

SEMIGROUPS WITH IF-THEN-ELSE AND HALTING PROGRAMS

MARCEL JACKSON AND TIM STOKES

ABSTRACT. The “if-then-else” construction is one of the most elementary programming commands, and its abstract laws have been widely studied, starting with McCarthy. Possibly the most obvious extension of this is to include the operation of composition of programs, which gives a semigroup of functions (total, partial or possibly general binary relations) that can be recombined using if-then-else. We show that this particular extension admits no finite complete axiomatization and instead focus on the case where composition of functions with predicates is also allowed (and we argue there is good reason to take this approach). In the case of total functions—modelling halting programs—we give a complete axiomatization for the theory in terms of a finite system of equations. We obtain a similar result when an operation of equality test and/or fixed point test is included.

1. MOTIVATION AND SUMMARY OF RESULTS

1.1. Motivation and background. Let X be a set. The semigroup $T(X)$, consisting of all transformations $X \rightarrow X$ under composition, is a generic semigroup: all semigroups embed in such examples. However, $T(X)$ is a somewhat impoverished object in terms of available operations. By comparison, the larger set $R(X)$ of binary relations on X possesses many operations. Aside from relational composition, all of the usual Boolean set-theoretic operations (including bottom and top) make sense in $R(X)$, as does relational converse. Adding in 1 to the mix (representing the diagonal relation), and abstracting out essential properties, gives rise to the theory of *relation algebras*, an enrichment of the theory of semigroups that is very rich indeed. One can further add in the operation of Kleene closure (reflexive transitive closure) to obtain yet another useful and important operation. Of these operations, $T(X)$ is closed under very few: only composition and the identity element can be retained!

One has more success if one broadens the outlook to consider the set $P(X)$ of (partial) functions $X \rightarrow X$. $P(X)$ is a semigroup under composition, but is also closed under intersection and indeed set difference, a number of variants of union, and various other operations relating to domain, range, domains of non-disagreement, and maximal iteration, as discussed in many articles: see the survey article by Schein [19], or that by the authors [10].

But there is at least one “operation” on $T(X)$ that still makes sense, that of piecewise combination. Thus, given $f, g \in T(X)$ and a “test” or predicate α defined on X , one can form the transformation h which agrees with f whenever α is satisfied and with g when it is not. Of course such a piecewise definition is a very familiar way of defining new transformations from old: the absolute value function has just such a description for example. (In that case, $X = \mathbb{R}$, $\alpha = “(x \geq 0)”$, $f(x) = x$ and

Key words and phrases. transformation; semigroup; if-then-else; equality test.

$g(x) = -x$.) By iterating the process of taking such pairwise combinations, one can obtain arbitrary piecewise combinations of transformations.

The operation just described corresponds to the notion of **if-then-else** for transformations on a set X , familiar from the theory of computer programs. If α is a predicate or “test” on X and $f, g \in T(X)$, then define “if α then f else g ” as follows:

$$\alpha[f, g](x) := \begin{cases} f(x) & \text{if } x \text{ satisfies } \alpha, \\ g(x) & \text{otherwise.} \end{cases}$$

Of course the notion of a piecewise-defined function has wide applicability: one speaks of piecewise-linear, piecewise-continuous, piecewise-differentiable functions, and so on.

Quite a lot of previous work has been done on the **if-then-else** operation. McCarthy gave an axiomatization of it in [15], having in mind a model of “programs” as functions $X \rightarrow Y$, and “tests” as comprising a Boolean algebra of predicates acting on the set of programs; in this formulation, composition is not present. In [1], Bergman defined the similar notion of a set with B -action, or a B -set, and gave a sheaf-theoretic representation. (This concept was extended to cover cases where the set had algebraic structure, with the operations modelling pointwise operations on functions: *algebras with B -action*.) Other investigations into the equational properties of various forms of **if-then-else** can be found in the articles of Bloom and Tindell [2], Meklar and Nelson [16], and Guessarian and Meseguer [5].

Manes considered an extension of the McCarthy approach to **if-then-else** that also included a join operation. Semantically, the programs were modelled as binary relations between two distinct sets such that for each binary relation, each element has a finite image: see [14]. The join operation then modelled relational union. The Boolean algebra of computable tests was also generalised, to allow modelling of tests which may not halt.

The case of **if-then-else** together with composition has received considerable attention also, mostly from theoretical computer scientists, in settings where programs are modelled as binary relations on a set. In the setting of program verification theory and program logics, the classes of dynamic algebras (see Pratt [18] for example) and Kleene algebras with tests (see Kozen [12] for example) furnish two algebraic environments in which **if-then-else** plus composition (and indeed “**while**”) can be captured, at least at some equational level, and relative to a relational interpretation of the program variables.

In the Kleene algebra with tests approach, the “tests” are embedded into the program sort, where they are interpreted as restrictions of the identity mapping. (This is implicitly true in the dynamic algebra with tests approach too: indeed this is inevitable if the identity and empty relations are part of the program signature, as is argued in [9].) However, such an approach is in principle impossible for algebras of transformations, since such restrictions are obviously not transformations! An intrinsically external notion of test is needed, more like the approach taken by McCarthy, Bergman and Manes, but with the capacity to capture composition as well.

In many settings, it is quite natural to consider composition of transformations with predicates. For example, the composition of the polynomial function x^2 with the piecewise function $|x| = (x \geq 0)[x, -x]$ is just $(x^2 \geq 0)[x, -x] = x$; in computing this, the predicate $(x^2 \geq 0)$ was obtained as the composition of the function x^2 with

the predicate $(x \geq 0)$. In any case, since such composites can be computed, it is reasonable to attempt to model this operation, rather than just composition of transformations.

The concept of the composition of a transformation with a predicate is a particular case of the “predicate transformers” considered by Dijkstra in [4]. These arise in the setting of formal methods of program verification via certain natural modal operators corresponding to “weakest preconditions”. Thus given a program f and a test α , the predicate $[f]\alpha$ is true exactly when an input to f gives rise to no output satisfying the negation of α (the “weakest liberal precondition” to f given the “postcondition” α); its natural dual is the predicate $\langle f \rangle \alpha$, which is true exactly when an input gives rise to some output satisfying α . If f is a transformation, the two predicates $[f]\alpha$ and $\langle f \rangle \alpha$ are equal, and indeed each is nothing but the composition of the transformation f with the predicate α as just discussed. The truth set of this predicate is the preimage of the truth set of α under the transformation f .

The notion of a Kleene algebra with domain (KAD), considered in [13], permits expression of such predicate transformers in terms of a *domain operator*, in which the test corresponding to the domain of a binary relation f , $\delta(f)$, is modelled as a restriction of the identity. Indeed if α is any such restriction of the identity, and f is a transformation, we have $\langle f \rangle \alpha = \delta(f\alpha)$ and $[f](\alpha) = \delta(f\alpha')$, where α' denotes the restriction of the identity to the complement of the domain of α . Again, both of these are just the composite of f with α if f is a transformation.

Recently, a partial function version of KAD has been considered; see [9]. Partial functions are not closed under union or Kleene closure, but are closed under certain natural analogs capable of capturing the **if-then-else** and **while** program operators for deterministic programs. Representation theorems rather stronger than known results for the non-deterministic case can be obtained. The current work can be viewed as a variation of this work in which all functions are assumed to be total (the “programs” are halting on all inputs), and the tests are necessarily external to the function algebra.

In practice, tests have specific forms, such as “ $(x \geq 0)$ ”. In the general setting in which the domain set X has no assumed structure, the only test that can sensibly be defined relates to equality of elements, and by extension, pointwise equality of transformations. Thus given two transformations f, g on X , one may form the predicate $f * g$, which is true at $a \in X$ exactly when $f(a) = g(a)$. This predicate $f * g$ is computable if both f, g are programs halting on all inputs. Such equality tests were considered in conjunction with **if-then-else** (but not composition) by Pigozzi [17], where the quasi-equational theory of such algebras was considered for the special case of the two-element Boolean algebra.

1.2. Summary of results. Our main aim here is to study abstract algebras of transformations and tests, equipped with exactly the operations referred to in the above discussion, namely the Boolean operations on the predicates, composition of transformations with each other and with tests, and a family of binary **if-then-else** operations on the transformations (one for each test). Hence these algebras are simultaneously enrichments of semigroups and of B-sets, but also possess a “predicate transformer” operation allowing composition of transformations and predicates. We characterize (in Section 3) the two-sorted algebras embeddable in the concrete transformation examples, this class proving to be a finitely based

(two-sorted) variety, and we solve the equational problem in this variety by showing how to compute in free algebras (Section 5). We also enrich the signature by adding in an equality test operation as described above; in this case the class of representable algebras continues to be a finitely-based variety and we give defining laws (Section 7). Along the way, we recover Pigozzi’s results for the case of **if-then-else** and equality tests without composition (Section 2).

In Section 6 we show that composition of transformations with predicates is necessary, in the sense that without it, no finite axiomatization is possible. There is interest here from a computing point of view because we are modelling a large fragment of the algebra of halting programs: if we begin with functions and tests that are computable for all inputs and compose them in any way possible, or if we take some piecewise combination of the functions using the tests, the result is still computable (here meaning “will halt on all inputs”). We also show no finite axiomatization is possible in both the partial function and general binary relation cases; moreover, in these non-transformation settings, the argument carries over to include the operation of **while**. Thus we show that the smallest algebraic language capable of expressing the language of “while” programs (namely Boolean operations on tests, composition, **if-then-else** and **while**) fails to have a finite axiomatization for the universal sentences that are true for either functional (deterministic) models, or relational (nondeterministic) models.

We adopt a uniform notation for all composites, using a dot if necessary or juxtaposition where possible. Thus any two transformations f, g can be composed to give a third, fg (“first f then g ”, throughout what follows), and a transformation f and a predicate α can be composed to give the predicate $f\alpha$ (“first f then α ”). So, implicitly, the transformations in $T(X)$ act on the right of X . Nevertheless, for the sake of notational familiarity we use the notation $f(x)$ to denote the image of an element $x \in X$ under a map $f \in T(X)$, so that the composite fg acts as $(fg)(x) = g(f(x))$. We frequently identify the power set of a set X and the set of characteristic functions 2^X on X .

2. IF-THEN-ELSE AND EQUALITY TEST WITHOUT COMPOSITION

We begin with the “flat” or “non-dynamic” case, in which one models **if-then-else** and equality tests defined on a set. We obtain subdirect product representations in terms of “basic” algebras (in which the two-element Boolean algebra is used). Importantly, the method of proof extends readily to the “dynamic” situation in which composition of transformations is modelled.

2.1. If-then-else only. Let B be a Boolean algebra, with the operations of meet, join and complement denoted by \wedge, \vee and $'$ respectively, and nullaries denoting the top and bottom denoted by \top and \bot respectively. For our purposes, a B -set is defined to be a pair (X, B) where X is a set and B is a Boolean algebra, such there is a B -action $B \times X \times X \rightarrow X$, here denoted $\alpha[a, b]$, $\alpha \in B$ and $a, b \in X$, and satisfying, for all $\alpha, \beta \in B$ and $a, b, c \in X$,

- (1) $\alpha[a, a] = a$,
- (2) $\alpha[\alpha[a, b], c] = \alpha[a, c]$,
- (3) $\alpha[a, \alpha[b, c]] = \alpha[a, c]$,
- (4) $\top[a, b] = b$,
- (5) $\alpha'[a, b] = \alpha[b, a]$ and

$$(6) \quad (\alpha \wedge \beta)[a, b] = \alpha[\beta[a, b], b].$$

This is essentially the definition given in [1], except that there, the choice of Boolean algebra was built into the title “ B -set” (as in “associative K -algebra” for example). Here, we view B -sets as two-sorted, with the Boolean algebra B allowed to vary.

Every Boolean algebra B is such that (B, B) is a B -set, if we define

$$\alpha[\beta, \gamma] = (\alpha \wedge \beta) \vee (\alpha' \wedge \gamma)$$

for all $\alpha, \beta, \gamma \in B$, and in what follows we use this notation freely for any Boolean algebra. For us, the important B -sets are those of the form $(T(X), 2^X)$, with $\alpha[f, g]$ as defined in the introduction.

A special case of B -sets arises by letting $B = \mathbf{2} := \{\mathbf{F}, \mathbf{T}\}$, the two-element Boolean algebra, and setting $\mathbf{T}[s, t] = s$ and $\mathbf{F}[s, t] = t$ for all $s, t \in S$; then (S, B) is a B -set, and we call any such B -set *basic*. These are essentially the *if-then-else* algebras considered by Pigozzi in [17] (although we are assuming a single non-Boolean sort here).

We next show that each B -set is embeddable in a direct product of basic B -sets. Hence the laws for B -sets capture the quasi-equational theory of basic B -sets, a result also shown in [17] in greater generality. The method of the representation is critical in the work to follow, so we include all proofs for completeness.

Throughout, let (S, B) be a fixed B -set. As mentioned, Greek letters denote elements of B .

Let F be a filter of B . Define

$$E_F = \{(s, t) \in S \times S \mid \exists \alpha \in F : \alpha[s, t] = t\}.$$

Proposition 2.1. E_F is an equivalence relation on S .

Proof. E_F is reflexive since $\mathbf{T}[s, s] = s$ for all $s \in S$ and $\mathbf{T} \in F$. It is symmetric because if $\alpha[s, t] = t$ then $\alpha[t, s] = \alpha[\alpha[s, t], s] = \alpha[s, s] = s$. It is transitive because if $\alpha[s, t] = t$ and $\beta[t, u] = u$ where $\alpha, \beta \in F$, then $(\alpha \wedge \beta)[s, t] = \beta[\alpha[s, t], t] = \beta[t, t] = t$, and $(\alpha \wedge \beta)[t, u] = \alpha[\beta[t, u], u] = \alpha[u, u] = u$, so

$$(\alpha \wedge \beta)[s, u] = (\alpha \wedge \beta)[(\alpha \wedge \beta)[s, t], u] = (\alpha \wedge \beta)[t, u] = u,$$

with $\alpha \wedge \beta \in F$. □

Lemma 2.2. For $a, b \in S$ with $a \neq b$, there is an ultrafilter G of B for which $(a, b) \notin E_G$.

Proof. Pick a, b as in the theorem statement. Let

$$F = \{\gamma' \in B \mid \gamma[a, b] = b\}.$$

Suppose $\alpha', \beta' \in F$, so that $\alpha[a, b] = b$, and $\beta[a, b] = b$. Then

$$\begin{aligned} (\alpha \vee \beta)[a, b] &= (\alpha' \wedge \beta')[b, a] \\ &= \alpha'[\beta'[b, a], a] \\ &= \alpha[a, \beta[a, b]] \\ &= \alpha[a, b] \\ &= b, \end{aligned}$$

so $\alpha' \wedge \beta' = (\alpha \vee \beta)' \in F$.

Further, if $\gamma' \geq \alpha' \in F$, then $\gamma \leq \alpha$, and

$$\gamma[a, b] = (\gamma \wedge \alpha)[a, b] = \gamma[\alpha[a, b], b] = \gamma[b, b] = b,$$

so $\gamma' \in F$. Hence F is a (clearly non-empty) filter of B . Extend it to an ultrafilter G of B . Suppose $(a, b) \in E_G$; then $\alpha[a, b] = b$ for some $\alpha \in G$. Then because $\alpha' \in F \subseteq G$, we have $F = \alpha \wedge \alpha' \in G$, a contradiction. Hence $(a, b) \notin E_G$. \square

Let $S = \prod_i S_i$ be a direct product of the sets $S_i, i \in \mathcal{I}$. Letting B be the power set of \mathcal{I} and for $a, b \in S$, defining $\alpha[a, b]$ to have i -th entry equalling that of a if $i \in \alpha$, and equalling that of b otherwise, it is easily seen that (S, B) is a B-set. Indeed it is fairly obvious that it is isomorphic to the direct product of the basic B-sets defined on the S_i . We show that every B-set is embeddable in such a B-set.

Theorem 2.3. *Every B-set is a subdirect product of basic B-sets.*

Proof. Let (A, B) be a B-set. For any ultrafilter F of B , E_F is easily seen to be a congruence on A (respecting each $\alpha[\ , \]$ operation), and of course induces a congruence on B as well. Indeed the pair of homomorphisms $f : B \rightarrow B/F \cong \mathbf{2}$ and $h : A \rightarrow A/E_F$ is easily seen to constitute a two-sorted homomorphism from the B-set (A, B) to the basic B-sets $(A/E_F, \mathbf{2})$. Suppose $(a, b) \in \cap_F E_F$, the intersection taken over all ultrafilters F of B . Suppose $a \neq b$; then letting G be as in the statement of Lemma 2.2, we have that $(a, b) \notin E_G$, a contradiction. Hence $\cap_F E_F$ is trivial. Likewise the intersection of all the ultrafilters of B is $\{\mathbf{T}\}$. So A is a two-sorted subdirect product of the basic B-sets $(A/E_F, \mathbf{2})$. \square

A sheaf-theoretic description of B-sets is given by Bergman in [1], and the above proof extends easily to the algebras with B -action considered there (in which A has additional algebraic structure).

2.2. Equality tests. In computer programs, perhaps the most common type of test encountered is an equality query. For two transformations $f, g \in T(X)$, define:

$$f * g = \{x \in X \mid f(x) = g(x)\}.$$

(Strictly, we define it to be the predicate on X having this truth set!)

As noted earlier, such a test is computable if both f and g are halting programs. We now seek to incorporate this operation into our algebraic formalism. Two similar operations have been considered in the setting of partial functions by the current authors in [7] and [22], but for total functions the two operations agree with $*$ as just defined. We adopt our notation and nomenclature from [7] (although here our functions act on the right rather than on the left).

A B-set (S, B) is *agreeable* if it is equipped with an operation $*$: $S \times S \rightarrow B$ satisfying the following for all $s, t, u, v \in S$ and $\alpha \in B$:

- (1) $s * s = \mathbf{T}$;
- (2) $(s * t)[s, t] = t$;
- (3) $\alpha[s, t] * \alpha[u, v] = \alpha[s * u, t * v]$.

In fact $*$ can be described in terms of the usual ordering on B . Throughout let (S, B) be a fixed agreeable B-set.

Lemma 2.4. *For $a, b \in S$, $a * b = \max\{\alpha \in B \mid \alpha[a, b] = b\}$, and if $\alpha \leq a * b$, then $\alpha[a, b] = b$.*

Proof. Suppose $\alpha[a, b] = b$ for some $\alpha \in B$. But then

$$\begin{aligned} \alpha \wedge (a * b) &= \alpha \wedge (\alpha[a, a] * \alpha[a, b]) \\ &= \alpha \wedge ((\alpha \wedge (a * a)) \vee (\alpha' \wedge (a * b))) \\ &= \alpha \wedge (\alpha \wedge \alpha' \wedge (a * b)) \\ &= \alpha, \end{aligned}$$

so $(a * b) \geq \alpha$. But $(a * b)[a, b] = b$, so $a * b$ is as described.

If $\alpha \leq a * b$, then

$$\alpha[a, b] = (\alpha \wedge (a * b))[a, b] = \alpha[(a * b)[a, b], b] = \alpha[b, b] = b.$$

□

The definition of E_F can be given a simpler formulation in the presence of $*$.

Proposition 2.5. *For F a filter of B , $(a, b) \in E_F$ if and only if $a * b \in F$.*

Proof. Suppose $(a, b) \in E_F$. Then $\alpha[a, b] = b$ for some $\alpha \in F$ so $a * b \geq \alpha$ by the last lemma, and so $a * b \in F$ also.

Conversely, if $a * b \in F$, then $(a * b)[a, b] = b$ shows that $(a, b) \in E_F$. □

It is easily seen that if an agreeable B-set is basic, then necessarily $a * b = \top$ if $a = b$, with $a * b = \text{F}$ otherwise. These are the two-sorted cases of the **if-then-else** algebras with equality tests considered in [17].

To extend Theorem 2.3 to the agreeable case, we need two useful laws.

Lemma 2.6. *For $a, b, c \in S$, $(a * b) \wedge (b * c) \leq (a * c)$ and $a * b = b * a$.*

Proof. The fact that $a * b = b * a$ can be obtained from Proposition 2.5 and the symmetry of E_F (for any filter F). Now for the first law,

$$\begin{aligned} ((a * b) \wedge (b * c))[a, c] &= ((a * b) \wedge (b * c))[(a * b) \wedge (b * c)[a, b], c] \quad \text{by B-set law (2)} \\ &= ((a * b) \wedge (b * c))[(b * c)[(a * b)[a, b], c] \quad \text{by B-set law (6)} \\ &= ((a * b) \wedge (b * c))[(b * c)[b, b], c] \quad \text{by agreeable B-set law (2)} \\ &= ((a * b) \wedge (b * c))[b, c] \quad \text{by B-set law (1)} \\ &= (a * b)[(b * c)[b, c], c] \quad \text{by B-set law (6)} \\ &= (a * b)[c, c] \quad \text{by agreeable B-set law (2)} \\ &= c \quad \text{by B-set law (1),} \end{aligned}$$

and so by Lemma 2.4, $(a * b) \wedge (b * c) \leq (a * c)$. □

Theorem 2.7. *Every agreeable B-set is a subdirect product of basic agreeable B-sets.*

Proof. In the language of the proof of Theorem 2.3, we wish to show that $h(a) * h(b) = f(a * b)$ (with the choice of ultrafilter F fixed). It is sufficient to show that if $(a_1, a_2) \in E_F$ and $(b_1, b_2) \in E_F$, and if $a_1 * b_1 \in F$, so is $a_2 * b_2$. So assume $a_1 * a_2 \in F$, $b_1 * b_2 \in F$, and $a_1 * b_1 \in F$. Then $F \ni (a_1 * a_2) \wedge (a_1 * b_1) = (a_2 * a_1) \wedge (a_1 * b_1) \leq a_2 * b_1$ by Lemma 2.6, hence also $F \ni (a_2 * b_1) \wedge (b_1 * b_2) \leq (a_2 * b_2)$, so $a_2 * b_2 \in F$ as required. □

3. B-SEMIGROUPS

We now bring composition of functions into the picture. As discussed, we consider not only composition of pairs of transformations, but also of transformations with predicates. (The case of composition of transformations only is treated in Section 6.)

A *B-semigroup* (S, B) is a B-set for which S is a semigroup and there is an operator $\cdot : S \times B \rightarrow B$, often denoted by juxtaposition, and satisfying the laws

- (1) $a\top = \top$,
- (2) $(a\alpha) \wedge (a\beta) = a(\alpha \wedge \beta)$,
- (3) $a(\alpha') = (a\alpha)'$,
- (4) $a(b\alpha) = (ab)\alpha$,
- (5) $\alpha[a, b] \cdot c = \alpha[ac, bc]$,
- (6) $a \cdot \alpha[b, c] = (a\alpha)[ab, ac]$,
- (7) $\beta[a, b] \cdot \alpha = \beta[a\alpha, b\alpha]$.

The convention that Greek letters denote elements of B is used above, and will be throughout the remainder of the article.

Recall that $\beta[a\alpha, b\alpha]$ is calculated in Boolean terms: it is $(\beta \wedge a\alpha) \vee (\beta' \wedge b\alpha)$, where semigroup multiplication is given precedence over Boolean operations. These axioms bear a similarity to those of dynamic algebra [18], as well as to those of Kleene algebra with domain [13], if $a\alpha$ is written as $[a]\alpha$ or indeed $\langle a \rangle \alpha$ (which are equal in the current setting). Laws (1)–(4) together assert that S acts as a semigroup of endomorphisms on B . The remainder express the interactions between **if-then-else** and the two composition operations.

For X a set, $(T(X), 2^X)$ is a B-semigroup if one defines all operations as in the introduction: it is a B-set as usual, $T(X)$ is a semigroup under composition, and $\cdot : T(X) \times 2^X \rightarrow 2^X$ is defined to be composition of transformations with predicates. Familiar examples are provided by the piecewise constant functions, piecewise polynomial functions, piecewise continuous functions, and so on, all mapping the reals to itself, and each equipped with a suitable Boolean algebra of “pieces”.

We now extend the representation of B-sets to B-semigroups. The same family of equivalence relations E_F (F an ultrafilter) is used. Throughout, let (S, B) be a fixed B-semigroup.

Proposition 3.1. *Let F be a filter of B . Then E_F is a right regular equivalence relation on S , meaning that if $(s, t) \in E_F$ then $(su, tu) \in E_F$ for all $s, t, u \in S$.*

Proof. If $(s, t) \in E_F$ then $\alpha[s, t] = t$ for some $\alpha \in F$, and so for any $u \in S$, $tu = \alpha[s, t] \cdot u = \alpha[su, tu]$, and so $(su, tu) \in E_F$. \square

For any filter F of B , let $S_F = S/E_F \cup \{1\}$, where $1 \notin S$. The following is now easily established.

Corollary 3.2. *The function $\phi_F : S \rightarrow T(S_F)$ given by $\phi_F(a) = \psi_a$, where $\psi_a(\bar{x}) = \bar{x}\bar{a}$ for all $\bar{x} \in S/E_F$, with $\psi_a(1) = \bar{a}$, is a semigroup homomorphism.*

Indeed ϕ_F is just the right action of S on the blocks of a right congruence of S . (We mention here that our method can be viewed as an example of the “determinative pair” method, as described by Schein in [20] for example.)

Next we consider the Boolean sort. Our goal is to show that each Boolean test element determines a predicate on S_F such that the Boolean operations correspond

to the usual predicate connectives. This is equivalent to showing that the assignment of a truth set to each test is a homomorphism from B to the power set of the domain space (viewed as a Boolean algebra in the usual way).

Again let F be a fixed filter of B . Define $f_F : B \rightarrow 2^{S_F}$ (viewed here as the power set of S_F equipped with its usual Boolean operations) by setting, for each $\alpha \in B$,

$$f_F(\alpha) := \{\bar{x} \in S/E_F \mid x\alpha \in F\} \cup (\alpha : F),$$

where $(\alpha : F) = \{1\}$ if $\alpha \in F$ and is otherwise empty.

Theorem 3.3. *For each ultrafilter F , f_F is a homomorphism. Moreover for each $\alpha \in B$ and $s \in S$, $s\alpha$ has truth set the inverse image of the truth set of α under $\phi_F(s)$.*

Proof. We must first show that $f = f_F$ is well-defined. If $\bar{x} = \bar{y}$, then $(x, y) \in E_F$, so $\beta[x, y] = y$ for some $\beta \in F$. If also $x\alpha \in F$, then

$$y\alpha = \beta[x, y] \cdot \alpha = \beta[x\alpha, y\alpha] = (\beta \wedge (x\alpha)) \vee (\beta' \wedge (y\alpha)) \geq \beta \wedge (x\alpha) \in F,$$

and so $y\alpha \in F$ also, as required.

For $\alpha, \beta \in B$,

$$\begin{aligned} f(\alpha \wedge \beta) &= \{\bar{x} \in S/E_F \mid x(\alpha \wedge \beta) \in F\} \cup (\alpha \wedge \beta : F) \\ &= \{\bar{x} \in S/E_F \mid x\alpha \wedge x\beta \in F\} \cup (\alpha \wedge \beta : F) \\ &= \{\bar{x} \in S/E_F \mid x\alpha \in F \text{ and } x\beta \in F\} \cup ((\alpha : F) \cap (\beta : F)) \\ &= f(\alpha) \wedge f(\beta), \end{aligned}$$

and

$$\begin{aligned} f(\alpha') &= \{\bar{x} \in S/E_F \mid x(\alpha') \in F\} \cup (\alpha' : F) \\ &= \{\bar{x} \in S/E_F \mid x\alpha' \in F\} \cup (\{1\} \setminus (\alpha : F)) \\ &= \{\bar{x} \in S/E_F \mid x\alpha \notin F\} \cup (\{1\} \setminus (\alpha : F)) \\ &= f(\alpha)'. \end{aligned}$$

Now for $s \in S$ and $\alpha \in B$, we want to show that $s\alpha$ has truth set in S_F the inverse image of the truth set of α under $\phi_F(s)$. Now let $\bar{x} \in S/E_F$. Then $\bar{x} \in f_F(s\alpha)$ if and only if $(xs)\alpha = x(s\alpha) \in F$, if and only if $\psi_s(\bar{x}) = \bar{x}s \in f_F(\alpha)$. It remains to check the fate of $1 \in S_F$. But $1 \in f_F(s\alpha)$ if and only if $s\alpha \in F$, if and only if $\psi_s(1) = s \in f_F(\alpha)$, as required. \square

Hence B acts as a Boolean algebra of predicates on S_F , in such a way that $s\alpha$ computes composites of transformations and predicates (equivalently, inverse images) as hoped.

In order to establish faithfulness, we need to show that each non-equal pair $a, b \in S$ can be “separated” by some ultrafilter. This follows easily from Lemma 2.2.

Theorem 3.4. *For $a, b \in S$ with $a \neq b$, there is an ultrafilter G of B for which $\phi_G(a) \neq \phi_G(b)$.*

Proof. Pick a, b as in the theorem statement. From Lemma 2.2, we know there is an ultrafilter G of B for which $(a, b) \notin E_G$. Then relative to G , $\psi_a(1) = \bar{a} \neq \bar{b} = \psi_b(1)$, so $\psi_a \neq \psi_b$. \square

We also want B faithfully represented.

Proposition 3.5. *For $\alpha \in B$ with $\alpha \neq F$, there is an ultrafilter G of B for which $f_G(\alpha) \neq \emptyset$.*

Proof. Let F be the filter of B generated by α (which does not contain F since $\alpha \neq F$). Then F embeds into an ultrafilter G containing α , so by definition, $1 \in f_G(\alpha)$ which is therefore non-empty. \square

The Boolean algebra B determines predicates on S_F with truth sets determined by f_F , and this induces a B-semigroup structure on $T(S_F)$.

Theorem 3.6. *Let G be an ultrafilter of B . Then for all $\alpha \in B$ and $s, t \in S$, $\alpha[s, t]$ is represented by ϕ_G as the transformation that agrees with the image of s when α is satisfied and with the image of t otherwise.*

Proof. Suppose $\alpha \in B$ with $s, t \in S$. First consider $\bar{x} \in S/E_F$. Suppose $\bar{x} \in f_G(\alpha)$, that is, $x\alpha \in G$. But $(x\alpha)[x\alpha[xs, xt], xs] = xs$, so $x \cdot \alpha[s, t] = x\alpha[xs, xt]$ is related to xs by E_G , that is,

$$\psi_{\alpha[s, t]}(\bar{x}) = \overline{x\alpha[s, t]} = \overline{xs} = \psi_s(\bar{x}).$$

Otherwise, suppose $\bar{x} \notin f_G(\alpha)$, so that $x\alpha \notin G$, so $(x\alpha)' \in G$ (since G is an ultrafilter). But $(x\alpha)'[(x\alpha)[xs, xt], xt] = (x\alpha)'[(x\alpha)'[xt, xs], xt] = xt$, so xt is related by E_G to $(x\alpha)'[xt, xs] = (x\alpha)'[xt, xs] = x \cdot \alpha'[t, s] = x \cdot \alpha[s, t]$, and so

$$\psi_{\alpha[s, t]}(\bar{x}) = \overline{x \cdot \alpha[s, t]} = \overline{xt} = \psi_t(\bar{x}).$$

Now consider 1. If $1 \in f_G(\alpha)$, that is, if $\alpha \in G$, then because $\alpha[\alpha[s, t], s] = s$, we have $(\alpha[s, t], s) \in E_G$, that is,

$$\psi_{\alpha[s, t]}(1) = \overline{\alpha[s, t]} = \overline{s} = \psi_s(1).$$

Otherwise, if $1 \notin f_G(\alpha)$, then $\alpha \notin G$, so $\alpha' \in G$ (since G is an ultrafilter). But $\alpha'[\alpha[s, t], t] = \alpha'[\alpha'[t, s], t] = t$, so $(\alpha[s, t], t) \in E_G$, that is,

$$\psi_{\alpha[s, t]}(1) = \overline{\alpha[s, t]} = \overline{t} = \psi_t(1).$$

\square

Now let F range over all ultrafilters of B . Let S_0 denote the disjoint union $\bigcup_F S_F$. Now paste together all the ϕ_F representations to give $\phi : S \rightarrow T(S_0)$, and all the f_F truth set representations of B to give $f : B \rightarrow 2^{S_0}$. (For example: for each $\bar{s} \in S_F \subseteq S_0$ we define $\phi(\bar{s}) := \psi_F(\bar{s})$.) It follows that ϕ separates any two unequal $a, b \in S$, and f sends no non-zero $\alpha \in B$ to the empty set (and hence is injective). Hence we have the following.

Theorem 3.7. *The B-semigroup (S, B) is embeddable as a two-sorted algebra into the B-semigroup $(T(S_0), 2^{S_0})$ for some set S_0 , with S_0 finite if S and B are finite.*

4. B-MONOIDS

In this section we are interested in modelling B-semigroups for which the semi-group sort has an identity element that behaves like the identity transformation.

We say that the B-semigroup (S, B) equipped with nullary operation $e \in S$ is a *B-monoid* if e is an identity element for S , satisfying the law

$$e\alpha = \alpha, \text{ for all } \alpha \in B.$$

It is clear that $(T(X), 2^X)$ is a B-monoid for any set X .

Let (S, B) be a B-monoid. Now the representation just given will not represent the identity e in S as the identity transformation because of its action on the introduced element 1: for each ultrafilter F of B , we have $\psi_e(1) = \bar{e} \neq 1$. We now sketch a representation that does work: it is very similar to the one for B-semigroups given in the last section so we omit the details.

First, in the definition of S_F , one need not introduce the new element 1 in order to ensure faithfulness of the representation: instead one defines E_F as before but sets $S_F = S/E_F$. The mapping $\phi_F : S \rightarrow T(S_F)$ is defined exactly as in Corollary 3.2 but without reference to 1, and is still a semigroup homomorphism. The definition of f_F can also be made without reference to 1, so that the $(\alpha : F)$ term is missing. The proof of the obvious analog of Theorem 3.3 proceeds as before but more easily. For Theorem 3.4, in the last line of the proof we need only replace $\psi_a(1)$ by $\psi_a(\bar{e})$ (and similarly for ψ_b). For Proposition 3.5, defining F, G as there, we note instead that $\bar{e} \in f_G(\alpha)$ since $e\alpha = \alpha \in F \subseteq G$, giving the same conclusion as before. In Theorem 3.6, the proof is simplified because 1 is not a separate case. Pasting together the representations exactly as before then gives the following variant of Theorem 3.7.

Theorem 4.1. *The B-monoid (S, B) is embeddable into a B-monoid of the form $(T(X), 2^X)$.*

This result does not imply Theorem 3.7, nor is it implied by it. An alternative strategy for the current article would have been to begin by showing that every B-semigroup can be embedded into a B-monoid: with such a result in hand, Theorem 3.7 would of course follow from Theorem 4.1. Such an embedding can indeed be achieved without any reference to representations, but is a slow if routine task, and so for brevity we have opted for the presentation given here.

5. FREE B-SEMIGROUPS AND THE EQUATIONAL PROBLEM

The class of representable B-semigroups is a two-sorted variety, as we have seen. We now solve the equational problem by describing the free algebras within this variety. First, some general observations.

In any B-semigroup (S, B) , suppose $\alpha_1, \alpha_2, \dots, \alpha_n \in B$ are such that $\sum_i \alpha_i = \mathbf{T}$ and $\alpha_i \alpha_j = \mathbf{F}$ if $i \neq j$. Then we say the α_i constitute a *partition of unity*, and for $s_1, s_2, \dots, s_n \in S$, the element $\sum_i \alpha_i s_i$, defined via

$$\sum_i \alpha_i s_i := \alpha_1[s_1, \alpha_2[s_2, \dots, \alpha_{n-1}[s_{n-1}, s_n] \dots]],$$

is a *disjoint affine combination*. Given Theorem 3.7, we may interpret $\sum_i \alpha_i s_i$ in functional terms as the transformation that agrees with s_i on α_i for each i . It follows that the order of occurrence of the terms $\alpha_i s_i$ in the formal sum does not matter. (This of course can be shown inductively from the defining laws, but follows easily from the representation theorem just shown.)

A large class of B-semigroups can be described as follows:

- there is a semigroup of transformations S on some base set X (for example polynomial functions);
- there is a collection of predicates B on X that can be used to define the piecewise-defined functions, and this collection of predicates forms a Boolean algebra and is closed under substitution of elements of S into

the predicates (for example, Boolean combinations of predicates such as $(p(x) \geq 0)$, for $p(x)$ a polynomial; this can be thought of as the functional composition of $p : X \rightarrow X$ and the test “ $x \geq 0$ ” considered as a function from X to $\{0, 1\}$).

Then our set of *disjoint affine combinations* of elements of S over B —written $\text{affine}(S, B)$ —is the set of all transformations $\sum_{i=1}^n \alpha_i(x) s_i(x)$, which denotes the transformation that agrees with $s_i(x) \in S$ when $\alpha_i(x)$ is *true*, and for which the α_i form a partition of unity. (We use the “dummy variable” x here for notational convenience, and in order to emphasise the similarity with the polynomial case.)

The set $\text{affine}(S, B)$ is closed under both composition and the **if-then-else** operations; indeed, (here reading composition of functions right to left), we have

$$\sum_i \alpha_i(x) p_i(x) \circ \sum_j \beta_j(x) p_j(x) = \sum_{i,j} (\alpha_i(p_j(x)) \wedge \beta_j(x)) p_i(p_j(x)),$$

$$\alpha \left[\sum_i \alpha_i(x) p_i(x), \sum_i \beta_i(x) p_i(x) \right] = \sum_i ((\alpha(x) \wedge \alpha_i(x)) \vee (\alpha(x)' \wedge \beta_i(x))) p_i(x).$$

Note that in order to perform these computations, we need to compute not only the composite of any two transformations in S , but also the composite of any transformation in S with any predicate in B (which we are permitted to do, by assumption). Indeed this latter two-sorted composition can easily be extended so that an element of $\text{affine}(S, B)$ can be composed with an element of B , via

$$\alpha \circ \sum_j \beta_j(x) p_j(x) = \bigvee_j (\beta_j(x) \wedge \alpha(p_j(x))).$$

In fact, $(\text{affine}(S, B), B)$ is a B-semigroup, and Theorem 3.7 shows that every B-semigroup arises in this way: with ϕ and f as defined there, (S, B) is isomorphic to $(\text{affine}(\phi(S), f(B)), f(B))$ (and as it happens, one also has $\text{affine}(\phi(S), f(B)) = \phi(S)$).

Free B -sets in the sense of Bergman [1] (in which the Boolean ring B is fixed) are exactly (bounded) *Boolean powers*, and admit a description quite similar to that given for $\text{affine}(S, B)$ above, though possessing only the **if-then-else** operations. Boolean powers can be viewed as the special case in which the elements of S are all constant functions.

As is well known, any Boolean algebra can be viewed as a Boolean ring with identity, with addition defined via $\alpha + \beta := (\alpha \vee \beta) \wedge (\alpha \wedge \beta)'$, and multiplication just the Boolean algebra meet; then $\alpha' = \mathbf{T} + \alpha$. Indeed the varieties are term equivalent. From now on, we shall freely view all Boolean algebras as Boolean rings in this section.

We now use the exclusive affine combination idea in order to describe free B-semigroups. Thus let X be a set of semigroup variables, with X^+ the free semigroup on X . Let A be a set of Boolean variables, A_X the union of A together with all distinct expressions of the form $\square_t[\alpha]$ where $t \in X^+$ and $\alpha \in A$. Let $B_{A,X}$ be the free Boolean ring generated by A_X . Let $M(X, A)$ be the free unital $B_{A,X}$ -module on the generating set X^+ , and let

$$\text{Fr}(X, A) = \left\{ \sum_i \alpha_i t_i \mid \alpha_i \in B_{A,X}, t_i \in X^+ \text{ and the } \alpha_i \text{ are a partition of unity} \right\}.$$

This set will be the underlying semigroup sort in our construction of the free B-semigroup on (X, A) , while $B_{A,X}$ will be the underlying Boolean ring/algebra sort. It remains to define the various remaining (non-Boolean) operations.

It is easy to check that $(\text{Fr}(X, A), B_{A,X})$ is a B-set if one defines

$$\alpha[x, y] := \alpha x + \alpha' y, \quad x, y \in \text{Fr}(X, A), \quad \alpha \in B_{A,X};$$

indeed it is the *Boolean power of X^+* by $B_{A,X}$, as is discussed in [1] and [21]. (In Bergman's sense, it is the free $B_{A,X}$ -set on the generating set X^+ .)

Now for each $t \in X^+$, we first show how to define the operation $\square_t : B_{A,X} \rightarrow B_{A,X}$, which models the composition operation between transformations and predicates (that is, $\square_t(\alpha)$ is the operation corresponding to $t\alpha$).

- For $\alpha \in A$, define $\square_t(\alpha) = \square_t[\alpha]$.
- For $\square_s[\alpha] \in A_X$, define $\square_t(\square_s[\alpha]) = \square_{ts}[\alpha]$.
- Extend the definition of \square_t to all of $B_{A,X}$ by specifying that it be the unique homomorphism $B_{A,X} \rightarrow B_{A,X}$ acting on the generators A_X as just indicated. (Recall that $B_{A,X}$ is free on A_X . Equivalently, to find the action of \square_t on an element β of $B_{A,X}$, just replace all occurrences of a Boolean variable α in β by $\square_t[\alpha]$ and of $\square_s[\alpha]$ by $\square_{ts}[\alpha]$.)

Next we extend the definition of \square so that its suffix argument can take any value in $\text{Fr}(X, A)$ (and not just X^+); define, for any $\beta \in B_{A,X}$ and any $\sum_i \alpha_i t_i \in \text{Fr}(X, A)$,

$$\square_{\sum_i \alpha_i t_i}(\beta) = \sum_i \alpha_i \square_{t_i}(\beta).$$

This is well-defined because it is nothing but a B-set homomorphism from the free B-set on the generators X^+ over the Boolean ring $B_{A,X}$ defined in terms of its action on the generators.

It remains to define the semigroup multiplication on $\text{Fr}(X, A)$. This is prompted by the discussion above. For $\sum_i \alpha_i t_i, \sum_j \beta_j t_j \in \text{Fr}(X, A)$ (with the t_i the same elements of X^+ in both, with some α_i or β_j zero as necessary), define

$$\left(\sum_i \alpha_i t_i\right) \cdot \left(\sum_j \beta_j t_j\right) = \sum_{i,j} \alpha_i \square_{t_i}(\beta_j) t_i t_j,$$

a definition independent of the particular way of writing $\sum_i \alpha_i t_i$, and $\sum_j \beta_j t_j$ because of the fact that $\square_t(\alpha_1 + \alpha_2) = \square_t(\alpha_1) + \square_t(\alpha_2)$. Note that the result is in $\text{Fr}(X, A)$, because the $\alpha_i \square_{t_i}(\beta_j)$ form a partition of unity as is easily checked.

Theorem 5.1. *$(\text{Fr}(X, A), B_{A,X})$ is the free B-semigroup on the semigroup generators X and Boolean generators A .*

Proof. We need to verify all the B-semigroup laws. Each is quite routine to check and this is left to the reader, although along the way it is necessary to show that for $s, t \in X^+$, $\square_s(\square_t(\alpha)) = \square_{st}(\alpha)$ for any $\alpha \in B_{A,X}$; but this is also easily shown by induction, based on the fact that it is true if $\alpha \in A$.

The fact that $(\text{Fr}(X, A), B_{A,X})$ is free follows from the fact that only the laws holding in all B-semigroups have been used to define the various operations on it. The unconvinced reader is again welcome to plough through the routine formal proof, showing that any function from the generators of each sort to the respective sorts of a B-semigroup (S, B) extends uniquely to a homomorphism $(\text{Fr}(X, A), B_{A,X}) \rightarrow (S, B)$. \square

It is straightforward to extend the previous construction to cover free B-monoids as well. We simply replace the free semigroup on X with the free monoid X^* on X (with empty word denoted 1), and modify the definition of $\text{Fr}(X, A)$ by replacing X^+ by X^* , thereby giving the larger set $\text{Fr}'(X, A)$. The definitions of A_X and $B_{A,X}$ are not altered, the operation $\square_1 : B_{A,X} \rightarrow B_{A,X}$ defined by simply setting $\square_1(\alpha) = \alpha$ for all $\alpha \in A$. The rest of the construction proceeds exactly as before. We thus obtain the following.

Theorem 5.2. *($\text{Fr}'(X, A), B_{A,X}$) is the free B-monoid on the monoid generators X and Boolean generators A .*

6. NON-FINITE AXIOMATIZABILITY OF **IF-THEN-ELSE** WITHOUT \square .

Although composition of transformations with predicates is natural in the setting of **if-then-else** with composition, there is nevertheless interest in the situation in which it is not present. The resulting class is just the class of subreducts of the B-semigroups of transformations of the form $(T(X), 2^X)$, so the solution to the word problem in the free algebras is essentially covered by the results of Section 5. Nevertheless, in this section we show that no finite axiomatization is possible for the class. The main theorem of this section is as follows.

Theorem 6.1. *In each of the following cases there is no finite system of first order sentences to characterise the representable algebras:*

- *semigroups/monoids of transformations with **if-then-else**;*
- *semigroups/monoids of partial maps with **if-then-else** and 0;*
- *semigroups/monoids of binary relations with **if-then-else** and 0.*

Here the monoid identity element (denoted by 1) is to be represented as the diagonal relation, while 0 is to be represented as the empty map. We mention also that each of the cases is easily seen to form a quasivariety (closed under isomorphisms, ultraproducts, direct products and substructures).

We concentrate on the transformation semigroup case. The non-monoid cases follow with small adjustments, while the monoid cases are a little messier but similar and we consider them last.

We begin the nonfinite basis proof by constructing an infinite transformation semigroup \mathbf{T} with **if-then-else**. We then use its structure to define, for each $n > 2$, a two-sorted structure \mathbf{T}_n that is not isomorphic to a transformation semigroup with **if-then-else**, but such that all n -generated substructures of \mathbf{T}_{4n+1} are isomorphic to a transformation semigroup with **if-then-else**: in fact they embed into \mathbf{T} .

6.1. Constructing \mathbf{T} . Let \mathbb{Z} denote the integers and let $Z := \mathbb{Z} \cup \{\infty\}$, with the usual order on \mathbb{Z} and with $i \leq \infty$ for every $i \in \mathbb{Z}$. We consider transformations on the set Z^3 .

We first consider a map z that constantly maps Z^3 to (∞, ∞, ∞) . The following maps (indexed by $i \in \mathbb{Z}$) also send elements of Z^3 to the point (∞, ∞, ∞) unless defined otherwise.

- $f_i : (j, \infty, \infty) \mapsto (j, i, \infty)$ when $j \leq i$.
- $g_i^b : (j, i, \infty) \mapsto (\infty, \infty, j)$ when $j < i$.
- $g_i^{\sharp} : (i, i, \infty) \mapsto (\infty, \infty, i)$.
- $h_i : (j, \infty, \infty) \mapsto (\infty, \infty, j)$ when $j \leq i$.

	g_i^b	g_i^\sharp	$s_{i,j}$	$g_{i,j}$	$g_{i,i}$	$g_{j,i}$	$t_{i,i}$	$t_{i,j}$	$u_{i,i}$	$u_{i,j}$
f_i	h_{i-1}	r_i	h_{i-1}	h_{i-1}	h_i	r_i	h_{i-1}	h_{i-1}	h_{i-1}	h_{i-1}
$s_{j,i}$	h_{i-1}	r_i	h_{i-1}	h_{i-1}	h_i	r_i	h_{i-1}	h_{i-1}	h_{i-1}	h_{i-1}

TABLE 1. Non- z products amongst F_1 .

- $r_i : (i, \infty, \infty) \mapsto (\infty, \infty, i)$.

Notice that $f_i g_i^b = h_{i-1}$ and $f_i g_i^\sharp = r_i$. All other products between the transformations are equal to z , so these form a subsemigroup of the full transformation semigroup on Z^3 . Let us denote the family of transformations considered so far by F_0 .

Remark 6.2. *If the point (∞, ∞, ∞) is removed from Z^3 , we can amend each transformation in F_0 to a partial function. If G_0 denotes this family of partial functions on $Z^3 \setminus \{(\infty, \infty, \infty)\}$ then it is easily seen that G_0 is a semigroup (under composition) isomorphic to F_0 .*

Now let us define a subset α of Z^3 by $\alpha := \{(j, i, \infty) \mid j < i < \infty\}$. If we generate tests from α , we obtain only α' , $F := \emptyset$ and $T := Z^3$. Observe that every map in F_0 except for those of the form g_i^b agree with z on α , while the maps g_i^b agree with z on α' . Hence if we make applications of **if-then-else** to the transformations in F_0 , then we obtain only the following new transformations (that is, not already in F_0):

- $\alpha[g_i^b, f_j] =: s_{i,j}$;
- $\alpha[g_i^b, g_j^\sharp] =: g_{i,j}$;
- $\alpha[g_i^b, h_j] =: t_{i,j}$;
- $\alpha[g_i^b, r_j] =: u_{i,j}$.

Let us denote the union of F_0 with the new set of transformations by F_1 .

Closure of F_1 under multiplication produces no new transformations. Table 1 summarises products amongst elements of F_i that do not produce the value z . The numbers i and j are arbitrary but distinct elements of \mathbb{Z} . Every product of elements of F_1 that does not take the value z can be found by taking a suitable choice of i and j . For example, $s_{5,-3}t_{-3,5}$ is not equal to z because if we choose $i = -3$ and $j = 5$ we obtain $s_{j,i}t_{i,j} = h_{i-1} = h_{-4}$. On the other hand $s_{2,3}t_{2,2}$ is equal to z : to find $s_{2,3}$ on the left of a product in the table one needs $j = 2$ but then (as $i \neq j$) there is no column for $t_{j,j}$.

Thus, when tests are chosen from the four element Boolean algebra of subsets $\{\alpha, \alpha', F, T\}$, the system F_1 forms a transformation semigroup with **if-then-else**. This is \mathbf{T} .

Remark 6.3. *(Continuing from Note 6.2.) One may similarly close the set G_0 under applications of $\alpha[\ , \]$, and produce a semigroup of partial functions on $Z^3 \setminus \{(\infty, \infty, \infty)\}$ with **if-then-else** that is isomorphic to \mathbf{T} .*

6.2. Constructing \mathbf{T}_n . Table 1 provides us with a purely abstract definition of \mathbf{T} . For each $n > 2$, consider the alphabet F_n of symbols

$$\{z\} \cup \{\bar{f}_i, \bar{g}_i^\sharp, \bar{g}_i^b, \bar{g}_{i,j}, \bar{h}_i, \bar{r}_i, \bar{s}_{i,j}, \bar{t}_{i,j}, \bar{u}_{i,j} \mid i, j \in \mathbb{Z}_n\}.$$

Of course, this alphabet has been chosen in very close analogy to the names of the transformations in F_1 . We will use Table 1 to define a product between the elements

of this alphabet in an obvious way: $\bar{x} \cdot \bar{y} = \overline{xy}$, where \cdot is the new multiplication and xy is just the composition of x with y in \mathbf{T} , except that subtraction in the subscripts is to be performed modulo n . For example $\bar{f}_0 \bar{g}_{0,2} = \bar{h}_{n-1}$. Now we want to add a new Boolean sort B with four elements, just as in \mathbf{T} . Since this part is genuinely isomorphic to the Boolean sort of \mathbf{T} , we abuse notation and use the same symbols $\alpha, \alpha', F, \bar{\cdot}$. We can again use the abstract behaviour of **if-then-else** on \mathbf{T} to define a ternary operation $\bar{_}[_, _]: B \times F_n \times F_n \rightarrow F_n$ by setting $\alpha[\bar{x}, \bar{y}] := \overline{\alpha[x, y]}$ (where $\alpha[x, y]$ is calculated in \mathbf{T}). This is easily seen to be well defined. The corresponding two-sorted algebra is denoted by \mathbf{T}_n .

6.3. The nonfinite basis proof.

Lemma 6.4. *\mathbf{T}_n is not faithfully representable as a transformation semigroup with **if-then-else**. Neither is it faithfully representable as a semigroup of binary relations with **if-then-else**, provided z is required to be represented as the empty map \emptyset .*

Proof. Assume that \mathbf{T}_n has been represented (under a representation $\bar{x} \mapsto \hat{x}$) as a system of transformations on a set X , with the Boolean sort B a Boolean algebra of subsets of B and with $\bar{_}[_, _]$ acting as **if-then-else**. We show that this representation cannot be faithful. Now let $a \in X$ be such that $\hat{h}_i(a) \neq \hat{z}(a)$; say $\hat{h}_i(a) = b$ (if there are no such $a \in X$, then the representation is not faithful, as claimed). We show that $\hat{h}_i(a) = \hat{h}_{i+1}(a)$ and then by applying induction over \mathbb{Z}_n we obtain that all of the \hat{h}_i act identically. Hence the representation cannot be faithful.

Now, $\hat{f}_i \hat{g}_{i,i} = \hat{h}_i = \hat{f}_{i+1} \cdot \alpha[\hat{g}_{i+1,i+1}, \hat{z}]$. If $(a) \hat{f}_{i+1} \notin \alpha$, then

$$\hat{f}_{i+1} \cdot \alpha[\hat{g}_{i+1,i+1}, \hat{z}](a) = \alpha[\hat{g}_{i+1,i+1}, \hat{z}](\hat{f}_{i+1}(a)) = \hat{z}(\hat{f}_{i+1}(a)) = \hat{f}_{i+1} \hat{z}(a) = \hat{z}(a),$$

a contradiction. Hence $\hat{f}_{i+1}(a) \in \alpha$ and then $\hat{f}_{i+1} \hat{g}_{i+1,i+1}(a) = \hat{g}_{i+1,i+1}(\hat{f}_{i+1}(a)) = b$. So $\hat{h}_{i+1}(a) = \hat{f}_{i+1} \hat{g}_{i+1,i+1}(a) = b$, as required.

The binary relation case is similar: consider a pair $(a, b) \in \hat{h}_i$, and use the fact that $\hat{z} = \emptyset$ to show that $(a, b) \in \hat{h}_{i+1}$. \square

Lemma 6.5. *Every n -generated substructure of \mathbf{T}_{4n+1} is faithfully representable as an **if-then-else** semigroup of transformations on a set (or as partial maps on a set, in which z is represented as the empty map).*

Proof. Let \mathbf{S} be a substructure of \mathbf{T}_{4n+1} generated by at most n elements. Since each element brings with it either none, one or two numerical subscripts, there are at most $2n$ numerical subscripts involved in the generators of \mathbf{S} . Since there are $4n + 1$ individual numerical subscripts available, it follows there must be two consecutive numbers that do not appear in the subscripts of any generator for \mathbf{S} . By applying the obvious automorphism of \mathbf{T}_{4n+1} that increments the indices modulo $4n + 1$, we may assume that these two consecutive numbers are $4n$ and 0 . Then \mathbf{S} is a substructure of the substructure on

$$\{\bar{z}\} \cup \{\bar{f}_i, \bar{g}_i^\#, \bar{g}_i^\flat, \bar{g}_{i,j}, \bar{h}_i, \bar{r}_i, \bar{s}_{i,j}, \bar{t}_{i,j}, \bar{u}_{i,j} \mid 1 \leq i < 4n\} \cup \{\bar{h}_0, \bar{t}_{i,0} \mid 0 \leq i < 4n\}$$

and including all four Boolean tests. By consulting Table 1 it is routine to verify that this algebra is isomorphic to the substructure of \mathbf{T} on the elements of the same name (just drop the bar over each symbol), hence is faithfully representable. \square

The following theorem is the transformation semigroup part of Theorem 6.1.

Theorem 6.6. *There is no finite system of quasi-identities characterising the quasivariety of all transformation semigroups with **if-then-else**.*

Proof. Let Σ be any finite set of universal sentences holding on the class of all transformation semigroups with **if-then-else**. Trivially, Σ involves only finitely many variables; say n . Then \mathbf{T}_{4n+1} satisfies Σ , by Lemma 6.5, and since \mathbf{T}_{4n+1} is not isomorphic to any transformation semigroup with **if-then-else** (by Lemma 6.4), Σ is not a complete system of axioms. \square

Next is the semigroup of partial maps or binary relations part of Theorem 6.1.

Theorem 6.7. *There is no finite system of quasi-identities characterising the quasivariety of all semigroups of partial maps or of general binary relations with **if-then-else** and 0 (representing the empty map).*

Proof. The same proof as for Theorem 6.6, but using Note 6.3, and the fact that \mathbf{T} may be represented as partial maps, but \mathbf{T}_n not even as general binary relations. \square

6.4. Nonfinite axiomatizability in the monoid case. The underlying semigroup of transformations in \mathbf{T} is not a monoid, so we cannot use it directly to prove nonfinite axiomatizability of monoids with **if-then-else**. Nevertheless, we can proceed by adjoining the identity element to the set of generating transformations for T . In this subsection we give a sketch of the argument, leaving details to the reader.

Rather than adjoin 1 directly, we instead adjoin two elements a and \bar{a} to the set F_0 . The element a acts as the identity on α and is equal to z on α' . The function \bar{a} is the dual, acting as z on α and 1 on α' . (Of course the full identity element 1 will be generated as $\alpha[a, \bar{a}]$.) If we close $F_0 \cup \{a, \bar{a}\}$ under composition, only two new kinds of elements are generated, namely $f_i a$ and $f_i \bar{a}$ (for each $i \in \mathbb{Z}$). We denote the set of generated elements by F_0^+ . It is also convenient to follow the idea of Note 6.2 and write G_0^+ for the partial map version of F_0^+ . We use the same names for the partial maps in G_0^+ as for transformations in F_0^+ .

Next we close under applications of $_{-}[\ , \]$. As before, we can restrict to applications of the form $\alpha[x, y]$ for $x, y \in F_0^+$, since nested applications of this operation (or ones using the test α') can be rearranged to be of this form. Aside from the element 1 and the four groups $s_{i,j}, g_{i,j}, t_{i,j}, u_{i,j}$ generated from F_0 , we also have elements of the following new kinds of elements:

- $\alpha[a, x]$ for $x \in \{f_i \bar{a}, f_i a, f_i, g_i^\#, h_i, r_i\}$ and for each $i \in \mathbb{Z}$;
- $\alpha[g_i^b, \bar{a}]$.

All other combinations produce elements already in F_0^+ . Let F_1^+ denote the set of all transformations generated so far. We next claim that F_1^+ is closed under composition.

To see why this is true, it is easier to consider the partial map version of F_1^+ , which we refer to as G_1^+ . The partial map case has the simple advantage that one is able to talk of functions “mapping into α ” (otherwise all maps agree with z on at least some points of both α and its complement). Of course, all of the

following statements can be translated back to the transformation setting, but the explanation is more cumbersome.

Now, to verify that G_1^+ is closed under composition, first notice that for any $x \in G_0^+$ except for $x \in \{f_i \mid i \in \mathbb{Z}\}$ we have that x either maps entirely into α or entirely into α' . Thus $x \cdot \alpha[u, v]$ either equals xu or xv . In the case of f_i we have $f_i \cdot \alpha[u, v] = f_i \alpha u \cup f_i \bar{\alpha} v$. Now, if one of $f_i \alpha u$ or $f_i \bar{\alpha} v$ is empty, then $f_i \alpha u \cup f_i \bar{\alpha} v$ is equal to an element of $G_0^+ \subseteq G_1^+$; namely z , $f_i \alpha u$ or $f_i \bar{\alpha} v$. If both $f_i \alpha u$ and $f_i \bar{\alpha} v$ are nonempty then it must be that $u = g_i^b$, and $v = g_i^\sharp$, in which case $f_i \cdot \alpha[u, v] = h_i \in G_0^+$.

Now consider any two elements $\alpha[x_1, y_1]$ and $\alpha[x_2, y_2]$ in G_1^+ , where $x_1, x_2, y_1, y_2 \in G_0^+$. (All elements of G_1^+ can be written in this form since $x = \alpha[x, x]$ always.) We have that $\alpha[x_1, y_1] \alpha[x_2, y_2] = \alpha[x_1 \cdot \alpha[x_2, y_2], y_1 \cdot \alpha[x_2, y_2]]$, and by the above argument both $x_1 \cdot \alpha[x_2, y_2]$ and $y_1 \cdot \alpha[x_2, y_2]$ are contained in G_0^+ . Hence $\alpha[x_1, y_1] \cdot \alpha[x_2, y_2] = \alpha[x_1 \cdot \alpha[x_2, y_2], y_1 \cdot \alpha[x_2, y_2]] \in G_1^+$, as claimed. Thus we can let \mathbf{T}^+ denote the **if-then-else** monoid on G_1^+ (or on F_1^+ , depending on whether we want partial maps or transformations).

Now one needs to be able to define the monoid “version” of the finite structures \mathbf{T}_n . We do this the same way: by defining everything according to their behaviour in \mathbf{T} except that the indices from \mathbb{Z} are to be interpreted modulo n . To establish the validity of this definition, one could in principle build up a table extending Table 1. However this is tedious and unnecessary, since the two required features of such a table are clear:

- multiplication (and applications of $\alpha[\ , \]$) between elements is not affected by any ordering of the indices, only by whether or not they coincide;
- if $S \subseteq F_0^+ \cup \{1\}$ is a finite set of generators, and $S_{\mathbb{Z}}$ is the finite subset of \mathbb{Z} consisting of all indices of elements of S , then every index i appearing in the subalgebra generated by S is either contained in $S_{\mathbb{Z}}$ or is one less than an element of $S_{\mathbb{Z}}$.

The rest of the proof carries over without any change, giving the monoid versions of Theorems 6.6 and 6.7, and hence the remaining cases of the main Theorem 6.1.

6.5. Adjoining while. Another very common programming construction beyond **if-then-else** is the operation of **while**. Indeed, the language of “while programs” is frequently considered as a basic formal language under which algorithms can be described: at the abstract level it consists precisely of program composition, **if-then-else**, and **while**. The **while** construction inevitably leads to programs that do not halt on all inputs: for example, if the identity element 1 is to be included (the **skip** program for example), then the program “**while T do 1**” halts on no inputs at all; under the relational semantics for programs as binary relations, this means that **while T do 1** is the empty map. In particular, this shows that it does no harm to assume the presence of a nullary 0 representing the empty map. We now wish to observe how the methods of the present section can be extended to include **while** in the partial map (modelling deterministic programs), or general binary relation (modelling nondeterministic programs) setting.

First observe that \mathbf{T}^+ satisfies the equation $x^2 = x^3$, as is easily verified. In this case it is also routinely verified that the **while** operation is already represented as a term function, namely

$$\mathbf{while} \ \beta \ \mathbf{do} \ x = \beta[x \cdot \beta[x \cdot \beta[0, 1], 1], 1],$$

since if β is true after both the first and second applications of x , then β will be true after any number of applications of x , and **while** β **do** x is undefined at such a point (that is, acts like 0).

Hence one can adjoin the operation of **while** to \mathbf{T}^+ , defined in accordance with the rule just observed, and the main proofs of this section continue to hold.

Corollary 6.8. *The universal theory of algebras of functions (or binary relations) with **if-then-else**, **while** and 1 has no finite axiomatization. Furthermore, there is no finite set of first order universal sentences characterising the isomorphic copies of the finite algebras of partial maps or of binary relations.*

In [9] the authors showed that in the partial map case, if one additionally includes the dynamic logic operators $[p]\beta$ and $\langle p \rangle\beta$ (which are essentially the analogues of the predicate transformer corresponding to composition of transformations with predicates, as considered in earlier sections), then the relevant class of finite algebras can be characterised up to isomorphism by a finite system of implications.

7. SPECIAL KINDS OF TESTS IN B-SEMIGROUPS

7.1. Equality tests. A B-semigroup (S, B) is *agreeable* if it additionally has an operation $*$: $S \times S \rightarrow B$ making it an agreeable B-set and satisfying the law

$$s(t * u) = (st * su).$$

$(T(X), 2^X)$ (where X is a set) is an agreeable B-semigroup if $*$ is defined as in Section 2. An *agreeable B-monoid* is an agreeable B-semigroup which is a B-monoid; again $(T(X), 2^X)$ is an example.

Theorem 7.1. *Every agreeable B-semigroup/monoid can be embedded in one of the form $(T(X), 2^X)$ for some set X .*

Proof. Let (S, B) be an agreeable B-semigroup. We shall show that the representation described prior to the statement of Theorem 3.7 also represents $*$ correctly. We need only do this for a particular (ϕ_F, f_F) pair.

Now for $\bar{x} \in S/E_F$, $\bar{x} \in f_F(a * b)$ if and only if $(xa * xb) = x \wedge (a * b) \in F$, if and only if $(xa, xb) \in E_F$, if and only if $\psi_a(\bar{x}) = \bar{x}a = \bar{x}b = \psi_b(\bar{x})$, if and only if $\bar{x} \in \phi_F(a) * \phi_F(b)$. Regarding 1 now, note that $1 \in f_F(a * b)$ if and only if $a * b \in F$, if and only if $(a, b) \in E_F$, if and only if $\psi_a(1) = \bar{a} = \bar{b} = \psi_b(1)$, if and only if $1 \in \phi_F(a) * \phi_F(b)$. Hence $f_F(a * b) = \phi_F(a) * \phi_F(b)$ as asserted.

The proof of the monoid case is just a simplified version of the non-monoid case just presented, in which the special case of the added-in element 1 need not be considered. \square

There is of course interest in free agreeable B-semigroups and the equational problem, as well as the question of axiomatizability in the absence of transformation-predicate composition. We conjecture that the equational problem has a similar solution to that presented for B-semigroups in Section 5. (An analogous case for partial functions with domain and $*$ has such a solution: see [8].) But we observe that the non-finite basis argument given in Section 6 will not carry over to the agreeable case. We therefore pose as an open problem the question of finite axiomatizability of transformation semigroups with **if-then-else** and equality tests.

7.2. Fixed point tests. Closely related to the equality test is the *fixed point test*. For a transformation $f \in T(X)$, define $I(f) \in 2^X$ as follows:

$$I(f) = \{x \in X \mid f(x) = x\}.$$

(Again, strictly we define it to be the predicate on X having this truth set.) In an agreeable B-monoid (S, B) , this operation is already modelled: $I(a) = a * 1$, where 1 is the identity element of S . However, it can be modelled independently of $*$.

The operation I has been considered in a general semigroup setting and in a partial function setting in the papers [11] and [10], where it is called an “interior operation” because of its formal resemblance to an interior operator in topology.

In contrast to $*$, there is no “flat” version of I : it is inherently related to transformations. Also note that the B-semigroup representation of Section 3 does not represent I correctly, because the added-in element 1 is not fixed by any ψ_a . Consequently we restrict attention to B-monoids, leaving the more general case for future work.

A B-monoid (S, B) is *interior* if it is equipped with an operation $I : S \rightarrow B$ satisfying the following equivalence: for all $s, t \in S$ and $\alpha \in B$:

$$\alpha[st, s] = s \Leftrightarrow \alpha \leq s \cdot I(t).$$

Hence the class of interior B-monoids is a finitely based quasivariety, clearly containing every example of the form $(T(X), 2^X)$.

Proposition 7.2. *For F a filter of B , $(ab, a) \in E_F$ if and only if $a \cdot I(b) \in F$.*

Proof. Suppose $(ab, a) \in E_F$; then $\alpha[ab, a] = a$ for some $\alpha \in F$ so $\alpha \leq a \cdot I(b)$, and so $a \cdot I(b) \in F$. Conversely, if $a \cdot I(b) \in F$, then $(a \cdot I(b))[a, b] = b$ shows that $(a, b) \in E_F$. \square

Theorem 7.3. *Every interior B-monoid can be embedded in one of the form $(T(X), 2^X)$ for some set X .*

Proof. We show that the representation (ϕ, f) of Section 4 represents $I(a)$ correctly as the fixed set of the image of a . We need only consider a particular (ϕ_F, f_F) pair. Now for $\bar{x} \in S/E_F$, $\bar{x} \in f_F(I(a))$ if and only if $xI(a) \in F$, if and only if $(xa, x) \in E_F$, if and only if $\psi_a(\bar{x}) = \bar{x}\bar{a} = \bar{x}$, if and only if $\bar{x} \in I(\phi_F(a))$. \square

REFERENCES

- [1] G.M. Bergman, *Actions of Boolean rings on sets*, Algebra Univers. **28** (1991), 153–187.
- [2] S.L. Bloom and R. Tindell, *Varieties of “if-then-else”*, SIAM J. Comput. **12** (1983), 677–707.
- [3] M.R. Bulmer, D. Fearnley-Sander and T. Stokes, *Towards a calculus of algorithms*, Bull. Austral. Math. Soc. **50** (1994), 81–89.
- [4] E. W. Dijkstra, *Guarded commands, nondeterminacy and formal derivation of programs*, Comm. ACM **18** (1975), 453–457.
- [5] I. Guessarian and J. Meseguer, *On the axiomatization of “if-then-else”*, SIAM J. Comput. **16** (1987), 332–357.
- [6] C.A.R. Hoare, *An axiomatic basis for computer programming*, Comm. Assoc. Comput. Mach. **12** (1969) 576–580.
- [7] M. Jackson and T. Stokes, *Agreeable semigroups*, J. Algebra **266** (2003), 393–417.
- [8] M. Jackson and T. Stokes, *Identities in the algebra of partial maps*, Internat. J. Algebra Comput. **16** (2006), 1131–1159.
- [9] M. Jackson and T. Stokes, *Towards an algebra of functions and deterministic computation*, submitted.

- [10] M. Jackson and T. Stokes, *Algebras of partial maps*, to appear in *Proceedings of the Special Interest Meeting on Semigroups and Related Mathematics*, University of Sydney, 2005.
- [11] A.V. Kelarev and T. Stokes, *Interior algebras and varieties*, *J. Algebra* **221** (1999), 50–59.
- [12] D.C. Kozen, *On Hoare Logic and Kleene algebra with Tests*, *ACM Trans. Comput. Logic* **1** (2000), 60–76.
- [13] B. Möller and G. Struth, *Algebras of modal operators and partial correctness*, *Theoret. Comp. Sci.* **351** (2006), 221–239.
- [14] E.G. Manes, *Adas and the equational theory of if-then-else*, *Algebra Univers.* **30** (1993), 373–394.
- [15] J. McCarthy, *A basis for a mathematical theory of computation*, in P. Braffort and D. Hirschberg (eds.), *Computer Programming and Formal Systems*. North-Holland (1963), 33–70.
- [16] A.H. Meklar and E.M. Nelson, *Equational bases for if-then-else*, *SIAM J. Comput.* **16** (1987), 465–485.
- [17] D. Pigozzi, *Equality-Test and If-Then-Else Algebras: Axiomatization and Specification*, *SIAM J. Comput.* **20** (1991), 766–805.
- [18] V.R. Pratt, *Dynamic algebras: Examples, constructions, applications*, Technical Report MIT/LCS/TM-138, M.I.T. Laboratory for Computer Science, July 1979.
- [19] B.M. Schein, *Relation algebras and function semigroups*, *Semigroup Forum* **1** (1970), 1–62.
- [20] B.M. Schein, *Lectures on semigroups of transformations*, *Amer. Math. Soc. Translat. Ser. 2.* **113** (1979), 123–181.
- [21] T. Stokes, *Sets with B-action and linear algebra*, *Algebra Univ.* **39** (1998), 31–43.
- [22] T. Stokes, *On Eq-monoids*, *Acta Sci. Math. (Szeged)* **72** (2006), 481–506.

DEPARTMENT OF MATHEMATICS, LA TROBE UNIVERSITY, MELBOURNE, VICTORIA 3086, AUSTRALIA

E-mail address: `m.g.jackson@latrobe.edu.au`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WAIKATO, HAMILTON, NEW ZEALAND

E-mail address: `stokes@math.waikato.ac.nz`