a dynamic algebra includes *-continuity as a requirement, along with conditions on \mathcal{R} axiomatizing it as a regular *-continuous algebra, in place of Segerberg’s 5a,b (which then becomes a theorem). Let us call this class CDA (for *-continuous DA). Example 6 contradicts $\text{SDA} \subseteq \text{CDA}$.

It can be shown that $\text{KRI} \subseteq \text{CDA}$, whence not all dynamic algebras are isomorphic to Kripke structures. Kozen has asked whether every *-continuous dynamic algebra is isomorphic to a Kripke structure.

One might ask whether CDA forms a variety. Now $\text{SDA} \subseteq \text{HSP}(\text{KRI})$ (the variety generated⁸ by KRI), as we show later. But since $\text{KRI} \subseteq \text{CDA}$, $\text{HSP}(\text{KRI}) \subseteq \text{HSP}(\text{CDA})$, whence $\text{SDA} \subseteq \text{HSP}(\text{CDA})$. Example 6 is in SDA but not in CDA, whence CDA is not a variety.

The last two examples are not examples of dynamic algebras so much as examples of their behavior under various operations.

Example 7: Loss of separability under homomorphisms and subalgebras. Let $\mathcal{D} = ((\{0, P, P', 1\} \vee ' 0) (\{A, B\} \cup ; *) \diamond)$ such that $BP = 1$, $Ap = Bp = p$ otherwise, and such that \mathcal{D} is a dynamic algebra (whence $\cup ; *$ are determined, \mathcal{D} being separable by construction.) Let $h : \mathcal{D} \rightarrow \mathcal{D}$ satisfy $h(P') = h(0) = 0$, $h(P) = h(1) = 1$, $h(A) = A$, $h(B) = B$, a homomorphism as the reader may verify. The homomorphic image $h(\mathcal{D})$ is not separable. Furthermore $h(\mathcal{D})$ is also a subalgebra of \mathcal{D} . Hence SDA is closed under neither homomorphisms nor subalgebras. That separability is expressible with a Horn sentence ensures that SDA is closed under direct products.

Example 8: Effect of homomorphisms on the regular component of a separable dynamic algebra. It may have occurred to the reader that the class of regular components of separable dynamic algebras might make a good candidate for the class of regular algebras, whatever they might be. This is discussed in more detail in the section on applications. Here we show non-preservation of this class under homomorphisms.

Let \mathcal{B} be the power set algebra on the set of all finite strings on an arbitrary alphabet. Let \mathcal{R} consist of those elements of \mathcal{B} containing λ (the empty string), with $\cup ; *$ having their standard interpretations on languages. Let \diamond be concatenation. Then $(\mathcal{B} \mathcal{R} \diamond)$ is a dynamic algebra, separable because $\{\lambda\} \in \mathcal{B}$. Let \mathcal{R}' be $(\{I, A, A^*\} \cup ; *)$ with $I \leq A \leq A^*$, asterates I, A^* , I acting as multiplicative identity, $AA = A$, all remaining products equal to A^* . This is not the regular component of any dynamic algebra because A is reflexive and transitive but not an asterate, contradicting Lemma 3.3. Define $h : \mathcal{R} \rightarrow \mathcal{R}'$ so that $\{\lambda\}$ goes to I , infinite sets to A^* , finite sets to A , visibly a homomorphism. This establishes that the class of regular components of separable dynamic algebras is not preserved under homomorphisms. The example is Conway's [Con71, p102], minus the element 0.

5. MAIN RESULT

The main result has two parts. The first part states that every free dynamic algebra is residually Boolean-finite. The second part states that every free separable dynamic algebra is residually separable-and-finite. The first part is adequate for showing completeness of the Segerberg axiomatization of propositional dynamic logic, and the reader wishing to see only that result may skip the second part. The first part is however inadequate for the next application, that every free separable dynamic algebra is residually KRI (isomorphic to a subdirect product of Kripke structures), whence every separable dynamic algebra is a homomorphic image of such a subdirect product. This is a useful representation theorem for dynamic algebras, though not as strong as showing

that every dynamic algebra is isomorphic to a Kripke structure, which Example 6 showed to be false for our notion of dynamic algebra.

We approach the main result via an abstract version of the modal logic technique of *filtration*, which in a Kripke structure setting is the process of dividing a Kripke model of a given formula p by an equivalence relation on its worlds to yield a finite Kripke model of p . Fischer and Ladner [FL79] showed that filtration could be made to work for propositional dynamic logic just as well as for modal logic. We extend their result to show that filtration does not depend on any special properties of Kripke structures but works for all dynamic algebras, even ones that are not $*$ -continuous in Kozen's sense [Koz79c]. Our proof is little more than the abstract version of that of [FL79]. We attend first to some prerequisites.

The reader may wish to look at the account of generator sets and free algebras in Section 8. We let $P, Q, \dots, A, B, \dots, X, Y, \dots$ range over the set of generators in $B, R, B \cup R$ respectively, and write B_0, R_0, D_0 for the respective generator sets.

Lemma 5.1. The regular component of any subalgebra of a dynamic algebra $\mathcal{D} = (\mathcal{B} \ \mathcal{R} \ \diamond)$ containing the regular generators of a generator set of \mathcal{D} coincides with \mathcal{R} .

Proof: Left to the reader. ■

The Boolean analogue of this lemma is false; the Boolean generators generate only part of the Boolean component (consider AP etc.).

FL-sets. An FL-set is a Boolean subset F of a predynamic algebra such that

$$\begin{array}{lcl} p \vee q \in F & \rightarrow & p, q \in F \\ p' \in F & \rightarrow & p \in F \\ ap \in F & \rightarrow & p \in F \\ (a \cup b)p \in F & \rightarrow & ap, bp \in F \\ (ab)p \in F & \rightarrow & a(bp) \in F \\ a^*p \in F & \rightarrow & p, aa^*p \in F \end{array}$$

These rules form a generative system with source set⁸ the Boolean elements of a free predynamic algebra, i.e. terms; we call the associated closure operator *FL-closure*, $FL(X)$.

Lemma 5.2. (Fischer-Ladner [FL79].) The FL-closure of a finite Boolean subset of a free predynamic algebra is finite.

See [FL79] for a proof.

Filtration. We are now in a position to state and prove the central theorem of this paper, which asserts the existence of filtrations.

Theorem 5.3. Given an SDA-free⁸ dynamic algebra $\mathcal{D} = (\mathcal{B} \ \mathcal{R} \ \diamond)$ and a finite subset B_g of B , there exists a dynamic algebra \mathcal{D}' and a homomorphism $f : \mathcal{D} \rightarrow \mathcal{D}'$ injective on B_g , with $f(\mathcal{D})$ finite and separable.

We call f a *filtration*, and $f(\mathcal{D})$ a *filtrate*, of B_g .

Proof: Our construction of \mathcal{D}' and f from \mathcal{D} and B_g proceeds via a series of steps given by the following diagram.

$$\begin{array}{ccccc}
 \mathcal{D}'' & \xrightarrow{h} & \mathcal{D} & \xrightarrow{f} & \mathcal{D}' \\
 & & \cup & & \nearrow \\
 & & \mathcal{B}_x & & \\
 & & \cup & & \\
 \mathcal{B}_F'' & \xrightarrow{h} & \mathcal{B}' & & \\
 \uparrow G & & \cup & & \uparrow G \\
 B_f'' & \xrightarrow{h} & B_f & & \\
 \uparrow FL & & \cup & & \\
 B_g'' & \xleftarrow{c} & B_g & &
 \end{array}$$

The components of the diagram are as follows. The arrows trace the order of construction, except for the top two arrows which represent homomorphisms. \mathcal{D} and B_g are given. \mathcal{D}'' is a free predynamic algebra generated by the generators of \mathcal{D} and $h : \mathcal{D}'' \rightarrow \mathcal{D}$ is the homomorphism fixing those generators (whence h is onto). B_g'' is $c(B_g)$ where $c : B \rightarrow B''$ is a choice function satisfying $hc(p) = p$; c exists by the Axiom of Choice and because h is onto. (This use of choice can be eliminated at the cost of a little more complication.) $B_f'' = FL(B_g'')$, finite by Lemma 5.2, and $\mathcal{B}_F'' = G(B_f'')$, the subalgebra generated by B_f'' . $B_f = h(B_f'')$, finite because B_f'' is, and $\mathcal{B}' = h(\mathcal{B}_F'') = h(G(B_f'')) = {}^8 G(h(B_f''))$, finite because finitely generated Boolean algebras are finite. \mathcal{D}' is the full, hence completely full, dynamic algebra on \mathcal{B}' (see Example 1 above; thus \mathcal{D}' is separable and finite). Since \mathcal{B}' is a finite and hence complete sublattice of \mathcal{B} it defines a closure operator⁸ J on \mathcal{B} . The homomorphism f agrees with J on B_0 (the Boolean generators of \mathcal{D}) and maps each regular generator A to “ JA ”, the function on B' that takes p to $J(Ap)$, which the reader may verify is

strict and finitely additive and so in R' . Such an f exists since \mathcal{D} is SDA-free. \mathcal{B}_x is the set of all fixed points of f , a Boolean subalgebra of \mathcal{B} as it happens.

The one inclusion shown in the diagram that requires verification is $\mathcal{B}' \subseteq \mathcal{B}_x$.

Claim (i). For all $a \in R''$ and $p \in B'$, $fh(a)p \geq h(a)p$.

Claim (ii). For all $ar \in B_F''$, $fh(a)h(r) = h(a)h(r)$.

We prove these claims by induction on R'' , proving (i) explicitly. For (ii) replace \geq by $=$ and p by $h(r)$ uniformly in the proof of (i). (We need $h(r)$ rather than p in (ii) to make use of B_F'' being an FL -set.) We write α, β for $h(a), h(b)$. Lemma 5.1 justifies confining the induction to R'' .

$$\begin{aligned}
fh(A)p &= \bigwedge \{q \in B' \mid q \geq h(A)p\} \\
&\geq h(A)p. \\
fh(a \cup b)p &= (f(\alpha) \cup f(\beta))p \\
&= f(\alpha)p \vee f(\beta)p \\
&\geq \alpha p \vee \beta p && \text{(Note1)} \\
&= h(a \cup b)p. \\
fh(ab)p &= f(\alpha)f(\beta)p \\
&\geq \alpha\beta p && \text{(Note2)} \\
&= h(ab)p. \\
fh(a^*)p &= f(\alpha)^*p \\
&= \min(f(\alpha)!_{B'}p) \\
&\geq \min(\alpha!_{B'}p) && \text{(Note3)} \\
&\geq \min(\alpha!_B p) && \text{(Note4)} \\
&= h(a^*)p.
\end{aligned}$$

Notes. (Each note applies to claim (i) or (ii) or both as indicated.)

1. (ii) $(a \cup b)r \in B_F''$ implies $ar, br \in B_F''$, B_F'' being FL -closed.

2. (ii) Similarly $(ab)r \in B_F''$ implies $a(br), br \in B_F''$.

3. (i) Since $f(\alpha)q \geq \alpha q$ for all q by induction, $f(\alpha)!_{B'}p \subseteq \alpha!_{B'}p$.

(ii) $a^*r \in B_F''$ implies $a(a^*r) \in B_F''$, so $f(\alpha)\alpha^*p = \alpha\alpha^*p$ by induction, so $\alpha^*p \in f(\alpha)!_{B'}p$, so $\min(f(\alpha)!_{B'}p) \leq \alpha^*p = \min(\alpha!_{B'}p)$, with equality then following by 3(i).

4. (i) Since $B' \subseteq B$, $\alpha!_{B'}p \subseteq \alpha!_B p$.

(ii) Since $\alpha^*p \in B'$, $\alpha^*p \in \alpha!_{B'}p$, so $\min(\alpha!_{B'}p) \leq \alpha^*p = \min(\alpha!_B p)$.

Claim (iii). For all $r \in B_F''$, $fh(r) = h(r)$.

We proceed by structural induction on B_F'' . The first step in the following uses the construction of f on generators, $P \in B_F''$, and that h preserves generators.

$$\begin{aligned}
fh(P) &= h(P) \\
fh(p \vee q) &= fh(p) \vee fh(q) \\
&= h(p) \vee h(q) \\
&= h(p \vee q). \\
fh(p') &= fh(p)' \\
&= h(p)' \\
&= h(p'). \\
fh(ap) &= fh(a)fh(p) \\
&= fh(a)h(p) \\
&= h(a)h(p) \quad (\text{by (ii)}) \\
&= h(ap).
\end{aligned}$$

We infer that $\mathcal{B}' = h(\mathcal{B}_F'') \subseteq \mathcal{B}_x$. This completes the defense of the diagram, establishing that \mathcal{D}' is finite and separable, and f fixes B_g . Clearly $f(\mathcal{D})$ is finite. To see that $f(\mathcal{D})$ is separable it suffices to observe that the Boolean component of $f(\mathcal{D})$ is \mathcal{B}' since $\mathcal{B}' \subseteq \mathcal{B}_x$; in fact $\mathcal{B}' = \mathcal{B}_x$ as it happens. The regular component of $f(\mathcal{D})$ may be smaller than that of \mathcal{D}' , but that will not compromise separability. ■

We now give the first part of the main result. Recall that a dynamic algebra is *Boolean-finite* when its Boolean component is finite.

Theorem 5.4. (Main result, first part.) Every free dynamic algebra \mathcal{D} is residually Boolean-finite.

Proof: Take the separating set of congruences to consist of the kernels of the filtrations of the doubletons of \mathcal{D} , together with the congruence relation that is the complete relation on the Booleans and the identity relation on the regular elements. The filtration kernels separate the Booleans while the other relation separates the regular elements. The corresponding quotients are Boolean-finite, by Theorem 5.3 for the kernels and obviously for the other. ■

The reader may at this point wish to skip the second part of the main result and go the next section, where the Segerberg completeness result is proved without depending on the second part.

Theorem 5.5. (Main result, second part.) Every free separable dynamic algebra is residually separable-and-finite⁴.

⁴In fact every free dynamic algebra is separable, as Némethi [Ném82] subsequently pointed

Proof: Take the kernels of the filtrations of the doubletons of \mathcal{D} as for Theorem 5.4, possible because free separable dynamic algebras are SDA-free. This set separates the Booleans. It also separates $a \neq b$, since by separability there exists p such that $ap \neq bp$, whence $ap \not\cong bp$ for some congruence, so $a \not\cong b$ for that congruence. Hence the kernels form a separating set. The corresponding quotients are separable and finite by Theorem 5.3. ■

For this theorem to be of any use, free separable dynamic algebras must exist. As Example 7 shows, SDA is not a variety and hence not a guaranteed source of free algebras. However $\text{HSP}(\text{SDA})$, the variety generated by SDA, does have free algebras, which we can show supply the necessary free separable dynamic algebras.

Lemma 5.6. Every free $\text{HSP}(\text{SDA})$ -algebra having at least one Boolean generator is separable.

Proof: Let \mathcal{D} be a free $\text{HSP}(\text{SDA})$ -algebra. If $a \neq b$ in \mathcal{D} then there exists \mathcal{D}' in SDA and a homomorphism $h : \mathcal{D} \rightarrow \mathcal{D}'$ which maps a, b to distinct elements⁸. Since \mathcal{D}' is separable there must exist $p \in B'$ such that $h(a)p \neq h(b)p$. If we take $g : \mathcal{D} \rightarrow \mathcal{D}'$ to be a homomorphism agreeing with h on the generators of R and satisfying $g(P) = p$, we have $g(aP) = g(a)g(P) = h(a)p \neq h(b)p = g(b)g(P) = g(bP)$, whence $aP \neq bP$. Hence \mathcal{D} is separable. ■

We do not know whether this lemma holds when there are no Boolean generators⁵. Fortunately for our application all we need are free algebras with *at least* a given number of generators.

6. APPLICATIONS

The first application uses the first part of the main result to show the completeness of the Segerberg axiomatization of propositional dynamic logic. Completeness of this system, as with other modal logics, is traditionally measured with respect to Kripke structures. In the program logic application this is because of the satisfactory way in which Kripke structures model computation, as discussed in Example 2. We first prove an easy result about finite Kripke structures.

Theorem 6.1. Every finite separable dynamic algebra is isomorphic to a (finite) Kripke structure.

out. This is an immediate consequence of the initiality of free algebras which my prior three months of exposure to the techniques of algebraic logic had been insufficient to make leap out at me at the time. In this revision I have accordingly dropped “separable” from the abstract’s wording of the main theorem.

⁵Solved affirmatively by I. Nemeti [Ném82].

Proof: Let $\mathcal{D} = (\mathcal{B} \ \mathcal{R} \ \diamond)$ be a separable finite dynamic algebra. Then by the fact that every finite Boolean algebra is isomorphic to the power set algebra of its atoms, and by separability, \mathcal{D} is isomorphic to a subalgebra of the full (hence completely full by finiteness of \mathcal{B}) dynamic algebra on the power set algebra of the atoms of \mathcal{B} , which is by definition a Kripke structure. ■

Let KRI^+ be the class of Kripke structures together with the class of Boolean-trivial dynamic algebras, viz. dynamic algebras with one Boolean element, and let DA be the class of dynamic algebras. Recall that $\text{HSP}(C)$ is the variety generated⁸ by class C .

Theorem 6.2. $\text{DA} = \text{HSP}(\text{KRI}^+)$.⁶

Proof: Certainly $\text{KRI} \subseteq \text{DA}$, and trivially the Boolean-trivial algebras are dynamic algebras, so since DA is a variety, $\text{HSP}(\text{KRI}^+) \subseteq \text{DA}$. Conversely, by Theorems 5.4 and 6.1 every free dynamic algebra is in $\text{SP}(\text{KRI}^+)$, whence every dynamic algebra is in $\text{HSP}(\text{KRI}^+)$, being a homomorphic image of some free dynamic algebra. ■

Although KRI^+ is a bigger class than KRI its Boolean theory cannot be decreased since the Boolean theory of Boolean-trivial algebras must include all Boolean identities holding in KRI . Thus Theorem 6.2 supplies an algebraic form of the Segerberg-Parikh completeness result for propositional dynamic logic. The connection with the Hilbert-style form of Segerberg's axiom system is easily made along the lines one would use to translate identities of Boolean algebra into their corresponding Hilbert-style axioms.

There are two unsatisfactory aspects to Theorem 6.2. First KRI^+ is somewhat artificial compared with KRI . Second the regular theory of KRI^+ is trivial (has only identities $x=x$) since the regular theory of Boolean-trivial dynamic algebras is trivial, whereas the regular theory of KRI is quite rich. The following shows just how rich it is.

Theorem 6.3. The projection of KRI on its regular coordinate (i.e. the class of regular components of Kripke structures) is the class of regular algebras of binary relations.

Proof: The former is certainly included in the latter. Conversely, any regular algebra of binary relations on a set W is the regular component of a subalgebra of the full dynamic algebra on W , a Kripke structure. ■

Hence the regular theory of KRI is the theory of regular algebras of binary

⁶Németi [Ném82] subsequently found the "right" strengthening of this result, to "every free dynamic algebra is representable" (i.e. isomorphic to a Kripke structure).

relations. Thus a connection with KRI would be much more rewarding than the connection with KRI^+ . The axiom of separability supplies exactly that connection inasmuch as it supplies the missing regular theory, as the following shows.

Theorem 6.4. $\text{HSP}(\text{SDA}) = \text{HSP}(\text{KRI})$.

Proof: Since KRI is the closure under subalgebras of full Kripke structures, which are separable, we have $\text{HSP}(\text{KRI}) \subseteq \text{HSP}(\text{SDA})$. Since free algebras in SDA are residually separable-and-finite, they are residually KRI by Theorem 6.1. Furthermore every separable dynamic algebra is a homomorphic image of some free separable dynamic algebra by Lemma 5.6, so $\text{HSP}(\text{SDA}) \subseteq \text{HSP}(\text{KRI})$. ■

Combining Theorems 6.2 and 6.4 we infer that $\text{HSP}(\text{SDA})$ may be defined axiomatically by the dynamic algebra axioms (whose completeness is given by 6.2) together with an appropriate set of axioms for binary relations. There is unfortunately no finite equational axiomatization of the latter [Red64], though the dynamic algebra axioms plus the axiom of separability is only a quantifier away from being one, namely the quantifier of p in the axiom of separability, $\forall p[ap = bp] \rightarrow a = b$. The system of Salomaa [Sal66] comes similarly close, to within a nonstandard inference rule⁷

Regular Algebras. Is there such a thing as a regular algebra? Unlike such satisfactorily defined classes as groups, rings, lattices, Boolean algebras, and even dynamic algebras, regular algebras have an identity problem: there is no agreed-on definition of a regular algebra. Moreover, a monograph by J.H. Conway [Con71] gives some insight as to why.

Conway exhibits five classes of algebras of type $(R \cup ; * 0 1)$, called *X-Kleene algebras* for X ranging over $\mathbf{S}, \mathbf{N}, \mathbf{R}, \mathbf{C}, \mathbf{A}$, in order of strictly increasing size. \mathbf{S} (Standard) consists of complete lattices under \cup in which $;$ is associative and distributes over all joins, and $a^* = \bigvee \{a^i \mid i \geq 0\}$. Furthermore, 0 and 1 are additive and multiplicative identities respectively. \mathbf{N} (Normal) is $\mathbf{S}(\mathbf{S})$ and \mathbf{R} (Regular) is $\text{HSP}(\mathbf{S})$, where $\mathbf{S}(\mathbf{S})$ is the closure of \mathbf{S} under subalgebras and $\text{HSP}(\mathbf{S})$ is the variety generated by \mathbf{S} (closure under subalgebras and homomorphisms— \mathbf{S} is already closed under direct products). \mathbf{C} is a variety strictly larger than $\text{HSP}(\mathbf{S})$ axiomatized by a finite set of axioms together with $a^* = (a^n)^* a^{<n}$, essentially Kleene's axiom schema (11) of 7.2 [Kle56]. \mathbf{A}

⁷Although Salomaa's system is presented as a schema, its form suggests that it should be easy to cast it as a universal Horn theory. Kozen [Koz91] has recently shown a certain lack of substitutivity in Salomaa's axioms that appears to prevent doing this directly. Kozen provides a universal Horn theory that does generate the equational theory of regular algebras, using two conditional axioms rather than the single one that would have been expected on the basis of Salomaa's axiomatization.

(Acyclic) is a finitely axiomatized variety whose significance is not apparent to us; it is Kleene's system less the infinite schema.

Kozen [Koz79c] has proposed a definition of regular algebra, of the same similarity type as Conway's, as a join semilattice in which \cdot is associative and distributes over finite joins and over $\bigvee\{a^i \mid i \geq 0\}$, the latter, defined as a^* , being the only infinite joins required to exist.

Theorem 6.4 suggests that the regular components of separable dynamic algebras might constitute an interesting class of regular algebras. Let $\mathcal{R} = (\{0, 2, 1\} \cup ; *)$ where $0 \leq 2 \leq 1$ (defining \cup), 1 is the only asterate, and \cdot is integer multiplication modulo 4 (so $2 \cdot 2 = 0$). This is Conway's third example of a **S**-algebra on p.101 of [Con71]. However Kozen has pointed out to us that in any separable dynamic algebra, if $a \leq I$ (where $I p = p$ for all p), $aa = a$. Thus \mathcal{R} is not the regular component of any separable dynamic algebra.

It is the case that every regular algebra of binary relations is the regular component of a separable dynamic algebra, namely a subalgebra of a full Kripke structure where all the Boolean elements are retained (to maintain separability). Furthermore every regular algebra of languages on the alphabet Σ is isomorphic to a regular algebra of binary relations; the isomorphism maps the language L to $\{(u, uv) \mid u \in \Sigma^*, v \in L\}$. Hence the regular components of separable dynamic algebras include all regular algebras of binary relations and of languages. Since these are of central importance in the theory of regular algebras we might be forgiven for excluding apparent oddities such as the example immediately above.

To do so however would be to exclude some well-motivated algebras. Kozen has pointed out to us that $(N \min + K0)$ is such an algebra, where $N = \{0, 1, 2, \dots\}$, $\min(9, 5) = 5$ etc., and $K0$ is the constantly zero function. This algebra is of central importance in the computation of minimal cost routes in networks; given a *choice* of routes one wants the cheaper of the two, whence \min ; the cost of a *sequence* of two routes is the sum of their costs, whence $+$; and the cheapest way to travel a route an arbitrary number of times is not to venture forth even once, whence $K0$.

To avoid excluding such algebras we propose to relax axiom 2b of the dynamic algebra conditions to:

$$2i. \quad ap \leq a(p \vee q).$$

We call such algebras *isodynamic*; every dynamic algebra is an isodynamic algebra as is easily verified. To see that the converse does not hold, let \mathcal{B} have elements $0, P, P', 1$ and let A be the function on \mathcal{B} mapping P' to P and fixing everything else. A is isotone but not finitely additive. Take \mathcal{R} to be $(\{A, A^*\} \cup ; *)$ where A^* maps P' to 1 and fixes everything else, with

\cup pointwise disjunction and $;$ composition. Take \diamond to be application. Then $(\mathcal{B} \mathcal{R} \diamond)$ is an isodynamic algebra, as may be verified by calculation.

A more interesting example takes \mathcal{B} to be the power set algebra on N and \mathcal{R} to contain for each $i \geq 0$ the function A^i on 2^N that removes the least i elements from its argument. These functions can be seen to be isotone but not finitely additive. Define $A^i \cup A^j = A^{\min(i,j)}$, $A^i ; A^j = A^{i+j}$, and $(A^i)^* = A^0$. Taking \diamond to be application, it can be verified that $(\mathcal{B} \mathcal{R} \diamond)$ is a separable isodynamic algebra whose regular component is isomorphic to the above-mentioned $(N \text{ min} + K0)$ algebra.

One would hope that Example 1 generalizes in the obvious way to isodynamic algebras. Suppose it did. Consider the full isodynamic algebra on $\{0, 1, 2\}$, with of course $a^*p = \min(a!p)$. Lemma 4.3 works for isotone functions so there is no question about \cup and $;$ and $*$ being defined. Now let us write p as $\sum_{i \in p} 2^i$, e.g. $\{1, 2\}$ is written as 6. Let A be some isotone function such that $A1 = A2 = 2, A3 = A5 = 7$. Then $A!1 = \{7\}$ and $2 \in A!2$, so $A^*1 = \min(A!1) = 7$ and $A^*2 = \min(A!2) = 2$. But $1 \vee A^*(1' \wedge A1) = 1 \vee A^*(6 \wedge 2) = 1 \vee A^*2 = 1 \vee 2 = 3$, contradicting axiom 5b. Thus “full isodynamic algebras” are not isodynamic and so Example 1 does not generalize. Incidentally this shows why isotonicity is inadequate for the \leftarrow direction of Lemma 3.1; if this direction worked with isotonicity the remaining arguments leading up to Example 1 would all go through for full isodynamic algebras. We do not at present have any nice characterization of which isotone functions on a complete Boolean algebra satisfy axiom 5b under the $\min(a!p)$ interpretation of a^*p .

We now exhibit a regular component of a separable isodynamic algebra which is not in HSP(SDA). Let P generate a four element Boolean algebra \mathcal{B} , and take A to be the function on B mapping P to P' and fixing the rest of B . Take I to be the identity function on B . Then $A(A \cup I)P = 1$ but $AA P = AIP = P'$ so $A(A \cup I)P \neq (AA \cup AI)P$. Hence $A(A \cup I) \neq AA \cup AI$, contradicting a law of HSP(SDA). The closest we can come to this law is $ab \cup ac \leq a(b \cup c)$. However $(a \cup b)c = ac \cup bc$ holds⁸. Note also that Theorem 3.4 holds, so $*$ is reflexive transitive closure in a separable isodynamic algebra.

With the above in mind we identify the class of regular components of separable isodynamic algebras as an interesting class of regular algebras.

7. LOOSE ENDS

Complexity. The set of valid formulas of propositional dynamic logic has been shown to be complete in deterministic exponential time, the lower bound

⁸This asymmetric distributivity is also encountered in connection with the subsequent extensive literature on bisimulation [Mil80]; by the time I became acquainted with this literature I had forgotten about this aspect of isodynamic algebras, and only noticed the connection while making these annotations.

appearing in [FL79], the upper bound in [Pra80c]. (A simpler proof of the upper bound appears in [Pra79a].) The intimate connection between that theory and the equational theory of dynamic algebras establishes the same complexity result for the latter. The theory of separable dynamic algebras has the same complexity since its theorems are those of dynamic algebras together with $a = b$ if and only if $aP = bP$ (P any propositional variable) is a theorem.

Origin. The origin of dynamic algebra is in dynamic logic ([Pra76], see also [Pra80a]), whose origin is in turn in modal logic. It is rather surprising that the calculus of binary relations was not incorporated into the Kripke semantics of modal logic earlier. As pointed out at the end of [FL79], the classical modal logics K, T, S3, S4, and S5 are all special cases of dynamic logic (S5 requires converse), so that decidability of satisfiability for each of these logics follows from the procedure given in [FL79] for propositional dynamic logic. Closely related logics, all addressed specifically at programming, are those of Hoare [Hoa69], Salwicki [Sal70], Dijkstra [Dij76], and Constable [Con77]. Example 4 makes a connection with the logic of Floyd [Flo67].

Terminology. We had originally called dynamic algebras Hoare algebras, after [Hoa69], which contains a Gentzen-like form of the dynamic logic theorems given in [Pra76] and extended by Segerberg to a propositionally complete system. We adopted Kozen’s term “dynamic algebra” after seeing [Koz79c]. Some time later we realized that Kozen’s definition of a dynamic algebra was strictly stronger than ours because of its assumptions about continuity; we have yet to resolve this terminological conflict⁹. The term “separable” is also Kozen’s, as is the analogy of dynamic algebras to modules with regular elements acting as “scalars.”

There is a close similarity between the axioms for dynamic algebras and those for modules. One might call a dynamic algebra a *Boolean semimodule with ** (cf. [SS78] which employs a concept of semimodule, though neither Boolean nor having *).

Open Problems. We repeat Kozen’s problem, is every *-continuous dynamic algebra isomorphic to a Kripke structure?¹⁰

We have treated neither tests, $p?$, nor converse, a^\smile . Converse formalizes the idea of running a program in reverse, and was studied in this context in [dBdR72]. See [Par78] for a complete axiomatization of converse (essentially $a(a^\smile p)' \wedge p = 0$ and $a^\smile(ap)' \wedge p = 0$). Is there an algebraic analogue of our

⁹We subsequently agreed [Koz81b] that a dynamic algebra was any model of the Segerberg axioms, and a *-continuous dynamic satisfied the additional condition $a^* = \bigvee_{i < \omega} a^i$ along with distributivity of \cup over such joins.

¹⁰Solved negatively by Kozen [Koz80a], further strengthened by Nemeti [Ném81].

Lemma 3.1 for converse?¹¹

What is the axiom rank and base rank [Coh65, p173] for dynamic algebras? The base rank is at most 3 since the Segerberg axiomatization only requires 3 variables in each axiom.

What reasonable axioms exist for regular complement (complement in the regular algebra) in dynamic algebra? We propose the term “complemented dynamic algebra” for a dynamic algebra expanded by adding regular complement.¹²

A related question that the reader may find better defined deals with the problem of recognizing identities for the complemented linguistic dynamic algebras $(\langle 2^L \vee ' \rangle) (\langle 2^L \cup ; * ' \rangle \diamond)$ where $L = S^* \cup S^\omega$ and S is an arbitrary alphabet. Is this problem even decidable?¹³

8. BACKGROUND

A grasp of the ideas in one of [Bir67, Coh65, Gra68] is more than sufficient preparation for this paper. The following first aid may prove convenient for the reader comfortable with the definitions of subalgebra, homomorphism, direct product, lattice, and Boolean algebra.

Minor Points. We use *poset* for partially ordered set, *sup(remum)* or *join*

for the least upper bound $\bigvee S$ of a subset S of a poset, and *inf(imum)* or *meet* for the greatest lower bound $\bigwedge S$. Note that $\bigvee \emptyset$ exists in a poset P just when P contains a least element, as it does when P is a Boolean algebra. A directed set is a non-empty set which contains upper bounds on each of its finite subsets. A *complete sublattice* of a poset P contains all its meets and joins as defined in P . The *power set algebra* on the set X is a complete Boolean algebra consisting of all subsets of the set X . A *field of sets* is any subalgebra of a power set algebra, necessarily Boolean, not necessarily complete or atomic; conversely every Boolean algebra is isomorphic to a field of sets [Sto36]. Being complete and being atomic are independent for infinite Boolean algebras and both true for finite ones.

Heterogeneous Algebras. As is shown in [BL70], all of the following carries through for heterogeneous algebras of the sort used here, in our case having up to two carriers B and R and up to eight operations, $\vee ' 0 \cup ; * \diamond ?$, of various types. Direct products and homomorphisms respect type in the heterogeneous

¹¹Solved affirmatively by the author [Pra81b].

¹²Solved negatively, long ago, by Monk [Mon64], since equipping the regular part with complement recreates relation algebras, making the Boolean part redundant. However the tractability resulting from segregating the regular and Boolean parts, in particular decidability and representability, is then lost [Bri81, Pra90].

¹³It is undecidable since it embeds the (undecidable) word problem for semigroups. Tarski remarked on this situation early on in his development of relation algebras [Tar41].

case; one would not find a Boolean element paired with a regular in a direct product, nor a homomorphism mapping a Boolean element to a regular.

Closure Operators. A *closure operator* on a poset P is an isotone reflexive idempotent function f on P ; that is, $x \leq y$ implies $fx \leq fy$, $x \leq fx$, and $ffx = fx$. A *closure system* on a poset is the set of fixed points of some closure operator. Every complete sublattice L of a poset P is a closure system with associated closure operator $J(x) = \bigwedge \{y \in L \mid x \leq y\}$. P itself need not form a complete lattice, though it must have a greatest element, which will appear in every complete sublattice as $\bigwedge \emptyset$. When P is a complete lattice, every closure system of P is a complete sublattice of P .

Generative Systems. A system of rules of the form, “if u is in the set then so are v, w, \dots ,” is called a *generative system*. The set of all subsets of a given *source set* each meeting all the rules is clearly closed under intersection and so forms a closure system. Hence associated with any generative system and source set is a closure operator on the power set of the source set.

Generator Sets. The set of subalgebras of an algebra \mathcal{A} is closed under intersection and so forms a closure system. The associated closure operator G yields for each subset X of A the subalgebra $G(X)$ *generated by* X . X is called a *generating set* of $G(X)$, and the elements of X are called *generators*. A useful property of G is that it commutes with homomorphisms; $h(G(X)) = G(h(X))$ for any subset X of A and homomorphism h from \mathcal{A} .

Free Algebras. Given a class C of similar algebras, an algebra \mathcal{A} of the same similarity type, not necessarily in C , is *C -free* when it contains a generating set A_0 such that any function from A_0 to an algebra in C extends to a homomorphism, necessarily uniquely by the previous paragraphs. When such an \mathcal{A} is in C we call \mathcal{A} a *free C -algebra*. (The standard notion of “free” is the latter; we need the former, *C -free*, for Theorem 5.3.)

Any non-trivial class closed under subalgebras and direct products has free algebras [Coh65, p138].

Varieties. A class of similar algebras closed under subalgebras, direct products, and homomorphisms is called a *variety*; equivalently [Bir35], a variety is any class defined by a (possibly infinite) set of equational identities. Thus Boolean and dynamic algebras form varieties, being defined purely by equational identities.

The class of all varieties of a similarity type is closed under arbitrary intersection (take the variety defined by the union of their theories) and so form a closure system, so that the variety generated by any class of similar algebras always exists. We write $\text{HSP}(C)$ for the variety generated by class C . If $a \neq b$ in any free algebra \mathcal{A} of $\text{HSP}(C)$ then there exists an algebra \mathcal{A}' of C and a

homomorphism $h : \mathcal{A} \rightarrow \mathcal{A}'$ for which $h(a) \neq h(b)$. (This is because any law of C algebras must be a law of $\text{HSP}(C)$ algebras.)

A set of congruences on an algebra \mathcal{A} is called *separating* when no two elements of \mathcal{A} are related in every congruence. \mathcal{A} is *residually P* , P being some property of algebras, when \mathcal{A}/\cong is P for each congruence \cong in some separating set of congruences on \mathcal{A} .

If \mathcal{A} is residually P then \mathcal{A} is isomorphic to a substructure of a direct product of P algebras, which themselves are homomorphic images of \mathcal{A} (the natural homomorphism onto the quotient). Substructures of direct products obtained in this way from separating sets of congruences are called *subdirect products*. Thus given a class C of residually P algebras it follows easily that $\text{HSP}(C) = \text{HSP}(P) \cap \text{H}(C)$ where $\text{HSP}(C)$ is the variety generated by C and $\text{H}(C) \subseteq \text{HSP}(C)$ is the class of homomorphic images of C .

Varieties always have free algebras. Lemma 5.8 shows that free separable dynamic algebras with at least one Boolean generator exist.

Letting SDA denote the class of separable dynamic algebras, we point out that both free dynamic algebras and free separable dynamic algebras are SDA-free, an essential aspect of Theorem 5.3.

Deductive Systems. It is a well-publicized fact that the pure predicate calculus has a deductive system that is complete in the sense that it can be used to prove from a set Γ of first-order formulas, the *non-logical axioms*, any first-order formula valid in the axiomatic class consisting of all models of Γ . Less well-publicized is the fact that the rules permitted for manipulating equations in high school algebra is complete in the same sense, with pure equations in place of first-order formulae and varieties in place of axiomatic classes. The rules invariably include implicitly those in the following system, whose only logical axiom is $X = X$ for some variable X .

Symmetry: $x = y \vdash y = x$.

Transitivity: $x = y, y = z \vdash x = z$.

Replacement: $x_1 = y_1, \dots, x_n = y_n \vdash f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$ for any n -ary f .

Substitution: $s(X) = t(X) \vdash s(y) = t(y)$ for any variable X and term y , i.e. the substitution of y for all occurrences of X in the theorem $s = t$.

These rules form a generative system with source set pairs of elements of a word algebra, and the associated closure operator is called deductive closure. The system is complete in the sense that the deductive closure of a set of axioms coincides with the set of equations holding identically in the variety defined by

the axioms. The proof is easier than for the first-order case, see e.g. [Hen77].

The significance of this fact for us is that we need not get involved in the details of deductive closure since this is now all taken care of for us. Our only obligation is to ensure that any given system of axioms really does define the same variety as that generated by the class of models we are interested in.

Retrospective citations. At this publication date of 1990 there are certain bibliographic items appearing subsequently to the 1979 appearance of this memo that are worth mentioning here.

Surveys on dynamic logic: Harel [Har84], Kozen and Tiuryn [KT89].

Papers on dynamic algebras (by no means a complete list): [Koz79d, Koz79c, Koz79a, Koz79b, Koz80a, Koz80b, Koz81b, Koz81a, Ném81, Koz91, Pra90].

Related work in the relation algebra (RA) community: Brink on Boolean modules [Bri81], Ng and Tarski on RAT's (RA's with transitive closure) [NT77, Ng84], both discussed in [Pra90].

Propositional Mu-calculus. Pratt [Pra81a] defined a propositional mu-calculus by turning Lemma 3.1 into a minimization construct, and showed that it was decidable. This mu-calculus was subsequently extended by Kozen [KP83].

Reflections. The present paper had two motivations. First I wanted to round out the diagram in [Pra79a] of inclusions among theories of various models of propositional dynamic logic. I had given algebraic proofs of all inclusions in that diagram except one, namely the inclusion of the theory of Kripke structures in that of dynamic algebras. This inclusion was available via Parikh's proof of completeness (with respect to Kripke structures) of Segerberg's axioms, but it was by no means obvious how to algebrize Parikh's proof, nor whether algebrization would strengthen the result (it didn't).

Second I had become quite frustrated with the existing proofs of completeness of Segerberg's axioms [Seg77] for PDL. My previous attempts to understand this completeness theorem resulted in [Pra80c], which drew principally on the techniques of Beth tableaux. I hoped that algebrization of the completeness theorem would shed light on why Segerberg's axioms should be complete.

I was very pleasantly surprised to find that many of the tedious calculations in [Pra80c] could be absorbed into general theorems of universal algebra and algebraic logic. This reduced the completeness proof to the calculations of the present paper. It also abstracted the essence of the result as a factorizability property: every free dynamic algebra can be factored as a subdirect product of finite dynamic algebras. This expression of the completeness result has the advantage of making no mention of Kripke structures, remaining in the abstract domain of dynamic algebras. Yet the basic argument is essentially Fischer and

Ladner’s original finite model argument, translated to the more general setting of dynamic algebras. Moreover it is milked not just for decidability but completeness of the Segerberg axioms, the main and intended payoff of this generalization.

I was less pleasantly surprised to find later on that hardly any computer scientists understood enough algebra to comprehend even the statements of the theorems, let alone their proofs. The essentially universal reaction seemed to be that I had now stopped studying dynamic logic and had turned my attention to algebra, a branch of pure mathematics having no bearing on the practical problems of reasoning about computer programs. Dexter Kozen was of course an exception to this rule, as were Istvan Németi, Hajnal Andréka, Fran Berman, Vera Trnkova, and Jan Reiterman. But many people whom I had expected to appreciate the elegance and simplicity of the algebraic approach were not only unmoved by it but saw no contribution whatsoever to the overall understanding of dynamic logic, let alone the fundamental one I was claiming!

Fortunately or otherwise I did not fully appreciate this lack of response until the late 1980’s. By that time I had ventured considerably deeper into the labyrinths of abstraction, going so far as to apply enriched category theory to modelling concurrency [CCMP91]. Here the lack of response was more immediately noticeable, and correspondingly more quickly diagnosed. The clear diagnosis was that the most sympathetic readers of the computer science literature can follow only so much abstract nonsense before they get out of their depth and must make an expensive investment in order to keep going.

I know of only two fixes for this problem. The first is for the perpetrators of such impenetrabilia to restrict themselves to the absolute minimum of abstract nonsense sufficient to achieve the end at hand. The second is to give the field at large the necessary encouragement it needs to get caught up.

My assessment is that the first of these fixes has two problems: it leads to unnecessarily clumsy exposition, and it is at best a stop-gap solution to the basic problem. In the long run only the second fix will have any lasting value. The more comfortable the field as a whole becomes with the “power tools” of abstract reasoning, the faster it will move forward.

Acknowledgments. D. Kozen was a most helpful source of ideas and pointers into the literature. J. Halpern provided many helpful comments on an early draft of the paper. Comments of I. Németi and D. Pigozzi substantially improved this publication version.

9. BIBLIOGRAPHY

References

- [Bir35] G. Birkhoff. On the structure of abstract algebras. *Proc. Cambridge Phil. Soc.*, 31:433–454, 1935.
- [Bir67] G. Birkhoff. *Lattice Theory*, volume 25. A.M.S. Colloq. Publications, 1967.
- [BL70] G. Birkhoff and J.D. Lipson. Heterogeneous algebras. *J. of Combinatorial Theory*, 8:115–133, 1970.
- [Bri81] C. Brink. Boolean modules. *J. of Algebra*, 71:291–313, 1981.
- [CCMP91] R.T Casley, R.F. Crew, J. Meseguer, and V.R. Pratt. Temporal structures. *Math. Structures in Comp. Sci.*, 1(2):179–213, July 1991.
- [Coh65] P.M. Cohn. *Universal Algebra*. Harper and Row, New York, 1965.
- [Con71] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, London, 1971.
- [Con77] R.L. Constable. On the theory of programming logics. In *Proc. 9th Annual ACM Symp. on Theory of Computing*, pages 269–285, Boulder, Col., May 1977.
- [dBdR72] J.W. de Bakker and W.P. de Roever. A calculus for recursive program schemes. In M. Nivat, editor, *Automata, Languages and Programming*, pages 167–196. North Holland, 1972.
- [Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, N.J., 1976.
- [FL79] M.J Fischer and R.E. Ladner. Propositional dynamic logic of regular programs. *JCSS*, 18(2), 1979.
- [Flo67] R.W. Floyd. Assigning meanings to programs. In J.T Schwartz, editor, *Mathematical Aspects of Computer Science*, pages 19–32, 1967.
- [Gra68] G. Graetzer. *Universal Algebra*. Van Nostrand, Princeton, NJ, 1968.
- [Hal62] P.R. Halmos. *Algebraic Logic*. Chelsea, New York, 1962.
- [Har84] D. Harel. Dynamic logic. In *Handbook of Philosophical Logic. II: Extensions of Classical Logic*, pages 497–604. D. Reidel, Boston, 1984.
- [Hen77] L. Henkin. The logic of equality. *Amer. Math. Monthly*, 84(8):597–612, October 1977.

- [Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:576–580, 1969.
- [Hor51] A. Horn. On sentences which are true of direct unions of algebras. *J. Symbolic Logic*, 16:14–21, 1951.
- [Kle56] S.C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–42. Princeton University Press, Princeton, NJ, 1956.
- [Koz79a] D. Kozen. On the duality of dynamic algebras and Kripke models. In E. Engeler, editor, *Proc. Workshop on Logic of Programs 1979, LNCS 125*, pages 1–11. Springer-Verlag, 1979.
- [Koz79b] D. Kozen. On the representation of dynamic algebras. Technical Report RC7898, IBM, October 1979.
- [Koz79c] D. Kozen. A representation theorem for models of $*$ -free PDL. Technical Report RC7864, IBM, September 1979.
- [Koz79d] D. Kozen. A representation theorem for models of $*$ -free PDL. Manuscript, May 1979.
- [Koz80a] D. Kozen. On the representation of dynamic algebras ii. Technical Report RC8290, IBM, May 1980.
- [Koz80b] D. Kozen. A representation theorem for models of $*$ -free PDL. In *Proc. 7th Colloq. on Automata, Languages, and Programming*, pages 351–362, July 1980.
- [Koz81a] D. Kozen. Dynamic algebra: Section in: Propositional dynamic logics of programs: a survey, by R. Parikh. In E. Engeler, editor, *Proc. Workshop on Logic of Programs 1979, LNCS 125*, pages 102–144. Springer, 1981.
- [Koz81b] D. Kozen. On induction vs. $*$ -continuity. In D. Kozen, editor, *Proc. Workshop on Logics of Programs 1981, LNCS 131*, pages 167–176. Springer-Verlag, 1981.
- [Koz91] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proc. 6th Annual IEEE Symp. on Logic in Computer Science*, pages 214–225, Amsterdam, July 1991.
- [KP83] D. Kozen and R. Parikh. A decision procedure for the propositional μ -calculus. In E. Clarke and Kozen D., editors, *Proc. Workshop on Logics of Programs 1983, LNCS 164*, pages 313–325. Springer-Verlag, 1983.

- [KT89] D. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*. North Holland, Amsterdam, 1989.
- [Mil80] R. Milner. *A Calculus of Communicating Systems, LNCS 92*. Springer-Verlag, 1980.
- [Mon64] J.D. Monk. On representable relation algebras. *Michigan Math. J.*, 11:207–210, 1964.
- [Ném81] I. Németi. Dynamic algebras of programs. In *Proc. Fundamentals of Computation Theory, LNCS 117*, pages 281–290. Springer-Verlag, 1981.
- [Ném82] I. Németi. Every free algebra in the variety generated by the representable dynamic algebras is separable and representable. *Theoretical Computer Science*, 17:343–347, 1982.
- [Ng84] K.C. Ng. *Relation Algebras with Transitive Closure*. PhD thesis, University of California, Berkeley, 1984. 157+iv pp.
- [NT77] K.C. Ng and A. Tarski. Relation algebras with transitive closure, Abstract 742-02-09. *Notices Amer. Math. Soc.*, 24:A29–A30, 1977.
- [Par78] R. Parikh. A completeness result for a propositional dynamic logic. In *LNCS 64*, pages 403–415. Springer-Verlag, 1978.
- [Pnu77] A. Pnueli. The temporal logic of programs. In *18th IEEE Symposium on Foundations of Computer Science*, pages 46–57, October 1977.
- [Pra76] V.R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proc. 17th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 109–121, October 1976.
- [Pra79a] V.R. Pratt. Models of program logics. In *20th Symposium on foundations of Computer Science*, San Juan, October 1979.
- [Pra79b] V.R. Pratt. Process logic. In *Proc. 6th Ann. ACM Symposium on Principles of Programming Languages*, pages 93–100, San Antonio, January 1979.
- [Pra80a] V.R. Pratt. Application of modal logic to programming. *Studia Logica*, 34(2/3):257–274, 1980.
- [Pra80b] V.R. Pratt. Dynamic algebras and the nature of induction. In *12th ACM Symposium on Theory of Computation*, Los Angeles, April 1980.

- [Pra80c] V.R. Pratt. A near optimal method for reasoning about action. *Journal of Computer and System Sciences*, 2:231–254, April 1980. Also MIT/LCS/TM-113, M.I.T., Sept. 1978.
- [Pra81a] V.R. Pratt. A decidable mu-calculus. In *Proc. 22nd IEEE Conference on Foundations of Computer Science*, pages 421–427, October 1981.
- [Pra81b] V.R. Pratt. Using graphs to understand PDL. In D. Kozen, editor, *Proc. Workshop on Logics of Programs 1981, LNCS 131*, pages 387–396. Springer-Verlag, 1981.
- [Pra90] V.R. Pratt. Dynamic algebras as a well-behaved fragment of relation algebras. In *Algebraic Logic and Universal Algebra in Computer Science, LNCS 425*, Ames, Iowa, June 1988, 1990. Springer-Verlag.
- [Red64] V.N. Redko. On defining relations for the algebra of regular events (Russian). *Ukrain. Mat. Z.*, 16:120–126, 1964.
- [Sal66] A. Salomaa. Two complete axiom systems for the algebra of regular events. *Journal of the ACM*, 13:158–169, 1966.
- [Sal70] A. Salwicki. Formalized algorithmic languages. *Bull. Acad. Pol. Sci., Ser. Sci. Math. Astr. Phys.*, 18(5), 1970.
- [Seg77] K. Segerberg. A completeness theorem in the modal logic of programs. *Notices of the AMS*, 24(6):A-552, October 1977.
- [SS78] A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag, New York, 1978.
- [Sto36] M. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.
- [Tar41] A. Tarski. On the calculus of relations. *J. Symbolic Logic*, 6:73–89, 1941.