**Abstract**

This paper will use the heuristics of a finite Boolean algebra with operators to examine a restricted family of setwise stabilizers of 2-modular representations of the symmetric group with which polynomial time algorithms might be possible. The key virtue of the approach will be to use single-voice musical passages (modeled as partial algebras over a free semigroup) to help the search.

# 1 Notation

Throughout this article, when it's clear that a field is being expected, I will denote, GF($k$) by $k$, so that the binary matrix algebra will be denoted $2^{n \times n}$. In addition, $n^{\{\}}$ will denote $\{1, \cdots, n\}$. It will be used both as a set and list (tuple). Oftentimes, I will place it in the indices of a matrix or Kronecker-delta to imply that the semantic position at which I place it will denote a free variable in the remaining function, whose image over $n^{\{\}}$ will be the final list. For example, $a_{in^{\{\}}} := (a_{i1}, a_{i2}, \ldots, a_{in})$. When convention will call to start at 0, I'll use the notation $n_0^{\{\}} := \{0, \cdots, n-1\}$ An $n$-bit-vector $v$ of atoms in a BA $2^n$ will correspond to its Boolean sum where needed.

The existence of an operator $f : (2^n)^a \to 2^n$ uniquely implies the existence of two more fundamental forms of a function on the generators (atoms), ${}^B f : n^{\{\}a} \to 2^n$ and ${}^V f : n^{\{\}a} \to 2^n$, where the first instance of $2^n$ refers to the Boolean algebra and the second instance of $2^n$ refers to the binary vector space. It will always be clear which is intended by the superscript of the function.

We will use *elementary Boolean operators* quite frequently, and by this I mean, a map $f : n^{\{\}a} \to n^{\{\}}$ which maps a single element in the Cartesian power $(x_1, \ldots, x_n)$ to $x$, (where the $x_i$s and $x$ are atom positive integers), and maps the rest to 0. We will denote such a function by $(x_1, \ldots, x_n) \mapsto_n x$, and drop the subscript if it's clear that the width is $n$.

We will only be interested in finite collections of operators of finite arity, and will utilize the multiplicity function $\mu : \mathbb{N}_0 \to \mathbb{N}_0$, whereby $\mu(a) = r$ if the BAO has $r$ operators of arity $a$. We will always assume that $\mu$ is zero on all but a finite number of elements. When we write $op^a_\mu(i)$ to denote the *ith* operator of arity $a$, we assume that $0 < i \leq \mu(a)$.

Whenever we are dealing with elements $v, w \in 2^n$, we will abbreviate the "dot product" by $v \cdot w := (v_1 \times w_1 + v_2 \times w_2 + \cdots + v_n \times w_n)$ where $v := (v_1, \ldots, v_n)$ and $w := (w_1, \ldots, w_n)$. Also, if we omit the dot, we will mean the binary vector operation, $vw := (v_1 \times w_1, v_2 \times w_2, \cdots, v_n \times w_n)$. As it turns out, Boolean multiplication will have to be written, as above, without the dot, to keep consistent.

# 2 Introduction

Let $n$ be some natural number, and consider a finite Boolean algebra $2^n$ with some multiplicity signature. It's easy to see that the automorphism group of the universe of $2^n$ is isomorphic to $S_n$, and then that raises the question of which permutations on the atoms induce which changes in the operators. The case of seeing which automorphisms respect the operators can be determined by looking at the automorphism group of the full structure (not just the universe) and won't be treated of here, as it is probably treated of elsewhere. However, if we examine which permutations of the atoms map one operator to another (assuming a "large" set of operators), things get more interesting, as now we are in a position to examine this using 2-modular representations that act on finite subspaces of $\bigoplus_{a=0}^{\infty} \mathbf{Op}_a^n$, where $\mathbf{Op}_a^n := 2^{n^{a+1}}$, the binary vector space. This is so defined because there are $n^{a+1}$ elementary Boolean operators of arity $a$, since each operator is determined by the single element domain $a$-vector and the image. We will verify that they obey the laws of a binary vector space, (or better, 2-algebra) below.

# 3 Boolean Operator Algebra

To verify that the elementary Boolean operators do indeed generate a 2-algebra, we proceed as follows : First, we recall that, in any binary field (a field of characteristic 2), addition corresponds to pointwise Boolean exclusive disjunction of the bit-vectors, multiplication corresponds to pointwise join, minus corresponds to pointwise complement, and 0 is the $n$-vector $(0, \ldots, 0)$ , while 1 is the $n$-vector $(1, \ldots, 1)$. The case of the operator space is not too different. We will restrict ourselves, here, to binary operators, with the certainty that the generalization to an arbitrary arity is straightforward. We will define the additive inverse of an elemntary binary operator $(a, b) \mapsto p$ to be the map that maps all atom tuples not equal to $(a, b)$ to 1, and $(a, b)$ to the complement $-p$. That is:

$$-(a, b) \mapsto p := (x, y) \mapsto \begin{cases} -p \text{ if } (x, y) = (a, b) \\ 1 \text{ if } (x, y) \neq (a, b) \end{cases} \tag{1}$$

There will be a slight abuse of notation in that the variable $p$, when occuring in an elementary operator represents an integer from 1 to $n$, whereas when occuring in the image of an arbitrary Boolean operator, represents an n-vector vector $(0, \ldots, 1, \ldots, 0)$, with the 1 in the $p$-th position, so that, $-p = (1, \ldots, 0, \ldots 1)$, with 0 in the $p$-th position, corresponding to the usual additive inverse in a binary vector space.

We define addition to map identical domains to the sum (exclusive join), and different domains to the bit-vectors themselves :

$$(a, b) \mapsto p \ + \ (c, d) \mapsto q := \begin{cases} (a, b) \mapsto p, (c, d) \mapsto q, \\ \text{all else} \mapsto 0, \text{ if } (a, b) \neq (c, d) \\ (a, b) \mapsto p + q, \\ \text{all else} \mapsto 0, \text{ if } (a, b) = (c, d) \end{cases} \tag{2}$$

There is, once again, the same abuse of notation that we discussed above, with $p+q$.

We define multiplication, similarly, as mapping elementary operators with different domsins to 0, while same domains to the product (Boolean meet).

$$(a,b) \mapsto p \cdot \ (c,d) \mapsto q := \begin{cases} 0 \text{ if } (a,b) \neq (c,d) \\ (a,b) \mapsto p \cdot q, \\ \text{all else} \mapsto 0, \ \text{if } (a,b) = (c,d) \end{cases} \tag{3}$$

It's easy to see that any arbitrary binary Boolean operator $f : n^{\{\}2} \mapsto 2^n$ can be decomposed uniquely to a sum of elementary binary operators by first resolving the function to a sum of operators defined on a single ordered pair in $n^{\{\}2}$, and then resolve each of those elementary operators according to the atom decomposition of the function value.

Once, we do that we can define multiplication and addition by using the distributive law on the decomposition, so that our space automatically satisfies the 2-algebra axioms (the field action being the trivial action that sends 0 to the 0 map, and 1 to the identity map).

Of course, it might be convenient to not have to decompose each time we apply a 2-algebra operation, and perhaps I will compute the operations directly, according to the domain sets, zero values and whatnot.

Thus we have defined where $\mathbf{Op}_2^n := 2^{n^3}$ as a 2-algebra, and the direct sum across all arities gives us the infinite-dimensional space $\mathbf{Op}^n := \bigoplus_{a=0}^{\infty} \mathbf{Op}_a^n$.

In practice, though we will have a fixed finite set of operators, and we will restrict ourselves to the finite space generated by them, and when we look at the representation of the symmetric group on the atoms, we will examine the setwise stabilizers of the given set of operators, within this generated subspace.

# 4 A New Transducer Approach

Rather than using partial algebras to model the musical process, we will try transducers. The input words will correspond to the passive internal genetic music, and more generally to an abstract genetic state, while the output words will correspond to the "transcription", the typed music. The formal states of the transducer, will be defined as subsets of the setwise stabilizer of the operators $\{\mu_\Xi\}$ in $A := (2^m, \mu_\Xi)$, That is, $Q := \mathscr{P}(S^{\{\mu_\Xi\}})$. These will also be the initial and final states of our transducer.

The input alphabet will be arbitrary subsets of permutations of $m$ elements (not just from the setwise stabilizer), and will be denoted $P := \mathscr{P}(S_m)$. The output alphabet will be $T := \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, *, -\}$, the musical alphabet. Therefore we define the *musical output transducer* as $\mathcal{T} := (Q, E)$, where $E \subseteq Q \times P^* \times T^* \times Q$.

We will also, tentatively, impose the requirement that a transition's origin be included in each of the input alphabetic states.

Essential to our argument will be the countable atomless Boolean algebra, and we will use as isomorphism type the interval algebra on rational endpoints. I still have to work out what the free generators are explicitly. I'm also learning about ordered permutation groups, so as to more closely characterize the types of actions of $A$ and $B$ on our interval algebra that are "allowed." Since $A$ and $B$ are finite, I'm optimistic.

With that in mind, we will also define two maps that associate characters in $T$ to subsets of operators of $A$ and $B$. This will be essential in exploring the connections between $P$ and $T$, which don't have to be explicitly computed, but should obey certain rules.

So, we'll assume the existence of $\alpha : T \to \mathscr{P}(\mu_\Xi)$ and $\beta : T \to \mathscr{P}(\eta_Z)$.

Somewhere along the line we will posit that input and output words in $\mathcal{T}$ be of the same length.

We will also define at some point, a *musical input transducer*, which will be a "reversal" of the above transducer in that it will assume a priori a musical passage $p \in T^*$, some or all of whose factors form the input words, and whose output words are the "Boolean states."

To return to our musical output transducer, we will explore utilizing the fact that its specification will imply a direct pattern between states in $P$ and $Q$, quite independent of the output alphabet, and to achieve this, we will correleate with it context-free grammar, $\mathscr{G} := (P\backslash Q, Q, R)$, where $R \subseteq (P\backslash Q) \times P^*$. This way, we can formaalize the fact that the purification of spiritual state is "algebraic", and only dependent on the finite states available, (and perhaps on local properties of the Boolean operators).

What we can say, for sure is that when the transducer reaches input words of the form $Q^*$, then the computations involved in producing the notes should be directly consequent from some automorphisnm group of (a subgroup of) $S^{\{\mu_\Xi\}}$.

Much of this is speculation, and before I continue with the exact nature of the computations involved, I think it will help to first work on the implications of the context-free grammar $\mathscr{G}$, coupled with certain "intuitive" facts concerning our metaphysical motivation.

Also, I am quite absorbed in learning about ordered permutation groups, so that once again, it will be clear which bijections from the atoms of $A$ and $B$ to free generators in our rational interval algebra will be compatible with permutations of those atoms.

That's all for now! Exciting stuff!

# 5 Parameterizing partial operations

Let $\alpha : \mathbb{N}_0 \to \mathbb{N}_0$ and $\beta : \mathbb{N}_0 \to \mathbb{N}_0$ be the multiplicity signatures of BAs $2^m$ and $2^n$ respectively, and let $\Xi := \bigcup_{a \in \mathbb{N}_0, \alpha(a)>0} \{a\} \times \alpha(a)^{\{\}}$ and $\Xi' := \bigcup_{a \in \mathbb{N}_0, \beta(a)>0} \{a\} \times \beta(a)^{\{\}}$ parameterize their operators, so that $A := ((2^m), f_\Xi)$ and $B := ((2^n), g_{\Xi'})$, $m > n$. Let $S := \Psi(f_\Xi)$ and $S' := \Psi(g_{\Xi'})$ be matrix subsets of $\mathcal{TM}_n$ corresesdponding to the operators of $A$ and $B$.

Let $\underline{P} := (T^+, (\mu_\sigma)_{\sigma\in S_m}), (\eta_g)_{g\in g_{\Xi'}})$ be a partial algebra on the free semigroup on $T := \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, *, -\}$ corresponding to a 13-letter musical alphabet that is convenient for 10-key musical typing. Guitarists can note that these are fret numbers of the A-string, a good place to start, while pianists have to do more work! My preferred interperetation of the symbols is 0-9 = A-F#, / = G, * = G#, - = rest/sustain. When dealing with the free monoid $T^*$, we will denote the neutral element by $\varepsilon$.

Next, let $F$ be the the interval algebra on rational endpoints (which is isomorphic to the the unique countable atomless BA or alternatively, the free countable algebra). Let $F := \langle E\rangle$ where $E := \{[0, r)|r \in \mathbb{Q}\}$. Any automorphism of $F$ would have to presreve the linear order of every rational endpoint of $E$, and vice versa, so we see, $\mathbf{Aut}(F) \cong \mathbf{Aut}_<(\mathbb{Q})$.

We will be interested in the *countable free kernels* $K$ and $L$, defined so that $2^m \cong F/K$ and $2^n \cong F/L$.

Also, we will equip $F$ with the signature $(F, f_\Xi, g_{\Xi'})$, where the analogous operators are chosen so that the free projections mentioned above are indeed homomorphisms. This and the fact that $A$ and $B$ are finite, means that the automorphisms of the universes of $A, B$, and $F$ will coincide with their automorphisms as BAOs.

Let $\Delta : S_m \to GL(\mathcal{TM}_m)$ and $\Gamma : S_n \to GL(\mathcal{TM}_n)$ be the infinite dimensional representations of $S_m$ and $S_n$, respectively, that permute columns and block columns and block columns of block columns in the natural, uniform, defining way.

Although this is useful, for our purposes we will have to focus ourselves on restrictions, so let $S_\Delta := \mathbf{Stab}_\Delta S$ and $S_\Gamma := \mathbf{Stab}_\Gamma S'$ be setwise stabilizers and $\delta : S_\Delta \to GL(\langle S\rangle)$ and $\gamma : S_\Gamma \to GL(\langle S'\rangle)$ be natural restrictions of $\Delta$ and $\Gamma$.

One important thing to note is that the signature of $\underline{P}$ induces an action of a symmetric group on the partial algebra by permuting the partial operations according to the indices of $\mu_\sigma$ and $\eta_f$. We will use the free quotient of the intersection of the free kernels $C := (F/(K\cap L), f_\Xi, g_{\Xi'})$, where the operators are chosen so that a member of a coset gets mapped to the same value as the finitary representative. The advantage of this is that the symmetric group $S_r$, where $r$ is the width of $C$, paramaterizes both $(\mu_\sigma)_{\sigma\in S_m}$ (through the composition of the given permutation's automorphism of $C$ with the canonical projection of $C$ on $A$) – and also parameterizes $(\eta_g)_{g\in g_{\Xi'}}$ (through the *action* of the composition of the given permutation's automorphism of $C$ with the canonical projection of $C$ on $B$).

The next step of course will be to see how the defined representations get extended on $S_r$, how the setwise stabilizers of $f_\Xi \cup g_{\Xi'} \subset \mathbf{Op}_r$ under those extensions are related to their component counterparts $S_\Delta$ and $S_\Gamma$.

Also, we will use our partial algebra to compare the paramaters which yield intersecting graphs for partial operations $\mu_\sigma$ and $\eta_g$ (i.e. overlapping musical word values on those partial operations) and compare the permutations from $S_r$ that determines them, and use these permutations to construct an element (preferably a generator!) in the setwise stabilizer $S_\Gamma$.

More to come.

# 6 Representations

Now, to return to the basic model of a finite Boolean algebra $2^n$ with normal, completely distributive operators $f_\xi, \xi \in \Xi$, in light of the 2-algebra isomorphism $\psi_1 : \mathbf{Op}_{1,n} \to 2^{n\times n}$ above, we can examine the linear action of $S_n$ on the vector space $2^{n\times n}$, and we see that this gives rise to the *defining Boolean operator representation*, $\delta : S_n \to GL(2^{n\times n})$ (afforded by the $2S_n$-module $2^{n\times n}$). It's clear that $\delta$ merely permutes the columns of a binary matrix. And of course we have the *defining Boolean representation*, similarly defined, corresponding to the action of a permutation of atoms on its Boolean sum: $\rho : S_n \to GL(2^n)$ (afforded by the $2S_n$-module $2^n$). That is, it's a permutation representation on the binary vector space.

It turns out that the two homomorphisms are inherently connected by the formula $\delta(\sigma)(R)v = R\rho(\sigma)(v)$, for $R \in 2^{n\times n}$, $v \in 2^n$, and $\sigma \in S_n$. (We can write these more clearly as actions, $(\sigma\cdot R)v = R(\sigma\cdot v)$. Indeed this follows from the fact that multiplying a Boolean vector by a matrix whose columns have been permuted by $\sigma$, is the same as multiplying the non-permuted matrix by the vector whose elements have been permuted by $\sigma$. Note, the above equality is **not** the same as $\sigma\cdot(Rv) = \Sigma_{i\leq Rv}\sigma(i)$, the sum of the atoms obtained by permuting those below the Boolean vector $Rv$ with $\sigma$.

# 7 Stabilizers

Given our defining unary Boolean operator representation, $\delta : S_n \to GL(2^{n\times n})$, a natural question to ask in its own right (which becomes more manageable in light of the operator algebra isomorphism) is which permutations stabilize (permute) sets of operators. In the matrix algebra formulation, this becomes: given a set $S \subset 2^{n\times n}$ of matrices, compute the stabilizer $\mathbf{Stab}_\delta(S) := \{\sigma \in S_n | \sigma S \subseteq S\}$.

So we lose nothing in merely restricting ourselves to the matrix representations in our quest for stabilizers. As one might expect, sets as important as these stabilizers don't come easy, in light of the current non-existence of polynomial time algorithsm to compute setwise stabilizers.

I'm in the process of trying to see if there's an elegant calculation of the automorphism group of $(2^n, S)$, in terms of perhaps the stabilizer $A := \mathbf{Stab}_\delta(S)$. It seems that they should be connected, and so far I reasoned that $\mathbf{Aut}(2^n, S) = \{\sigma \in S_n | \exists \eta \in A, \forall M \in S, x \in 2^n, M(\sigma\cdot x) = \sigma((\eta M)\cdot x)\}$ I'll let you know how the calculation turns out! Before, I'll get to the "music theory," I'll leave you with a

**Conjecture 1.** *For all $n \in \mathbb{N}$, there exists an $S \subset 2^{n\times n}$, such that the defining unary Boolean operator representation representation $\delta : S_n \to GL(2^{n\times n})$, when restricted to $S' := \boldsymbol{Stab}_{2^{n\times n}}(S)$, the setwise stabilizer of $S$, is equivalent to the restriction of the defining unary Boolean permutation representation $\rho : S_n \to GL(2^n)$ to $S'$.*

The reason I believe this might be true has to do with the fact that digital Boolean algebras have their "brains" embedded in their Boolean existence, and therefore they both register and manifest automorphisms in the same digital space.

# 8 Music Theory

We are not interested in a completist account of music theory which explains all of its mysterious combinatorics or neurological phenomena, or tries to create music algorithmically, because that is foreign to our purposes. We need the bare minimum of music theory required to aid in computation.

## 8.1 New Partial Approach

It's clear that since we're interested in the connections between a group (the setwise stabilizers in the defining representations) and certain strings of notes, that we should be interested in the free semigroup on 13 generators (12 for each note and one signifying the rests/sustains that provide the rhythm). Octave ambiguities wouldn't matter in the sense that all musical passges would still be musical with notes of the same value but different octave. It might strike one as less catchy than another, but it would undeniably be music, and the mere existence of a certain aesthetic best among equivalents is enough to earmark the equivalence class, even if not define a canonical representative.

So, the idea is that we will employ a partial algebra whose universe is the free semigroup on 13 generators, which we'll denote by $F_{13}$, and whose function symbols are any labels of the "build-up / resolution" pattern that is central to musical sensation.

The "built-up passages" (which of course can be nested), correspond to word arguments in a partial function, whose resolution is denoted by the function value word. By earmarking the parts through an audio device and a human listener, we can assemble a set of partial functions corresponding to the sensory flavor of the passage. Of course, a passage grows, and peformers usually like to add to songs, as do composers and improvisers, so this just corresponds to new partial functions on the words (subwords of the musical passage).

The advantages of partial algebras is that they can construct "relatively free algebras over partial relative substructures" and we can already see some proof of this in connection with partial subalgebras. These come in three flavors – closed, relative, and weak relative. The closed subalgebra would correspond to some notion of "algorithmic melody construction" which as some algorithmic composers have found, can be used to undeniably create music. The relative subalgebra would correspond to restricting against certain examples of music that always sound bad to you. And the weak relative would correspond to filtering by removing notes and perhpas preserving rhythm.

This is more intuitive, and there are subtle points that have to be worked out, to both solidify the ideas, and so that we dont' lose sight of the forest for the trees.

More to come.

## 8.2 Old Partial Approach

As an alternative to the above beginnings, let's try using the technique of partial algebras in conjunction with Boolean algebras with operators, to utilize an interaction between the possible members of $13^L$, where $L$ is the length of our musical passage. We denote the set of all elements $\{1, \ldots, L\}$ by $L_1$. To reflect the structure in an improvised (or composed) passage, it's nice to use partial functions to express the possibilities and constraints that go through a musicians head as he improvises. For instance, a musician might know the last two notes he wants to play, and the first two notes, he wants to play, but not the middle two notes. This can be represented informally as, $t_1 \mapsto t_2 \to 13 \to 13 \mapsto t_3 \mapsto t_4$. In general, for any subset $S \subset L_1$, we can imagine an $L-$ bit-vector of the mask of the set, as it shifts throughout all the $L - |b_S|$ possibilities, where $b_S$ is the number of spaces between the left-most 1 and the right-most 1. So, each partial function in the algebra that we will soon formally define represents, a musical choice, within the context of preceding notes, with or without gaps. So, the first 4 notes in the musical passage $p$, and if we know thisnote is $t_5$, we notate this by $t_1 \mapsto t_2 \mapsto t_3 \mapsto t_4 \mapsto t_5$. So, when we wish to examine all choices corresponding to this pattern of 4 consecutive notes, then we simply use an operation of arity 4, which is in $13^4 \to 13$. But, we are implicitly assuming that this corresponds to 4 consecutive notes. As we noted above, there can be very real musical choices made with gaps in the middle, so to remedy that we can using the power set algebra $2^L$ to index partial dictive functions So, the value $\{t_i\}_4$,

With this basic idea, we can examine the partial algebra $\mathcal{D}(\surd) := \{f_S | S \subset L_1\} where Listhelengthofp$

A brief outline :

The letter $\mathcal{D}$ was chosen to denote diction.

More to come!

## 8.3 Old Flawed Approach

So, we start by looking at the Cartesian product of $13^M$, where $M$ is the (presumably large) number of notes in the musical passage, and 0-11 signify A-F#, while 12 signifies a rest or sustain. The idea is to get the bare minimal notion of music that could drive our computations. We don't distinguish between notes an octave apart, and we use the blank note 12 to define rhythm, which of course is quite complicated for complicated rhythms, but assuming nothing too avant-garde, we can capture most hummable melodies (in an equivalence class whose octave ambiguities are probably still musical).

We define the projection function $i_p : \{M\} \to 13$ which maps the index of the passage $p$ to its note value. For instance, if the 3rd note in passage $p$ is a

C#, then $i_p(3) = 4$.

To create the BA with ops, we will use the standard procedure of starting with a relational structure, and then defining its uniquely determined complex algebra.

We'll let $R_p$ denote a collection of relations of unrestricted arities determined by the passage $p$ (and perhaps determined by music-theoretic principles). The primary relations that are uniquely determined by $p$ are the unary relations $\rho_t \subset 13^M$, where $\rho_t a \leftrightarrow i(a) = t$. One useful class of operators is the $k$-ary relations $\sigma_{k,t} \subset 13^K$ where $(t_1, \ldots, t_k) \in \sigma_{k,t}$ iff there is a successive sequence of $k$ components in $p$ which contain $t$. We can also, instead of specifying only one tone to search for, search for a potential sub-passage. In any event, we will restrict our attention to the complex algebra $B_p$ of $R_p$, which is the power set algebra on $\{M\}$, with an operator $f_\rho : \{M\}^k \to M$, for some relation $\rho \subseteq \{M\}^{k+1}$, sticking to the convention that

$$(t_1, \cdots, t_{k+1}) \in \rho \leftrightarrow f_\rho(t_1, \cdots, t_k) = t_{k+1}$$

.

So, the hope is that since Boolean homomorphisms are tantamount to ideals, which in finite BAs are simple enough, the main work will be generating enough relevant operators in $B_p$ that are compatible with the defining Boolean operator matrix representation of $S_N$ (which it'll be helpful to calculate explicitly soon!!) and a given set of operators $S$ whose setwise stabilizer you want to find in the given BA $B := 2^N$, which imply the existence hopefully of a homomorphism of BAs with ops between $B_p$ and $B$, whose stabilizers under the actions of the symmetric groups are comparable.

The hope is that it'll be much easier to compute setwise stabilizers in $B_p$ than in $B$, since we get to choose our operators that are generated by the primary ones, and by whatever aesthetic constraints or persistent musical patterns we can describe. The variety is much more manageable in a music theoretic BA than in a BA with arbitrary operators.

# 9 More elaboration of philosophical conjecture

FIX THIS, BUT JUST IGNORE THE TENSOR LEFTOVERS. To elaborate more on the topic of setwise stabilizers, here's another conjecture :

**Conjecture 1.** *The setwise stabilizer* $\{G_\Xi := \boldsymbol{Stab}(f(\Xi))$ *of the Boolean Operator Representation acting on the operators' generated subspace of the tensor matrix algebra is always a symmetric group.*

Pretty bold claim, and one that I can only describe in terms of a hypothetical digital system primarily determined by the variables and opeators of one specific very large finite Boolean algebra, corresponding to the gates. Now, it's not my claim that the Boolean properties of the gates determine most of the functionality, but the types of universal disfluencies that might threaten fault tolerance are all dependent and dare I say manipulable by the symmetries in the gates. So, the reason I believe this superstitious and bold statement is true is the following : The only automorphisms possible in exact digital circuitry abiding by the laws of physics are those which map an operator to another, thereby using existing functionality in new symmetric ways. And those are the flukes and some men's only hope. Enough of those inevitably signal the Physics Police, because then the broken automorphisms ensue (they're **not** in the setwise stabilizer!). My point is that the full automorphism group of a subalgebra (with restricted operators) of the entire Boolean algebra would have magical and neutral behavior. Which is precisely the setwise stabilizer of given operators of our given representation, provided such things exist, and perhaps a crystallographer can make a better and more sane case than that, but now it's time to prove this supposedly obvious statement.

There's also

**Conjecture 1.** *The quotient of the setwise stabilizer* $G_\Xi := \boldsymbol{Stab}(f(\Xi))$ *by the pointwise stabilizer* $H_\Xi := \mathbf{Stab_1}(f(\Xi))$ *of the Boolean Operator Representation acting on the operators' generated subspace of the direct sums of powers of binary matrix algebras is always a symmetric group.*

Similar reasoning – factoring out the netural elements doesn't alter behavior. More to come.

# 10 Loose Outline of Proof of Conjecture

Utilizing the setting and structures already established, we have $A, B$, and the BORs $\Gamma : S_A :\to \langle S\rangle$, $\Delta : S_B \to \langle S'\rangle$. $S_A := \mathbf{Stab}_A(S)$ $S_B := \mathbf{Stab}_B(S')$. And we are interested in the quotients of the repsective setwise stabilizers and pointwise stabilizers $Q_A$ and $Q_B$. We will try to use the parameterization of $S_A$ and $S_B$ into $(\mu_\sigma)$ and $(\eta_\sigma)$, which is a bit of a change from above. I seem to believe that utilizing the setwise stabilizer to paramaterize, rather than some composition of operator and permutation/automorphism, is better and more natural. So, we use the permutations in the setweise setabilizers to parmaaterize, and we examine the quotients, to see if we can prove that an arbitrary coset corresponds to a transposition in the left regular action. And if we can embed an arbitrary transposition in the stabilizer quotient, then we're fine. So, to being the proof, we will assume that we have two Boolean algebras A, and B, mentioned above, and described above in prewvious sections. And $m < n$, where $m$ and $n$ are the repsective widths, and by assuming that in both $A$ and $B$ there is an element in the symmetric group, whose coset of the pointwise stabilizer corresponds to a transposition in the left regular action not contained in $Q_A$ or $Q_B$, we can derive the contradiction that perhaps $m = n$, or some such, by presupposing and "exhausting" the finitary paramterization mentioned above, and using free monoids on the musical alphabet to derive a correpsondence between overlapping partial operations that classify growing musical scores, and the permutations that determine these operatorions, in the not-yet described paramaterization. So, it's just a question of exhausting patterns, because everything is finitary. The infinite words, and whatnot, will only be used up to a point, there will be some convergence of words, and blam, a contradiction. Now for learning more free semigroup theory and combinatorics! And trying to translate overlapping words in a free monoid with automorphisms of the direct product $A \times B$, and decompos directly perhaps, but more learning!

## 11 Automorphisms of First-Order Structure : Know Best Focus

So, we look at an ordered permutation group $G \subseteq \mathrm{Aut}(\mathbb{Q}, <)$, which we associate with the countable atomless free algebra and a chain of free generators $\{I_q\}_{q \in \mathbb{Q}}$ order-isomorphic to $\mathbb{Q}$. By a well-known theorem [??], It turns out that $G$ is the automorphism group of a model characterizesd by a theory if and only if it's oligomorphic. So, the idea is that the model of this theory would have to have certain properties, namely some type of "maximum of 'available' automorphisms," by which I mean with respect to certain setwise stabilizers. Another property it would be nice to have is some type of maximal universality, if not pure freedom. So, whether or no I'm right that $G$ and not just $\mathrm{Aut}(\mathbb{Q}, <)$ is guaranteed to be oligomorphic, we can at least explore that condition. So, first suppose $G$ is oligomorphic, and thereefore that it is the automorphism group of a countable model determined by an $\mathscr{L}$-sturcture $\mathfrak{M}$ that is the unique model of a theory $T$ in a first-order language $\mathscr{L}$. We wish to explore the relationship between $\mathfrak{M}$ and the subojects, in light of Fraisse's theorem. (Which I have to pore through.) We know the number of orbits of $\mathrm{Aut}(\mathbb{Q}, <)$ on a tuple of distinct elements of $\mathbb{Q}$ is $n!$. This I got from Cameron's "Oligomorphic Permutation Groups" One thing that might be critical is as is discussed, characterizing the sequences of orbit frequencies over lengths of tuples, which I will now read more about. Good to be naturally curious over something that's not conventionally behavioral!

## 12 Tristone Systems

One can create a ternary alphabet $C = \{+, ?, -\}$, and informally view the symbols as, repectively, $\{$*yes*, *freely generate*, *no*$\}$, corresponding to a sort of philosophical Stone's Representation Theorem Trichotomy system of decisions, whereby a binary decision can be reached either by conventional binary means, corresponding to + or -, or it can be achieved by manipulating the exhaustive and fundamental two-valued decisions (2-valued homomorphisms) guaranteed by Stone's Representation Theorem, corresponding to ?. For this last case, there's a not fully determined interface with free algebras, all of which might generalize to infinite BAs too one day, we'll see.

We'll start things off by defining actions. First, there's the induced action of $S_n$ on the BA $2^n$, which acts merely as automorphisms, or in other words, permutes the atoms below the given element.

Then there are the following *tristone actions*

1. (a) The left action $S_n$ on $S_{2^n}$, by $(\sigma\cdot f)x := \sigma(fx)$, where $\sigma \in S_n$, $f \in S_{2^n}$, and $x \in 2^n$, and the action is the induced action above.

   (b) The right action of $S_n$ on $S_{2^n}$, by $(f\cdot\sigma)x := f(\sigma x)$, again under the induced action.

2. The left action of $S_n$ on $\mathbf{Op_n}$, by permuting the elementary restrictions, as discussed above, i.e. $\sigma(i \mapsto x) := (\sigma i) \mapsto x$, for each atom-value pair.

3. The left action of $S_{2^n}$ on $\mathbf{Op}_n$, where this time the permutation acts on the value. $f(i \mapsto x) = i \mapsto (fx)$, for each atom-value pair.

Having defined these actions, a *tristone* is merely a triple of atom permutations, Boolean permutations, and Boolean operators, whose tristone actions between them are compatible as setwise stabilizers. We refer to $Q$, $F$, and $O$ as *stones.*

**Definition 1.** *An n-tristone is a triple $(Q, F, O)$, with $Q \subseteq S_n$, $F \subseteq S_{2^n}$, and $O \subseteq \mathbf{Op}_n$ such that*

*3#1* $Q \subseteq Stab_{S_n} F$, *for each (left, right) action.*

*3#2* $Q \subseteq Stab_{S_n} O$.

*3#3* $F \subseteq Stab_{S_{2^n}} O$.

Eventually we hopefully will realize systems of tristones that are loosely stable (closed) with respect to changes in $Q, F$, or $O$. It's unclear which changes are admissible, and which further constraints they uniquely determine. To be determined. We can have atoms permutations act on $Q$ (or get added to $Q$), Boolean permutations act on or get added to $F$, and operators get added to $O$.

The rest of what follows is tentative.

**Definition 2.** *An n-tristone system (or n-3# system) is a collection of n-tristones each member $(Q, F, O)$ of which is contained in the set of tristones determined by any of the following changes in a single stone.*

*Q* *(a)* $RQ := \{\gamma\sigma | \gamma \in R, \sigma \in Q\}$

*(b)* $QR := \{\sigma\gamma | \gamma \in R, \sigma \in Q\}$.

*(c)* $R \cup Q$.

*F* *(a)* $GF := \{\gamma\sigma | \gamma \in G, \sigma \in F7\}$.

*(b)* $FG := \{\sigma\gamma | \gamma \in G, \sigma \in F\}$.

*(c)* $F \cup G$

*O* *(a)* $P \cup O$

The connection with this and our "ternary" alphabet $C$ is that, for a tristone system $\mathcal{T}$, we label, $(Q', F', O') = (Q, F, O) \oplus (R, G, P)$ if $Q' = Q \cup R$. (Perhaps, $G$ or $P$ can be left out if the result is already determined.) Likewise, $(Q', F', O') = (Q, F, O) \oslash (R, G, P)$ if $F' = F \cup G$, and $(Q', F', O') = (Q, F, O) \ominus (R, G, P)$ if $O' = O \cup P$.

So, $C := \{+, ?, -\}$ corresponds to respectively to ©$:= \{\oplus, \oslash, \ominus\}$, with the necessary paramaters as an extra specification, and given a tristone $\mathcal{T}$, we can determine how a word over $C$ can "instruct" some sequence of commands on $\mathcal{T}$.

The question of generation is one that will be postponed, since it might be as pluralistic as in partial algebras.

The next natural question is how many or which tristones can exist for a given stone. If $(Q, F, O)$ is a tristone, we write $3\#(Q, F, O)$. And to denote the set of all tristones with one or more component given, we write, $3\#(S) := \{(Q, F, O) | S \in \{Q, F, O\}$ and $3\#(Q, F, O)$ and $3\#(S, T) := \{(Q, F, O) | \{S, T\} \subset \{Q, F, O\}$ and $3\#(Q, F, O)$.

So, it would be nice if $3\#(S, T)$ were small, maybe even a singleton, for specific values of $S$ and $T$, and soon we will go through consequences of non-singleton values of $3\#(S, T)$. We will also do basic inclusion lemmas for this, to express intuitive patterns like that, an increase in the number of operators in $O$, decreases the size of $Q$ for a fixed $F$.

And in general other characterizations of $3\#(Q, F, O)$ will be sought.

One big goal in all of this is to see if we can determine "new" permutations in $Q$ or elements in the other stones, which could have applications to computational group theory in situations where you are given only a specification of a group, and need to determine concrete generators. So, we will find ways to describe $3\#(F, O)$ given $3\#(Q, F, O')$, and $O \subset O'$. So, let's start with some easy lemmas.

**Lemma 1.** *$3\#(Q', F, O')$ and $O \subsetneq O'$ implies, there exists a $Q$ such that $3\#(Q, F, O)$ and $Q \supseteq Q'$.*

*Proof.* Choose $Q := \mathrm{Stab}_{S_n} F \cap \mathrm{Stab}_{S_n} O$. By hypothesis, $Q' \subseteq \mathrm{Stab}_{S_n} F \cap \mathrm{Stab}_{S_n} O' \subseteq \mathrm{Stab}_{S_n} F \cap \mathrm{Stab}_{S_n} O = Q$, and by construction, (3#1) and (3#2) are satisfied. For (3#3), we see $F \subseteq \mathrm{Stab}_{S_{2n}} O' \subseteq \mathrm{Stab}_{S_{2n}} O$. □

But the real excitement about something like this, would be if the containment were proper, and I'm investigating deciding if/when, proper containment is preserved by full setwise stabilizers. It might be. In which case I'll strengthen the above.

In relation to music, can we associate with any musical passage, or, any formal word, a word in $\{+, ?, -\}*$ that can compute parameters in the given passage? Perhaps too vaguely stated.

Building some type of holistic non-determinism in the very core of a ternary computational alphabet, should be interesting, if only to just bounce back and forth between all the dependent notions!

Of course we will work with automata theory, in conjunction with music, and arbitrary semigroups, and hopefully even formal languages that can be turned into a compiler! Lots of semi-routine fodder to evaluate, like the compatibility constraints in ternary words under given tristone systems.

Also interfaces with 2-modular representation theory in relation to the generated Boolean operator spaces of a tristone.