September 25, 2015 UMDEPP 08-010, SUNY-O/667

arXiv:0806.0050v1 [hep-th] 31 May 2008

Topology Types of Adinkras and the Corresponding Representations of N-Extended Supersymmetry

C.F. Doran[a], M.G. Faux[b], S.J. Gates, Jr.[c], T. Hübsch[d], K.M. Iga[e], G.D. Landweber[f], and R.L. Miller[g]

[a] *Department of Mathematics, University of Washington, Seattle, WA 98105,* doran@math.washington.edu
[b] *Department of Physics, State University of New York, Oneonta, NY 13825,* fauxmg@oneonta.edu
[c] *Center for String and Particle Theory, Department of Physics, University of Maryland, College Park, MD 20472,* gatess@wam.umd.edu
[d] *Department of Physics & Astronomy, Howard University, Washington, DC 20059,* thubsch@howard.edu
DAMTP, Delaware State University, Dover, DE 19901, thubsch@desu.edu
[e] *Natural Science Division, Pepperdine University, Malibu, CA 90263,* Kevin.Iga@pepperdine.edu
[f] *Mathematics Program, Bard College, Annandale-on-Hudson, NY 12504-5000,* gregland@bard.edu
[g] *Department of Mathematics, University of Washington, Seattle, WA 98105,* rlmill@math.washington.edu

ABSTRACT

We present further progress toward a complete classification scheme for describing supermultiplets of N-extended worldline supersymmetry, which relies on graph-theoretic topological invariants. In particular, we demonstrate a relationship between Adinkra diagrams and quotients of N-dimensional cubes, where the quotient groups are subgroups of $(\mathbb{Z}_2)^N$. We explain how these quotient groups correspond precisely to doubly even binary linear error-correcting codes, so that the classification of such codes provides a means for describing equivalence classes of Adinkras and therefore supermultiplets. Using results from coding theory we exhibit the enumeration of these equivalence classes for all cases up to 26 supercharges, as well as the maximal codes, corresponding to minimal supermultiplets, for up to 32 supercharges.

1 Introduction, Review and Synopsis

In the past decade there has been a renewed interest among physicists in certain basic questions in supersymmetry and, in particular, the classification of supersymmetric theories. Dualities and advances in string theory and its M- and F-Theory extensions have raised an awareness of a range of supersymmetric theories that were not previously well-studied. In many cases, dimensional reduction has been a useful—and sometimes the only—way to analyze these theories. In particular, dimensional reduction to one dimension (time) is often particularly enlightening. This reduction may, in fact, retain all the information necessary to reconstruct its higher-dimensional preimage, by the reverse process of dimensional "oxidization" [1]. Independently, N-extended worldline supersymmetry also governs the dynamics of wave functionals in supersymmetric quantum field theories and so applies to all of them also in this other, more fundamental way.

In addition, the past decade has seen a renewed interest by mathematicians in physical supersymmetry, in part due to the corresponding interest by physicists, and in part because of new induced advances in algebraic geometry [2,3,4,5] and four-dimensional topology [6,7,8] (see Ref. [9] for a survey accessible to physicists). This has brought about a desire to approach the foundations of the subject more systematically.

1.1 Adinkras

Studies of one-dimensional supersymmetric theories have led to the development of the $\mathcal{GR}(d,N)$ algebras [10,11,12], to the *Adinkras* [13], and to other efforts [14,15,16,17,18,19,20]. The main goal of these works is a comprehensive, constructive and *conveniently usable* classification of representations of supersymmetry. Herein, we focus on supersymmetry with no central charge, and examine those representations which may be described using Adinkras.

Adinkras are directed graphs with various colorings and other markings on vertices and edges, which in a pictorial way encode all details of the supersymmetry transformations on the component fields within a supermultiplet [13]. They can be related to superspace constructions [21,22], aid the understanding of complex systems of transformation rules [23,24], and provide a more systematic and complete classification tool for representations of supersymmetry.

The study of Adinkras, then, naturally leads to the question of their classification.[1] This classification of Adinkras naturally falls into four steps:

1. Determine which topologies are possible (the topology of an Adinkra is the underlying graph of vertices and edges without colorings, as, for instance, in Ref. [21]).

2. Determine the ways in which vertices and edges may be colored. The topology of the Adinkra, together with the colorings of vertices and edges, will be called the *chromotopology* of the Adinkra. It is chromotopologies that are classified in this paper.

[1] It must be cautioned that there are two ways in which a classification of Adinkras falls short of classifying $D = 1$ off-shell supermultiplets: first, not every such supermultiplet can be described by an Adinkra, and second, sometimes such a supermultiplet can be described by more than one Adinkra. Nevertheless, the classification of Adinkras is an important step in such a program.

3. Determine the ways in which edges may be chosen as dashed or solid. This is closely related to the well-known Clifford algebra theory, and will be studied in a future effort.

4. Determine the ways in which arrows may be directed along each edge. This issue is addressed in Ref. [21], and shown to be equivalent to the question of "hanging" the graph on a few sinks. Alternately, we can start with an Adinkra where all arrows go from bosons to fermions, then perform a sequence of vertex raises to arrive at other choices of arrow directions.

As it happens, it is convenient to do 1. and 2. together; that is, to classify chromotopologies. Herein, we show that the classification of Adinkra chromotopologies is equivalent to another interesting question from coding theory: the classification of doubly even binary linear codes. Much work has already been done in this area [25,26,27], and the work described in this paper goes even further in developing this classification; see Appendix C.

We emphasize that we focus here on the supersymmetric representation theory, not the dynamics of supersymmetric theories. This is logically necessary in any classification endeavor, as we need to first know the full palette of supersymmetric representations before discussing the properties of the dynamics in theories built upon such representations. Clearly, presupposing a standard, uncoupled Lagrangian for the supermultiplets that we intend to classify would necessarily limit the possibilities; there do exist supermultiplets which can only have interactive Lagrangians [28,29]. Herein, we defer the task of finding Lagrangians involving the supermultiplets considered in this paper. In Refs. [22, 30], we have in fact started on such studies, and, using Adinkras, have constructed supersymmetric Lagrangians for some of the supermultiplets that are also discussed herein.

In units where $\hbar = 1 = c$, all physical quantities may have at most units of mass, the exponent of which is called the *engineering dimension* and is an essential element of physics analysis in general. The engineering dimension of a field $\phi(\tau)$ will be written $[\phi]$; for more details, see Refs. [21,31].

1.2 Main Result

Our main result about the chromotopology types of Adinkras and the corresponding supermultiplets, up to direct sums, may be summarized as follows:

We define the function:

$$\varkappa(N) := \begin{cases} 0 & \text{for } N < 4, \\ 1 & \text{for } N = 4, 5, \\ 2 & \text{for } N = 6, \\ 3 & \text{for } N = 7, \\ 4 + \varkappa(N-8) & \text{for } N \geq 8, \text{ recursively.} \end{cases} \tag{1.1}$$

1. Every Adinkra can be separated into its connected components. (The supermultiplet corresponding to such an Adinkra breaks up into a direct sum of other supermultiplets, each of which corresponds to one of the connected components of the Adinkra).
2. Each such component has the topology of a k-fold ($0 \leq k \leq \varkappa(N)$) iterated $\mathbb{Z}_2$-quotient of the N-cube. (The corresponding supermultiplet is likewise a quotient of the supermultiplet corresponding to the N-cube).

(a) The result of iterating these $\mathbb{Z}_2$-quotients is equivalent to quotienting by a subgroup of $(\mathbb{Z}_2)^N$, that is, by a binary linear $[N, k]$ code.
(b) A code can be used for such a quotient if and only if it is a doubly even code.
(c) There is a one-to-one correspondence between chromotopologies of Adinkras and doubly even codes, under which column-permutation equivalence in codes corresponds to swapping colors on the edges of the Adinkra, and R-symmetries of the supermultiplet.
(d) The number of distinct doubly even codes grows combinatorially with N and k, and distinct codes produce distinct Adinkra chromotopologies.
(e) A large subset of distinct doubly even codes are also permutation-inequivalent; two permutation-inequivalent codes produce distinct Adinkra topologies. This occurs for $N \geq 8$, but for $N \geq 10$ this occurs even when $k = \varkappa(N)$, giving rise to topologically distinct *maximal quotients*, *i.e.*, *minimal supermultiplets*. The number of these also grows combinatorially with N, k; see Table 4.

It is possible to peruse these results backwards: One may start with the minimal supermultiplets in the entry (2e) above, and then reconstruct the non-minimal ones by reversing the quotienting procedure described herein. We will not explore the combinatorially complex mechanics of such procedures here, but it reflects the very large numbers of $[N, k]$-codes in Table 4.

— ⋆ —

The paper is organized as follows: Section 2 is a brief introduction to codes, and Section 3 provides a review of Adinkras and their relationship with supermultiplets. The main material is in Section 4, where Adinkra topologies are classified in terms of doubly even codes, which we explore further in Section 5. We then turn to discuss the subtleties in identifying Adinkras with supermultiplets, which lead to the second part of our main result presented in full detail in Section 6.

A number of technical details are deferred to the Appendices. Details of the supersymmetry action in real supermultiplets are discussed in Appendix A, while those of the complex case occupy Appendix B. Finally, Appendix C presents the necessary details of the classification of doubly even binary linear error-correcting codes.

2 Codes

We begin with a brief introduction to the theory of codes so as to make it easier to spot the emergence of its elements in the subsequent sections. The Reader who wants a more thorough introduction to coding theory can consult a standard reference such as Refs. [25,26,27].

Generally, a code is a set of strings of characters, each of which is given some kind of meaning in order to communicate between two people (say). We do not assume these are secret codes, in that we are not necessarily trying to hide this communication from a third party. Given a fixed alphabet, there are many possible strings of letters from this alphabet, but a code will specify only some of these as codewords. In this paper we do not assign meaning to the codewords, and thus we consider a code simply to be a set of codewords.

In this paper, we will focus on *binary block codes*. In this case, the available characters from which the codewords are composed are the binary digits 0 and 1, and we fix a positive integer N, and require that each codeword have length N. Thus, a binary block code of length N is a subset of

$\{0,1\}^N$. Though the standard notation for an element of a cartesian product is $(x_1, x_2, \cdots, x_N)$, in practice we frequently abandon the parentheses and the commas, so that the element $(0,1,1,0,1)$ may be written more succinctly as the codeword 01101. The components of such an N-tuple are called bits, and the N-tuple is called a binary number.

If we think of $\{0,1\}$ as the group $\mathbb{Z}_2$, and $\{0,1\}^N$ as the N-fold product of this group with itself, then a binary block code is called *linear* when it is a subgroup of $\{0,1\}^N$. This is equivalent to the statement that if v and w are codewords, then $v \boxplus w$ (their bitwise sum modulo 2, or equivalently their exclusive or) is also a codeword. For instance, if $N = 3$, then $\{010, 011\}$ is a binary block code of length 3, but it is not linear because the linear combination, $010 \boxplus 011 = 001$, is itself not one of the listed codewords. On the other hand, $\{000, 001, 010, 011\}$ is a linear code.

Now, $\mathbb{Z}_2$ is not only a group; it is also a field, so $\{0,1\}^N$ can be viewed as a vector space over $\{0,1\}$. All the concepts of linear algebra then apply, but with $\mathbb{R}$ replaced with $\mathbb{Z}_2$. Elements of $\{0,1\}^N$ may be thought of as vectors, with vector addition the operation $\boxplus$ of bitwise addition modulo 2, scalar-multiplication by 0 setting every bit to zero, and scalar-multiplication by 1 leaving the vector unchanged. As in standard linear algebra, there is the concept of a basis. Given a linear subspace (that is, a linear code), we can find a finite set of codewords $g_1, \ldots, g_k$ so that every codeword can be written uniquely as a sum

$$\sum_{i=1}^{k} x_i g_i, \tag{2.1}$$

where the coefficients $x_1, \ldots, x_k$ are each either 0 or 1. The set $g_1, \ldots, g_k$ is then a basis, but in coding theory, it is also called a generating set for the linear code. As in real linear algebra, there is no expectation that a generating set is uniquely determined by the linear code, but the number k, called the *dimension* of the code, is always the same for a given linear code. It is common to say we have an $[N, k]$ linear code when N is the length of the codewords and k is the dimension. It is traditional to denote a generating set as an $N \times k$ matrix, where each row is an element of the generating set.

If $v \in \{0,1\}^N$, we define the *weight* of v, written $\mathrm{wt}(v)$, to be the number of 1s in v. For instance, the weight of 01101 is $\mathrm{wt}(01101) = 3$.

A binary linear block code is called *even* if every codeword in the code has even weight. It is called *doubly even* if every codeword in the code has weight is divisible by 4. Examples of doubly even codes are given in Section 5.1 below.

If v and w are in $\{0,1\}^N$, then $v \,\&\, w$ is defined to be the "bitwise and" of v and w: the ith bit of $v \,\&\, w$ is 1 if and only if the ith bit of v and the ith bit of w are both 1. A basic fact in $\{0,1\}^N$ is

$$\mathrm{wt}(v \boxplus w) = \mathrm{wt}(v) + \mathrm{wt}(w) - 2\ \mathrm{wt}(v \,\&\, w). \tag{2.2}$$

There is a standard inner product. If we write v and w in $\{0,1\}^N$ as $(v_1, \ldots, v_N)$ and $(w_1, \ldots, w_N)$, then

$$\langle v, w \rangle \equiv \sum_{i=1}^{N} v_i\, w_i \pmod 2. \tag{2.3}$$

We call v and w orthogonal if $\langle v, w\rangle = 0$. This occurs whenever there are an even number of bit positions where both v and w are 1. Note that $\langle v, v\rangle \equiv \mathrm{wt}(v) \pmod 2$, and thus, when $\mathrm{wt}(v)$ is even, v is orthogonal to itself. Also note that $\langle v, w\rangle \equiv \mathrm{wt}(v \,\&\, w) \pmod 2$. One important consequence for us is that if $\mathrm{wt}(v)$ and $\mathrm{wt}(w)$ are multiples of 4, then (2.2) implies that $\mathrm{wt}(v \boxplus w)$ is a multiple of 4 if and only if v and w are orthogonal.

3 Supersymmetric Representations and Adinkras

The N-extended supersymmetry algebra without central charges in one dimension is generated by the time-derivative, ∂_τ, and the N supersymmetry generators, $Q_1, \ldots, Q_N$, satisfying the following supersymmetry relations:

$$\left\{ Q_I \,,\, Q_J \right\} = 2\, i\, \delta_{IJ}\, \partial_\tau, \quad \left[\partial_\tau \,,\, Q_I \right] = 0, \quad I, J = 1, \ldots, N. \tag{3.1}$$

In this section we determine some essential facts about the transformation rules of these operators on fields for which it is possible to maintain the physically motivated concept of engineering dimension. We note that since the time-derivative has engineering dimension $[\partial_\tau] = 1$, the supersymmetry relations (3.1) imply that the engineering dimension of the supersymmetry generators is $[Q_I] = \frac{1}{2}$.

3.1 Supermultiplets as Representations of Supersymmetry

A real supermultiplet $\mathscr{M}$ is a real, unitary, finite-dimensional, linear representation of the algebra (3.1), in the following sense: It is spanned by a basis of real bosonic and fermionic *component fields*, $\phi_1(\tau), \ldots, \phi_m(\tau)$ and $\psi_1(\tau), \ldots, \psi_m(\tau)$, respectively; each component field is a function of time, τ. The supersymmetry transformations, generated by the Hermitian operators $Q_1, \cdots, Q_N$, act linearly on $\mathscr{M}$ while satisfying Eqs. (3.1), *i.e.*, Eqs. (3.3), below. The supermultiplet is *off-shell* if no differential equation is imposed on it[2]. The number of bosons as fermions is then the same, guaranteed by supersymmetry.

The full supersymmetry transformation, which preserves the reality of the component fields $\phi_A(\tau)$ and $\psi_A(\tau)$ and is generated by the Q_I, takes the form:

$$\begin{bmatrix} \phi_A(\tau) \\ \psi_A(\tau) \end{bmatrix} \;\mapsto\; e^{\delta_Q(\epsilon)} \begin{bmatrix} \phi_A(\tau) \\ \psi_A(\tau) \end{bmatrix}, \quad \delta_Q(\epsilon) := -i\epsilon^I\, Q_I, \quad \phi_A(\tau), \psi_A(\tau) \in \mathscr{M}, \tag{3.2}$$

where ϵ^I are the (anticommuting) parameters of the transformation. The infinitesimal transformation operators, $\delta_Q(\epsilon)$, in turn satisfy the relation:

$$\left[\delta_Q(\epsilon) \,,\, \delta_Q(\eta) \right] = 2\, i\, (\epsilon^I \delta_{IJ}\, \eta^J)\, \partial_\tau. \tag{3.3}$$

Both the relations (3.1) and (3.3) equivalently specify the supersymmetry algebra, and the infinitesimal transformation operator $\delta_Q(\epsilon)$ may be—and, in the physics literature, is—used to study the supersymmetry transformations in supermultiplets.

[2] Logically, it is possible for some—but not all—component fields to become subject to a differential equation. This does not violate the literal definition of the off-shell supermultiplet. However, it does obstruct standard methods of quantization, which is our eventual purpose for keeping supermultiplets off-shell. For an example in 4-dimensional supersymmetry, see Ref. [24].

3.2 Building Supermultiplets from Adinkras

Refs. [13,21,22] introduced and then studied Adinkras, diagrams that encode the transformation rules (3.2) of the component fields under the action of the supersymmetry generators $Q_1, \ldots, Q_N$.

Remark 3.1: Classical Lie groups have one, or in the case of Spin($2n$) two, *fundamental* irreducible representations from which all (infinitely many) other finite-dimensional, unitary representations can be constructed by tensoring the fundamental one(s), symmetrizing in all possible ways and subtracting "traces" using the invariant tensors of the group. The analogous method using Adinkras was sketched in Refs. [21,32]. Thus, Adinkras and the corresponding supermultiplets should represent at least the *fundamental* representations of supersymmetry, from which to build the infinite tower of all others, very much akin to the case of classical Lie algebras.

Without further ado, we focus on the supermultiplets that *can* be depicted by Adinkras:

Definition 3.1 *A supermultiplet $\mathscr{M}$ is* **adinkraic** *if it admits a basis, $(\phi_1, \cdots, \phi_m \,|\, \psi_1, \cdots, \psi_m)$, of component fields such that each $Q_I \in \{Q_1, \cdots, Q_N\}$ acts upon each $\phi_A \in \{\phi_1, \cdots, \phi_m\}$ so as to produce:*

$$Q_I\, \phi_A(\tau) = c\, \partial_\tau^\lambda\, \psi_B(\tau), \quad \textit{where} \quad c = \pm 1, \quad \lambda \in \{0, 1\}, \quad \psi_B \in \{\psi_1, \cdots, \psi_m\}, \tag{3.4a}$$

and the right-hand side choices depend on I and A. In turn, this Q_I acting on this ψ_B produces:

$$Q_I\, \psi_B(\tau) = \frac{i}{c}\, \partial_\tau^{1-\lambda}\, \phi_A(\tau), \tag{3.4b}$$

and the pair of formulae (3.4) exhausts the action of each Q_I upon each component field.

Every adinkraic supermultiplet may be depicted as an Adinkra:

Definition 3.2 *The* **Adinkra**, *$\mathcal{A}_{\mathscr{M}}$, of an adinkraic supermultiplet $\mathscr{M}$ is a labeled, directed graph, (W, E, C, O, D), where W is a bipartitioned set of vertices (V): one half white, the other black; E is the set of edges, each edge connecting two vertices of opposite color; C the set of colorings of the edges; O the set of their orientations; and D the set of their dashedness.*

A vertex is assigned to each component field of $\mathscr{M}$, white for bosons, black for fermions. A transformation rule of the form (3.4a) corresponds to an I-colored edge connecting the vertex corresponding to ϕ_A to the vertex corresponding to ψ_B. If $c = -1$, the edge is dashed, and the edge is oriented from ϕ_A to ψ_B if $\lambda = 0$ and the other way around if $\lambda = 1$; see Table 1.

We can also use the Adinkra to reconstruct the adinkraic supermultiplet, since the Adinkra contains all the information necessary to write down the transformation rules. Thus, an Adinkra is simply a graphical depiction of the transformation rules (3.4).

Remark 3.2: Owing to Remark 3.1, we do not expect the class of *adinkraic supermultiplets* to exhaust the representations of supersymmetry (3.1), but rather to serve as the *fundamental* representations from which to build all the other ones. In addition to the usual construction (tensoring, symmetrizing and taking traces), this will also involve the question when is it is possible for *all* the

Adinkra	Q-action	Adinkra	Q-action
	$Q_I \begin{bmatrix} \psi_B \\ \phi_A \end{bmatrix} = \begin{bmatrix} i\dot{\phi}_A \\ \psi_B \end{bmatrix}$		$Q_I \begin{bmatrix} \psi_B \\ \phi_A \end{bmatrix} = \begin{bmatrix} -i\dot{\phi}_A \\ -\psi_B \end{bmatrix}$
	$Q_I \begin{bmatrix} \phi_A \\ \psi_B \end{bmatrix} = \begin{bmatrix} \dot{\psi}_B \\ i\phi_A \end{bmatrix}$		$Q_I \begin{bmatrix} \phi_A \\ \psi_B \end{bmatrix} = \begin{bmatrix} -\dot{\psi}_B \\ -i\phi_A \end{bmatrix}$

The edges are here labeled by the variable index I; for fixed I, they are drawn in the Ith color.

Table 1: The correspondences between the Adinkra components and supersymmetry transformation formulae: vertices $\leftrightarrow$ component fields; vertex color $\leftrightarrow$ fermion/boson; edge color/index $\leftrightarrow Q_I$; edge dashed $\leftrightarrow c = -1$; and orientation $\leftrightarrow$ placement of ∂_τ. They apply to all ϕ_A, ψ_B within a supermultiplet and all Q_I-transformations amongst them.

supersymmetry transformations in a supermultiplet to conform to Eqs. (3.4) *simultaneously*; we will address this under separate cover. In addition, certain exceptional adinkraic supermultiplets may be depicted, for high enough N, by more than one Adinkra. A case-by-case determination if two given Adinkras correspond to the same supermultiplet is not too difficult, but a systematic study is deferred to a subsequent effort [33].

In view of Remarks 3.1–3.2, the fact that most important physical examples *can* be described using Adinkras reinforces their interpretation as *fundamental* and provides the underlying motivation for our present study. We therefore focus on classifying those supermultiplets that do correspond to Adinkras, and so the question of which supermultiplets cannot be so depicted remains outside our present scope. In turn, Ref. [21] shows that all Adinkras sharing the same topology and their corresponding supermultiplets—and indeed their superfield representations—may be obtained from any one of them; it then remained to classify the topologies available to Adinkras, up to Remarks 3.1–3.2, and we now turn to that.

4 Adinkra Chromotopologies

It is the purpose of this paper to classify the possible *topologies* for Adinkras. To this end, we will need the precise definition:

Definition 4.1 *The* **topology** *of an Adinkra,* $\mathsf{T}(\mathcal{A}_{\mathscr{M}})$*, is the graph* (V, E) *consisting of only the (unlabeled) vertices and edges of the Adinkra; cf. Definition 3.1. Also,* $\mathsf{T}(\mathscr{M}) := \mathsf{T}(\mathcal{A}_{\mathscr{M}})$*.*

In particular, from Definition 3.2 of an Adinkra, we forget the bipartition (black or white) of the vertices and all additional information associated to the edges, namely, the edge-coloring (to which Q_I corresponds), dashedness ($c = \pm 1$), and direction (the exponent of ∂_τ in Eqs. (3.4), and so in fact all ∂_τ).

Definition 4.2 *The* **chromotopology** *of an Adinkra is the topology of the Adinkra, together with the vertex bipartition (coloring each vertex black or white), and the edge coloring (assigning a color to each edge).*

In particular, from Definition 3.2 of an Adinkra, we forget the dashedness of the edges and the direction of the arrow along the edge.

4.1 Cubical Adinkras

The fundamental example of an Adinkra topology is that of the N-cube, $I^N = [0,1]^N$. It has 2^N vertices and $N{\cdot}2^{N-1}$ edges. We may embed it in $\mathbb{R}^N$ by locating the vertices at the points $\vec{p} = (p_1, \cdots, p_N) \in \mathbb{R}^N$, where $p_I = 0, 1$ in all 2^N possible combinations. An edge connects two vertices that differ in precisely one coordinate. For every vertex, $\vec{p}$, the *weight of* $\vec{p}$, written $\mathrm{wt}(\vec{p})$, equals the number of $J \in \{1, \cdots, N\}$ for which $p_J = 1$.

There is a natural chromotopology associated to the N-cube: As the weights of the vertices are either odd or even, we color them either black or white, respectively; flipping this choice is called the 'Klein flip'. We associate the numbers from 1 to N with N different colors, and then color each edge of the N-cube that connects vertices which differ only in the Ith coordinate with the Ith color. The result is called the colored N-cube.

We now construct four supermultiplets: $\mathscr{M}^{\diamond}_{I^N}$, $\mathscr{M}^{=}_{I^N}$ and their Klein-flips, all with the chromotopology of a colored N-cube, I^N.

4.1.1 The Exterior Supermultiplet

We start with a bosonic, real field $\phi_0(\tau)$, write

$$Q^{\vec{p}} := Q_1^{p_1} \cdots Q_N^{p_N}, \tag{4.1}$$

and define all other component fields in reference to $\phi_0(\tau)$:

$$\mathscr{M}^{\diamond}_{I^N} := \left\{ F_{\vec{p}}(\tau) := (-i)^{\binom{\mathrm{wt}(\vec{p})}{2}} Q^{\vec{p}}\, \phi_0(\tau),\ \vec{p} \in I^N \right\}. \tag{4.2}$$

Since

$$\left((-i)^{\binom{\mathrm{wt}(\vec{p})}{2}} Q^{\vec{p}}\right)^{\dagger} = \left((+i)^{\binom{\mathrm{wt}(\vec{p})}{2}}\right)\left((-1)^{\binom{\mathrm{wt}(\vec{p})}{2}} Q^{\vec{p}}\right) = (-i)^{\binom{\mathrm{wt}(\vec{p})}{2}} Q^{\vec{p}}, \tag{4.3}$$

the component fields $F_{\vec{p}}(\tau)$ in Eq. (4.2) are real. Each vertex $\vec{p}$ is marked by a white node if $\mathrm{wt}(\vec{p})$ is even, and Eq. (4.2) assigns it the bosonic component field $F_{\vec{p}}(\tau)$; when $\mathrm{wt}(\vec{p})$ is odd, the assigned $F_{\vec{p}}(\tau)$ is fermionic and its node is black. This binary "cubist" notation, $F_{\vec{p}}$, is not hard to translate into the one more familiar to physicists: each component, $p_I = 0, 1$, specifies if the index I is absent or present, respectively, in the antisymmetrized multi-index of the component field. Thus, *e.g.*, $F_{000} = F = \phi_0(\tau)$, $F_{100} = F_1$, $F_{101} = F_{[13]}$, $F_{011} = F_{[23]}$, $F_{111} = F_{[123]}$, where the antisymmetrization on multiple indices stems from the anticommutivity of the Q_I's.

Next, using that

$$Q_I\, Q^{\vec{p}} = (-1)^{\mathrm{wt}(\vec{p}<I)}\, (i\partial_\tau)^{\vec{p}\cdot\vec{e}_I}\, Q^{\vec{p}\boxplus\vec{e}_I}, \tag{4.4}$$

where $\mathrm{wt}(\vec{p} < I)$ counts the number of nonzero components of $\vec{p}$ before the Ith, $\vec{e}_I$ is the unit vector in the Ith direction and

$$\vec{p} \boxplus \vec{e}_I \equiv (\vec{p} + \vec{e}_I) \pmod 2, \tag{4.5}$$

is the component-wise exclusive or, we compute by direct application of Q_I:

$$Q_I\, F_{\vec{p}}(\tau) = (-1)^{\mathrm{wt}(\vec{p}<I)+(\vec{p}\cdot\vec{e}_I)(\mathrm{wt}(\vec{p})+1)}\, i^{\mathrm{wt}(\vec{p})}\, \left(\partial_\tau^{(\vec{p}\cdot\vec{e}_I)} F_{\vec{p}\boxplus\vec{e}_I}(\tau)\right). \tag{4.6}$$

Thus, for each I, the Q_I-transformation connects precisely those two vertices that differ in the Ith coordinate. The corresponding edge in the Adinkra, drawn in the Ith color, is oriented from the vertex where that coordinate is 0 to where it is 1. It is either solid or dashed depending on the exponents of (-1) and i. This makes the topology of the N-cube manifest and also proves that $\mathscr{M}^{\diamond}_{I^N}$ is adinkraic, seeing that Eq. (4.6) conforms precisely to Eqs. (3.4).

The Klein-flip of $\mathscr{M}^{\diamond}_{I^N}$ is obtained by starting from a fermion, $\psi_0(\tau)$, in place of a boson, $\phi_0(\tau)$.

The engineering dimensions of the component fields of $\mathscr{M}^{\diamond}_{I^N}$ are:

$$[F_{\vec{p}}] = w_0 + \tfrac{1}{2}\,\mathrm{wt}(\vec{p}), \tag{4.7}$$

where w_0 is a constant determined by the choice of a Lagrangian. The number of component fields, listed by their increasing engineering dimension, determines the $\mathbb{Z}$-graded dimension of this representation:

$$\begin{aligned} \dim(\mathscr{M}^{\diamond}_{I^N}) &= \dim\big(F_{0\cdots 0} = \phi_0 | F_{10\cdots 0}, F_{010\cdots 0}, \cdots | \cdots | F_{11\cdots 1}\big), \\ &= \Big(1\big|N\big|\tbinom{N}{2}\big|\tbinom{N}{3}\big|\cdots\big|\tbinom{N}{N-1}\big|\tbinom{N}{N}\Big), \end{aligned} \tag{4.8}$$

so that there are 2^{N-1} bosonic and 2^{N-1} fermionic component fields in $\mathscr{M}^{\diamond}_{I^N}$.

Adinkras corresponding to $\mathscr{M}^{\diamond}_{I^N}$ were called Top Clifford Algebra superfields [13]. The supermultiplets themselves will be recognized by supersymmetry practitioners to be also representable as real, "unconstrained" Salam–Strathdee superfields, in the familiar θ-expansion of which $F_{\vec{p}}(\tau)$ occurs as the coefficient of the $\theta_1^{p_1}\cdots\theta_N^{p_N}$ monomial. Since the superspace θ's generate an *exterior* algebra, we refer to $\mathscr{M}^{\diamond}_{I^N}$ as the *exterior supermultiplet*.

In closing this section, let us note the definition of the component fields introduced above is sufficient for the purposes of the present work as our goals simply involved studying the representation theory of one-dimensional supersymmetry. If we were to consider the case of some dynamical theory, *i.e.*, write a superfield Lagrangian to specify some dynamics, then an alternative definition of the components would be necessary, as was done in Ref. [22], for example. In all manifestly supersymmetrical theories, there exist 'twisted' versions of the supercharges denoted by $\mathrm{D}_1,\ldots,\mathrm{D}_N$, satisfying the following relations:

$$\big\{\, Q_I\,,\,\mathrm{D}_J\,\big\} = 0, \quad \big\{\,\mathrm{D}_I\,,\,\mathrm{D}_J\,\big\} = 2i\,\delta_{IJ}\,\partial_\tau, \quad \big[\,\partial_\tau\,,\,\mathrm{D}_I\,\big] = 0, \quad I,J = 1,\ldots,N. \tag{4.9}$$

An operator analogous to (4.1) may be defined by

$$\mathrm{D}^{\vec{p}} := \mathrm{D}_1^{p_1}\cdots\mathrm{D}_N^{p_N}, \tag{4.10}$$

and this is applied to superfields to define components upon taking the limit as all Grassmann coordinate vanish. This entire process is known as defining components by projection.

4.1.2 The Clifford Supermultiplet

Another important supermultiplet, $\mathscr{M}^{=}_{I^N} := \{\bar{F}_{\vec{p}}(\tau), \vec{p}\in\{0,1\}^N\}$, may be obtained from $\mathscr{M}^{\diamond}_{I^N}$ *via* the *non-local* transform:

$$\bar{F}_{\vec{p}}(\tau) := \partial_\tau^{\lfloor (N-\mathrm{wt}(\vec{p}))/2\rfloor}\,F_{\vec{p}}(\tau), \qquad i.e., \qquad F_{\vec{p}}(\tau) = \partial_\tau^{\lfloor (\mathrm{wt}(\vec{p})-N)/2\rfloor}\,\bar{F}_{\vec{p}}(\tau). \tag{4.11}$$

Remark 4.1: Since the transformation $\mathscr{M}^{\diamond}_{I^N} \leftrightarrow \mathscr{M}^{=}_{I^N}$ is non-local for $N > 1$, these two supermultiplets must be regarded as distinct. With increasing N, it is clear that there is also a combinatorially growing multitude of "intermediate" supermultiplets, all having the same, N-cubical topology, and obtainable one from another by means of non-local transformations of the kind (4.11) [21].

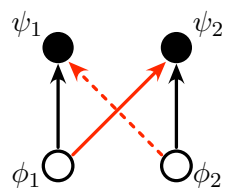
Notice that, in contrast to Eq. (4.7),

$$[\bar{F}_{\vec{p}}] = \begin{cases} w_0 + \lfloor \frac{N}{2} \rfloor & \text{if wt}(\vec{p}) \text{ is even,} \\ w_0 + \lfloor \frac{N}{2} \rfloor + \frac{1}{2} & \text{if wt}(\vec{p}) \text{ is odd.} \end{cases} \tag{4.12}$$

The Adinkra corresponding to the supermultiplet $\mathscr{M}^{=}_{I^N}$ was called the "base" Adinkra in Ref. [13]. Also, all off-shell bosonic component fields have the same engineering dimension, w_0, and so do all fermionic ones, equal to $w_0+\frac{1}{2}$. The number of component fields, listed by their increasing engineering dimension, is $(2^{N-1}|2^{N-1})$—considerably simpler than Eq. (4.8).

Example 4.1 The simplest nontrivial example of a Clifford supermultiplet occurs at $N = 2$ and consists of two bosons ϕ_1, ϕ_2 and two fermions ψ_1, ψ_2, where the two supersymmetry generators act as follows:

$$Q_1\,\phi_1 = \psi_1, \tag{4.13a}$$
$$Q_1\,\phi_2 = \psi_2, \tag{4.13b}$$
$$Q_1\,\psi_1 = i\dot{\phi}_1, \tag{4.13c}$$
$$Q_1\,\psi_2 = i\dot{\phi}_2, \tag{4.13d}$$

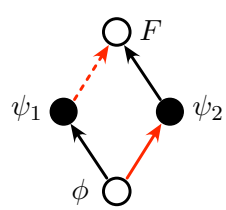


$$Q_2\,\phi_1 = \psi_2, \tag{4.13a}$$
$$Q_2\,\phi_2 = -\psi_1, \tag{4.13b}$$
$$Q_2\,\psi_1 = -i\dot{\phi}_2, \tag{4.13c}$$
$$Q_2\,\psi_2 = i\dot{\phi}_1, \tag{4.13d}$$

The black (vertical) edges represent Q_1-action and red (diagonal) edges that of Q_2. Forgetting either one of Q_1, Q_2, we remain with two *disjoined* $N = 1$ supermultiplets. The Q_2-action involves here a scaling factor of $c = -1$ in one pair, depicted by a dashed edge. This could be remedied by redefining ψ_1, for instance, but only to induce a $c = -1$ scaling factor in the Q_1-action $\phi_1 \leftrightarrow \psi_1$. No such redefinition will eliminate the need for a minus sign in at least one pair of the transformations (4.13a)–(4.13d).

The $N = 2$ exterior supermultiplet and Adinkra is:

$$Q_1\,\phi = \psi_1, \tag{4.14a}$$
$$Q_1\,\psi_1 = i\dot{\phi}, \tag{4.14b}$$
$$Q_1\,\psi_2 = iF, \tag{4.14c}$$
$$Q_1\,F = \dot{\psi}_2, \tag{4.14d}$$



$$Q_2\,\phi = \psi_2, \tag{4.14a}$$
$$Q_2\,\psi_1 = -iF, \tag{4.14b}$$
$$Q_2\,\psi_2 = i\dot{\phi}, \tag{4.14c}$$
$$Q_2\,F = -\dot{\psi}_1, \tag{4.14d}$$

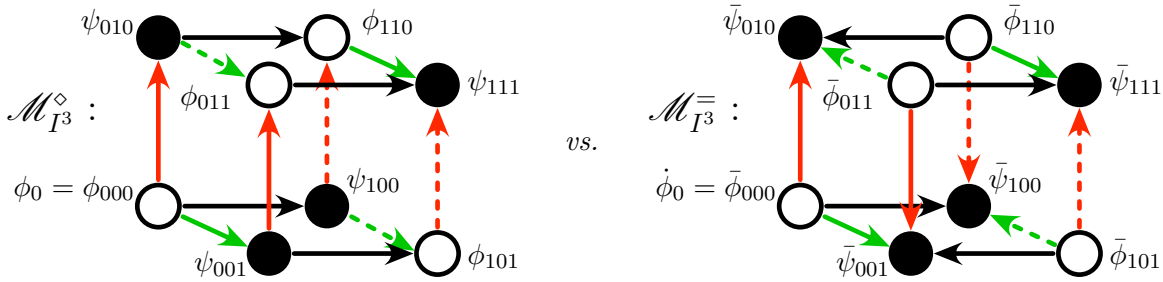
Notice that $(\phi, F|\psi_1, \psi_2) \to (\phi_1, \dot{\phi}_2|\psi_1, \psi_2)$ and $(\phi_1, \phi_2|\psi_1, \psi_2) \to (\phi, \partial_\tau^{-1}F|\psi_1, \psi_2)$ is the manifestly non-local bijective field redefinition between the two supermultiplets.

Together with their Klein-flips, these are all the $N = 2$ Adinkras and supermultiplets [13,21].

Example 4.2 As another example, let $N = 3$. For $\mathscr{M}^{\diamond}_{I^3}$, denote the component fields $F_{\vec{p}}$ alternatively as $\phi_{\vec{p}}$ for wt$(\vec{p})$ even (bosons), and $\psi_{\vec{p}}$ for wt$(\vec{p})$ odd (fermions). For $\mathscr{M}^{=}_{I^3}$, denote the component fields $\bar{\phi}_{\vec{p}}$ and $\bar{\psi}_{\vec{p}}$, respectively, where Eq. (4.11) implies, for example:

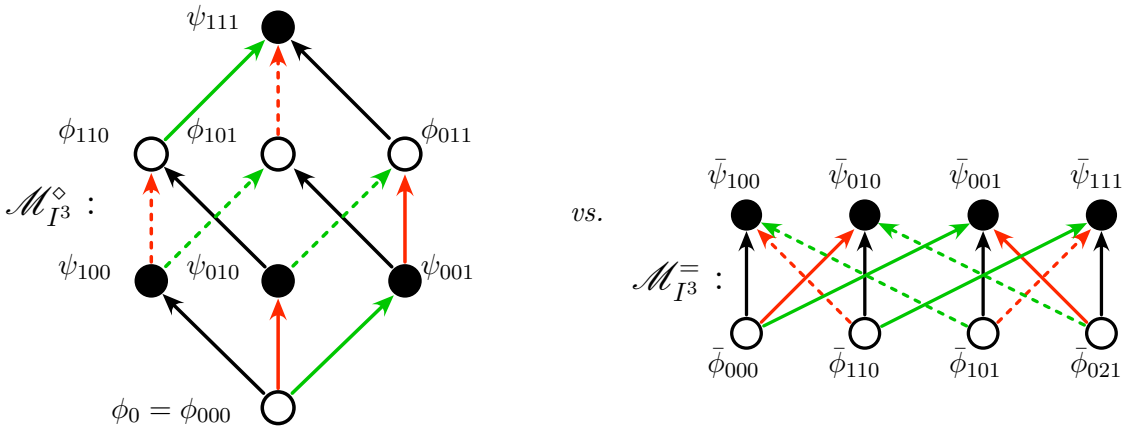
$$\bar{\phi}_{000} = \dot{\phi}_{000}, \qquad \bar{\psi}_{100} = \dot{\psi}_{100}, \qquad \bar{\phi}_{110} = \phi_{110}, \qquad \bar{\psi}_{111} = \psi_{111}. \tag{4.15}$$

Following Definition 3.2, we then draw:



(4.16)

The distinction between these becomes clearer in the convention of Ref. [21], with the nodes drawn at a height proportional to the engineering dimension of the corresponding component field:



(4.17)

The Adinkra corresponding to the Klein-flip of $\mathscr{M}^{\diamond}_{I^N}$ is obtained from Eq. (4.16) by reversing the vertex coloring, black $\leftrightarrow$ white.

We now turn to determine what other topologies are available to Adinkras, and correspondingly to supermultiplets.

4.2 Quotients of the N-Cubes

Given an N-cubical Clifford supermultiplet, $\mathscr{M}^{=}_{I^N}$, a novel opportunity emerges for $N \geq 4$: Certain identifications amongst the 2^{N-1} bosons and the 2^{N-1} fermions are possible such that the induced action of the $Q_1, \cdots, Q_N, \partial_\tau$ upon the components of the so projected supermultiplet remains consistent with Eqs. (3.1). This is *not* possible for $N < 4$.

Example 4.3 For example, attempting to identify $\phi_1 = \phi_2$ in Eqs. (4.13) leads either to a trivial supermultiplet consisting of a single bosonic constant, or to an immediate contradiction between the left-hand side and the right-hand side equations (4.13a) and (4.13b):

$$Q_1\,\phi_1(\tau) = \psi_1(\tau), \qquad Q_2\,\phi_1(\tau) = \psi_2(\tau), \tag{4.13a$'$}$$

$$Q_1\,\phi_1(\tau) = \psi_2(\tau), \qquad Q_2\,\phi_1(\tau) = -\psi_1(\tau). \tag{4.13b$'$}$$

From the left-hand side pair, it follows that $\psi_2(\tau) = +\psi_1(\tau)$, whereas the right-hand side pair implies $\psi_2(\tau) = -\psi_1(\tau)$. This is consistent only if $\psi_1(\tau) = 0 = \psi_2(\tau)$, whereupon Eq. (3.1) implies that $\partial_\tau \phi_1(\tau) = 0$, reducing this supermultiplet to a trivial, single constant boson. Alternately, adding the left-hand side Eq. (4.13a$'$) to the right-hand side Eq. (4.13b$'$) leads to $Q_1\,\phi_1 + Q_2\,\phi_1 = 0$. In a similar manner, subtracting the second result in Eq. (4.13a$'$) from the first result in Eq. (4.13b$'$) leads to $Q_1\,\phi_1 - Q_2\,\phi_1 = 0$. Together, these imply that $Q_1\,\phi_1 = Q_2\,\phi_1 = 0$.

4.2.1 The $N = 4$ Projection

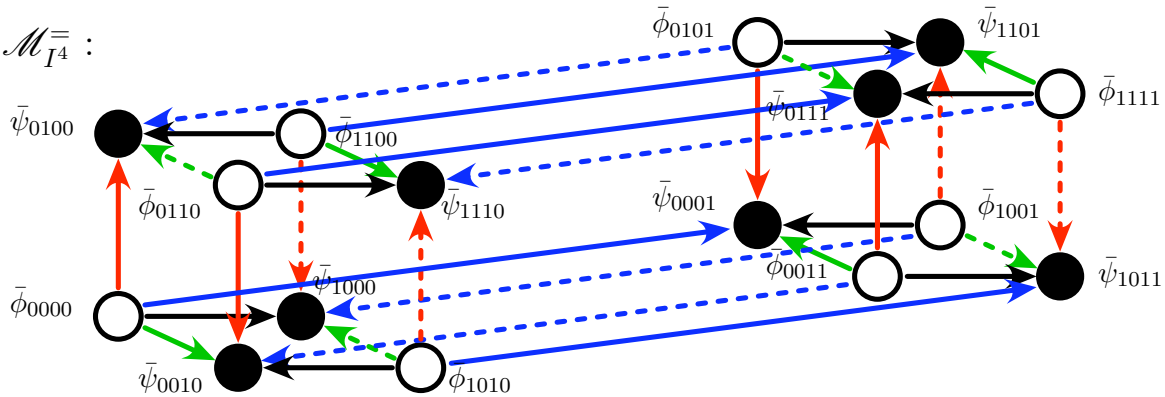
To clarify how such projections *can* occur, we examine the simplest of them, for $N = 4$.

Example 4.4 The $N = 4$ Isoscalar supermultiplet (*i.e.*, the Clifford supermultiplet with bosons on the bottom) is

$$\mathscr{M}_{I^4}^{=} = \{\, \bar{\phi}_{0000}, \bar{\phi}_{1100}, \ldots, \bar{\phi}_{1111} | \bar{\psi}_{1000}, \ldots, \bar{\phi}_{1110}, \ldots \}, \tag{4.18}$$

where the ellipses indicate component fields the subscript of which is obtained as a permutation on the one preceding the ellipses. The Adinkra of this supermultiplet is:

$\mathscr{M}_{I^4}^{=}$:



(4.19)

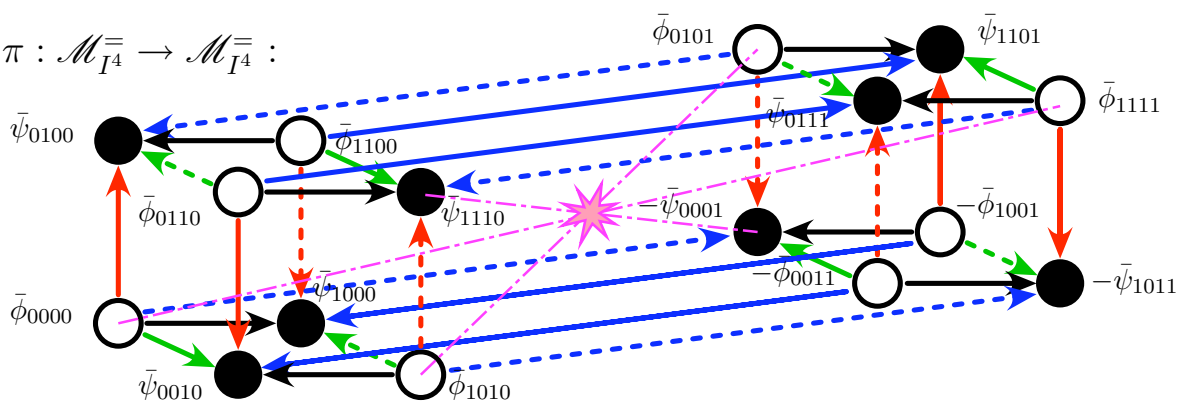
By replacing, for convenience, in the bottom square of the right-hand side 3-cube:

$$\bar{\phi}_{0011} \to -\bar{\phi}_{0011}\ , \qquad \bar{\phi}_{1001} \to -\bar{\phi}_{1001}\ , \tag{4.20}$$

$$\bar{\psi}_{0001} \to -\bar{\psi}_{0001}\ , \qquad \bar{\psi}_{1011} \to -\bar{\psi}_{1011}\ , \tag{4.21}$$

the Adinkra (4.19) becomes:

$\pi : \mathscr{M}_{I^4}^{=} \to \mathscr{M}_{I^4}^{=}$:



(4.22)

where the reflection across the central point marked by the star is a symmetry of the Adinkra, including all the labelings. For the sake of clarity in this diagram, we have only indicated three of the eight vertex pairs being reflected into each other by the dot-dash purple lines. The full set of reflection pairs is:

$$\bar{\phi}_{0000} \leftrightarrow \bar{\phi}_{1111}\ , \qquad -\bar{\phi}_{0011} \leftrightarrow \bar{\phi}_{1100}\ , \qquad -\bar{\psi}_{0001} \leftrightarrow \bar{\psi}_{1110}\ , \qquad \bar{\psi}_{0010} \leftrightarrow \bar{\psi}_{1101}\ , \tag{4.23a}$$

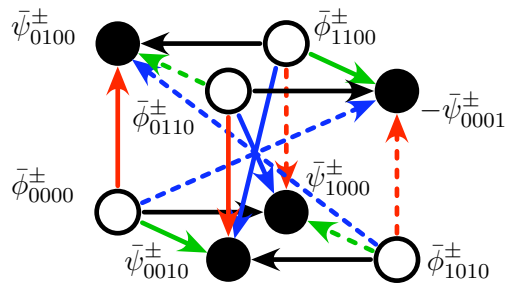
$$\bar{\phi}_{0101} \leftrightarrow \bar{\phi}_{1010}\ , \qquad \bar{\phi}_{1001} \leftrightarrow \bar{\phi}_{0110}\ , \qquad \bar{\psi}_{0100} \leftrightarrow -\bar{\psi}_{1011}\ , \qquad \bar{\psi}_{1000} \leftrightarrow \bar{\psi}_{0111}\ . \tag{4.23b}$$

It is thus possible to "orbifold" the Adinkra (4.22) by identifying, *e.g.*, $\bar{\phi}_{0000}^{\pm}$ with $\bar{\phi}_{0000} \pm \bar{\phi}_{1111}$, and consequently also identifying each two such vertices and so also the corresponding component fields:

$$\pi\ :\quad (p_1, p_2, p_3, p_4)\ \mapsto\ (1-p_1, 1-p_2, 1-p_3, 1-p_4), \qquad p_I \in \{0,1\}. \tag{4.24}$$

Both resulting Adinkras have the topology of (the 1-skeleton of) the projective hypercube in $N = 4$:

$(\mathscr{M}^{=}_{I^4}/\mathbb{Z}_2)^{\pm}$:



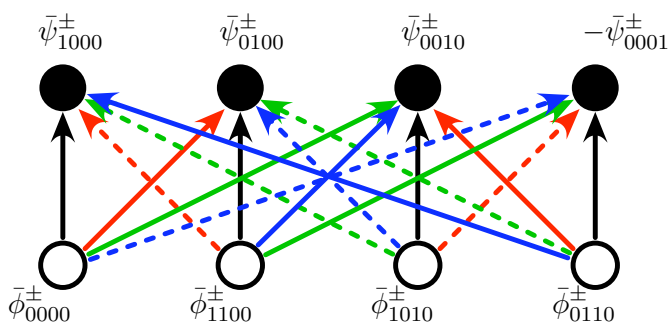
(4.25)

where we have defined

$$(\mathscr{M}^{=}_{I^4}/\mathbb{Z}_2)^{\pm}: \quad \bar{F}^{\pm}_{\vec{p}}(\tau) := \bar{F}_{\vec{p}}(\tau) \pm (-1)^{p_2+p_4}\bar{F}_{\vec{1}-\vec{p}}(\tau) \tag{4.26}$$

for the component fields corresponding to the orbifolded Adinkras. As it turns out, the dimensional reduction of the $d = 4$, $\mathcal{N} = 1$ chiral supermultiplet has the topology (4.25) [21] but the height assignments (engineering dimensions) of two of its bosons differ from the ones shown here. Note that every boson is connected to every fermion, and *vice versa*. This becomes clearer in the convention of Ref. [21], where we place the vertices at a height proportional to the engineering dimensions of the corresponding component fields, so that all arrows point upward:

$(\mathscr{M}^{=}_{I^4}/\mathbb{Z}_2)^{\pm}$:



(4.27)

In graph theory, this is known as $K(4, 4)$, the complete bipartite graph connecting four bosons with four fermions. Though it may appear at first that there could be further identifications, a closer look at the Adinkra (4.27) reveals that the locations of the dashed lines prevent all further identification; that is, no further projection is consistent with Eq. (3.1).

Remark 4.2: A combination of Eq. (4.26) and Eq. (4.11) shows that attempting an analogous projection in $\mathscr{M}^{\diamond}_{I^4}$ would require the definitions:

$$\text{“}(\mathscr{M}^{\diamond}_{I^4}/\mathbb{Z}_2)^{\pm}\text{”}: \quad F^{\pm}_{\vec{p}}(\tau) := \partial_\tau^{\lfloor(4-\mathrm{wt}(\vec{p}))/2\rfloor}F_{\vec{p}}(\tau) \pm (-1)^{p_2+p_4}\partial_\tau^{\lfloor(\mathrm{wt}(\vec{p}))/2\rfloor}F_{\vec{1}-\vec{p}}(\tau). \tag{4.28}$$

The appearance of the ∂_τ's indicates that this is not a quotient of the exterior supermultiplet $\mathscr{M}^{\diamond}_{I^4}$ itself, but of a multiple *vertex-raise* [21] of $\mathscr{M}^{\diamond}_{I^4}$, which is in turn isomorphic to the $\mathscr{M}^{=}_{I^4}$. It should be manifest that the differing engineering dimensions of the component fields in $\mathscr{M}^{\diamond}_{I^4}$ (and the locality requirements) present the key obstruction to such a $\mathbb{Z}_2$ projection in $\mathscr{M}^{\diamond}_{I^4}$. This proves that the possibility to construct a local $\mathbb{Z}_2$-quotient of a supermultiplet strongly depends on its component fields' engineering degrees: while $\mathscr{M}^{=}_{I^4}$ does have a local $\mathbb{Z}_2$-quotient, $\mathscr{M}^{\diamond}_{I^4}$ does not.

4.2.2 $N > 4$ Projections

More generally, given an Adinkra, we can sometimes identify vertices, if all labeling of the corresponding vertices and edges matches up. This results is a *quotient Adinkra*. A correspondingly projected supermultiplet is a *quotient supermultiplet*. We will now focus on quotients of the N-cube. We show in Section 4.3 that all connected Adinkra topologies can be obtained in this way.

We start with an Adinkra with the topology of an N-cube, the vertices of which are the N-tuples $(x_1, \cdots, x_N) \in \{0,1\}^N$. Suppose that the desired projection, π, is to identify $\vec{0} = (0, \cdots, 0)$ with some other vector $\vec{x} = (x_1, \cdots, x_N)$. Now consider $(1, 0, \cdots, 0)$, which is connected to $\vec{0}$ by an edge of color 1. There is also an edge of the same color, 1, incident with the vertex at $(x_1, \cdots, x_N)$, and this edge connects $(x_1, \cdots, x_N)$ to $(1 - x_1, x_2, \cdots, x_N)$. If the edges are to match up, π must also identify $(1, 0, \cdots, 0)$ with $(1 - x_1, x_2, \cdots, x_N)$. Following similarly the edges of all other colors, we can prove by induction that if $(v_1, \cdots, v_N)$ is any vertex of $\{0,1\}^N$, then π identifies $(v_1, \cdots, v_N)$ with $(x_1 + v_1, \cdots, x_N + v_N) \pmod 2$; we write $\vec{v} \equiv_\pi \vec{v} \boxplus \vec{x}$. Thus, every projection π is completely determined by the vertex which it identifies with $\vec{0}$.

More generally, the projection π may identify more than one vertex with $\vec{0}$. Suppose $\vec{x}$ and $\vec{y}$ are two such vertices, identified with $\vec{0}$. Then π establishes an equivalence relation so that $\vec{0} \equiv_\pi \vec{x}$, and $\vec{x} \equiv_\pi \vec{x} \boxplus \vec{y}$. Thus, $\vec{0}$ is identified with $\vec{x} \boxplus \vec{y}$.

Now, we note that the inverse $-\vec{x}$ is equal to $\vec{x}$ modulo 2. Therefore, the set of vertices that are to be identified with $\vec{0}$ forms a group under component-wise addition modulo 2. Since we must identify only bosons with bosons and only fermions with fermions, $\mathrm{wt}(\vec{x})$ must be even. As we will see in § 4.3, the weight $\mathrm{wt}(\vec{x})$ must actually be a multiple of 4, or the dashedness of the edges cannot possibly match up.

This yields the following construction:

1. Start with an Adinkra, $\mathcal{A}$, with the topology an N-cube, $I^N = [0,1]^N$.
2. Let G be a subgroup of $(\mathbb{Z}_2)^N$ consisting only of vectors $\vec{x}$ with $\mathrm{wt}(\vec{x}) \equiv 0 \pmod 4$.
3. Let G act on the vertex set $\{0,1\}^N$ by component-wise addition modulo 2.
4. If this preserves the dashing of the edges, then $\mathcal{A}/G$ is an Adinkra whose topology is I^N/G, the corresponding quotient of the N-cube.

4.3 Finding all Adinkra Chromotopologies

Suppose an Adinkra is not connected. Then the corresponding supermultiplet splits into a direct sum, each component of which corresponds to a connected component of the Adinkra. Now it is possible for a connected Adinkra to correspond to a direct sum, though it may not be immediately apparent by the Adinkra. This issue will be discussed fully in Ref. [33], where we also specify precisely the conditions under which a supermultiplet may be represented by two topologically distinct Adinkras. For now, all we need to note is that the problem of classifying Adinkras reduces to classifying connected Adinkras.

As we will see, every connected Adinkra chromotopology is obtained by taking a colored N-dimensional cube, then possibly identifying nodes and edges using a doubly even code. Thus, the

topology of an Adinkra, obtained by forgetting the coloring of the edges and vertices, arises from quotienting an (uncolored) N-dimensional cube using a doubly even code. First, some notation.

Suppose we have an Adinkra for N-extended supersymmetry, and we consider the chromotopology. That is, we ignore the arrows on the edges and ignore whether an edge is dashed. What is left is a vertex set $V = \{v_1, \cdots, v_{2m}\}$, corresponding to all component fields $(F_1(\tau), \cdots, F_{2m}(\tau))$,[3], the coloring of these vertices, and the edge set E with its coloring. Since this graph is inherited from a proper Adinkra, for every vertex and every $I \in \{1, \ldots, N\}$ there is an edge corresponding to applying Q_I to the field corresponding to that vertex. Applying this Q_I may involve derivatives, a sign, and/or a factor of i, but it will be convenient for now to suppress this. To this end, we define functions $q_1, \cdots, q_N$ from the vertex set $V = \{v_1, \cdots, v_{2m}\}$ to itself, such that whenever A, B, and I are such that there is an equation of type (3.4a) or (3.4b),

$$Q_I\, F_A(\tau) = c\, \partial_\tau^\lambda F_B, \qquad \Longrightarrow \qquad q_I(v_A) = v_B. \tag{4.29}$$

Notice that in defining q_I from Q_I, we are forgetting the coefficients $c = \pm 1$, as well as the ∂_τ's which encode the differences in engineering dimensions. The supersymmetry algebra (3.1) then implies:

$$q_I^2 = \mathbb{1}, \qquad i.e. \qquad q_I\big(q_I(v)\big) = v, \text{ for all } v \in V, \tag{4.30}$$

and

$$q_I q_J = q_J q_I, \qquad i.e. \qquad q_I\big(q_J(v)\big) = q_J\big(q_I(v)\big), \text{ for all } I, J, \text{ and all } v \in V. \tag{4.31}$$

Theorem 4.1 *Every connected Adinkra chromotopology is a quotient of a colored N-dimensional cube.*

Proof: Suppose we have a connected Adinkra with N edge colors. Let $V = \{v_1, \cdots, v_{2m}\}$ be the set consisting of all its vertices. Now pick, without loss of generality, any one bosonic vertex $v_* \in V$ and fix it. We take the colored N-cube $[0,1]^N$, and consider its vertex set $\{0,1\}^N$. We then define a mapping

$$\pi : \begin{cases} \{0,1\}^N \to V, \\ (x_1, \cdots, x_N) \mapsto q_1^{x_1}\big(\cdots q_N^{x_N}(v_*)\big), \end{cases} \tag{4.32}$$

and write $\pi(\vec{x}) := q_1^{x_1}\big(\cdots q_N^{x_N}(v_*)\big)$.

For any $\vec{x} \in \{0,1\}^N$, we apply Eq. (4.32) to v_*. Then, for any $I \in \{1, \cdots, N\}$, apply q_I to $\pi(\vec{x})$:

$$q_I\big(\pi(\vec{x})\big) = q_I\left(q_1^{x_1}\big(\cdots q_N^{x_N}(v_*)\big)\right) = q_1^{x_1}\Big(\cdots\, q_I^{1-x_I}\big(\cdots\, q_N^{x_N}(v_*)\big)\Big), \tag{4.33}$$

$$= \pi(\vec{x} \boxplus \vec{e}_I), \qquad \forall v_* \in V,\ \forall \vec{x} \in \{0,1\}^N, \tag{4.34}$$

where we have applied Eq. (4.31) to commute q_I through the other q's in Eq. (4.33) and so obtain Eq. (4.34), using the notation (4.5) for the component-wise "exclusive or" operation.

[3] Previously, we labeled the bosons $\phi_1, \cdots, \phi_m$ and the fermions $\psi_1, \cdots, \psi_m$. Here, it will be convenient for notation to treat them on the same footing.

Thus, each pair of vertices $(\pi(\vec{x}), \pi(\vec{x} \boxplus \vec{e}_I)) \in V \times V$ is connected by an edge, labeled by I, in the Adinkra. Since there is an edge labeled by I connecting $\vec{x}$ with $(\vec{x} \boxplus \vec{e}_I)$ in the N-cube, and since all edges in the cube $[0,1]^N$ are of this form, the map (4.32) induces a map π_E that maps edges colored I of $[0,1]^N$ to edges colored I in the Adinkra.

To see that π is surjective, first observe that the Adinkra is connected. That is, every vertex $v \in V$ is connected, via a path of edges, to the fixed $v_* \in V$. In the Adinkra, these edges have colors, forming a sequence, $I_1, \cdots, I_k$, when tracing from the vertex v to v_*. If we then apply to v_* a corresponding sequence of q_I's, we get:

$$q_{I_1}\big(\cdots q_{I_k}(v_*)\big) = v. \tag{4.35}$$

Using the commutativity of the q's to put them in numerical order and Eq. (4.30) to eliminate those that appear more than once, we can write this as

$$\pi(\vec{x}) = q_1^{x_1}\big(\cdots q_N^{x_N}(v_*)\big) = v. \tag{4.36}$$

The fact that π sends bosons to bosons and fermions to fermions can be seen by the fact that it sends the boson $(0,\ldots,0) \in \{0,1\}^N$ to the boson v_*, and the fact that every vertex in V is connected to v_* by a sequence of edges, each of which alternates between bosons and fermions.

Let $G = \pi^{-1}(v_*)$, be the collection of points $\vec{x} \in \{0,1\}^N$, which Eq. (4.32) maps to v_*. Being a subset of $\{0,1\}^N$, G may be interpreted as a subset of $(\mathbb{Z}_2)^N$.

We will now show G is a subgroup of $(\mathbb{Z}_2)^N$. In fact, as we will see, G is the group such that $[0,1]^N/G$ is the chromotopology of the Adinkra in question. Trivially, $\vec{0} \in G$. Inverses exist since every element is its own inverse in $(\mathbb{Z}_2)^N$. Finally, let $\vec{x}$ and $\vec{y}$ be elements of G. Compose $q_1^{x_1}\ldots q_N^{x_N}$ with $q_1^{y_1}\ldots q_N^{y_N}$. The composition sends v_* to itself, and we can commute the q_I past each other to write $q_1^{x_1+y_1}\cdots q_N^{x_N+y_N}$. Thus,

$$q_1^{x_1+y_1}\cdots q_N^{x_N+y_N}(v_*) = v_*. \tag{4.37}$$

Since $q_I^2 = 1$, we can reduce the exponents of these modulo 2, and thus, $\pi(\vec{x} \boxplus \vec{y}) = v_*$. Thus, $\vec{x} \boxplus \vec{y} \in G$. Therefore, G is a group.

We now define a bijection between $\{0,1\}^N/G$ and the vertex set V of the Adinkra. The function $\pi : \{0,1\}^N \to V$ has the property that for all $\vec{x} \in G$, $\pi(\vec{y} + \vec{x}) = \pi(\vec{y})$. Thus it is well-defined to define a function on the cosets of G, $f : \{0,1\}^N/G \to V$, so that $f(\vec{y} + G) = \pi(\vec{y})$.

The function f is one-to-one, since if $\pi(\vec{y}) = \pi(\vec{z})$, we would have

$$q_1^{y_1}\ldots q_N^{y_N}(v_*) = q_1^{z_1}\ldots q_N^{z_N}(v_*), \tag{4.38}$$

and this would imply

$$q_1^{y_1+z_1}\ldots q_N^{y_N+z_N}(v_*) = v_*, \tag{4.39}$$

so that $\vec{y} \boxplus \vec{z} \in G$. The function f is onto, since π is, as was shown above. Thus, f is a bijection on vertices. The argument that π sends edges to edges shows that the edges are similarly in bijection. We therefore see that G is equal to the group of identifications on $\{0,1\}^N$, and the orbit space $\{0,1\}^N/G$ is the Adinkra. The map π realizes this quotient. We can use π to pull back all the

labels from the Adinkra to $\{0,1\}^N$, and in this way, we have a cubical Adinkra the quotient of which by G is the Adinkra in question. ☑

We next consider what kinds of groups G can be used to quotient a cubical Adinkra. First, note that G is a subgroup of $(\mathbb{Z}_2)^N$. As we saw in Section 2, G is then a binary linear code. In the analysis to far, we have used only the edge-coloring in the Adinkra.

Remembering also the vertex-coloring will force G to be even, that is, for each $\vec{x} \in G$, $\mathrm{wt}(\vec{x})$ must be even. As we will see shortly, remembering the edge-dashedness will force G to be *doubly* even, that is, for each $\vec{x} \in G$, $\mathrm{wt}(\vec{x})$ must be a multiple of 4.

Theorem 4.2 *If an Adinkra is a quotient of an N-cube by a group G, then G is a doubly even binary linear code.*

Remark 4.3: The converse is also true, that is if G is a doubly even binary linear code, then there is a family of adinkraic supermultiplets, the chromotopology of the Adinkra of which is a quotient by G of a colored N-cube. In order to prove this fact, we must first discuss the relationship between supersymmetry and Clifford algebras, which will be treated in the sequel, Ref. [33], in which for every doubly even code, G, we construct a family of adinkraic supermultiplets with the $\{0,1\}^N/G$ chromotopology.

Proof: Recall from Section 2 that a binary linear code is a subgroup of $(\mathbb{Z}_2)^N$, and that it is called doubly even if every element of it has weight a multiple of 4. So we need to prove that for every $x \in G$, $\mathrm{wt}(x) \equiv 0 \pmod 4$.

To this end, the q_I's no longer suffice, and we will need to use the Q_I's; in particular, we need to "remember" the scaling constants c encoding the edge-dashedness in an Adinkra, and the equipartition of the vertices into bosonic and fermionic ones.

The statement that $\vec{x} \in G$ means that

$$\pi(\vec{x}) = q_1^{x_1}\big(\cdots q_N^{x_N}(v_*)\big) = v_*, \tag{4.40}$$

and thus

$$Q_1^{x_1}\cdots Q_N^{x_N}\, F_1(\tau) = C\,\partial_\tau^{\mathrm{wt}(\vec{x})/2}\, F_1(\tau) \tag{4.41}$$

for some complex number C. The exponent of ∂_τ follows simply from comparing engineering dimensions of the left-hand side and the right-hand side.

Since this sequence of Q_I operators, corresponding to a closed path in the Adinkra, must send bosons to bosons and fermions to fermions, it must be that $\mathrm{wt}(\vec{x})$ is even, so that $\frac{1}{2}\,\mathrm{wt}(\vec{x})$ is indeed integral and Eq. (4.41) is well-defined.

Applying $Q_1^{x_1}\cdots Q_N^{x_N}$ twice to $F_1(\tau)$, we find:

$$Q_1^{x_1}\cdots Q_N^{x_N}\cdot Q_1^{x_1}\cdots Q_N^{x_N} F_1(\tau) = C^2\partial_\tau^{2\Lambda} F_1(\tau). \tag{4.42}$$

On the left side, using the supersymmetry algebra (3.1), we can anti-commute the Q_I past each other. Rearrange these to regroup the result into

$$= (-1)^{\binom{\mathrm{wt}(\vec{x})}{2}} Q_1^{2x_1}\cdots Q_N^{2x_N} F_1(\tau), \tag{4.43}$$

which, using the supersymmetry algebra (3.1) again), becomes

$$= (-1)^{\binom{\mathrm{wt}(\vec{x})}{2}}\, i^{\mathrm{wt}(\vec{x})}\, \partial_\tau^{\mathrm{wt}(\vec{x})} F_1(\tau). \tag{4.44}$$

Comparing the exponents of ∂_τ in Eq. (4.42) with that in Eq. (4.44) confirms Eq. (4.41). Comparing the numerical coefficients produces:

$$C^2 = (-1)^{\binom{\mathrm{wt}(\vec{x})}{2}}\, i^{\mathrm{wt}(\vec{x})} = i^{\mathrm{wt}(\vec{x})(\mathrm{wt}(\vec{x})-1)+\mathrm{wt}(\vec{x})} = i^{\left(\mathrm{wt}(\vec{x})^2\right)}. \tag{4.45}$$

Since $\mathrm{wt}(\vec{x})$ is even, we know that $\mathrm{wt}(\vec{x})^2$ is a multiple of 4. Thus, the left hand side of this is 1.

Using (3.4a) and (3.4b) repeatedly, we see that C is ± 1 if $\mathrm{wt}(\vec{x}) = 0 \pmod 4$, and $\pm i$ if $\mathrm{wt}(\vec{x}) = 2 \pmod 4$. But $C^2 = 1$ implies that $C = \pm 1$, and this can happen only if $\mathrm{wt}(\vec{x}) = 0 \pmod 4$. ☑

5 Codes, Again

5.1 Examples of Doubly-Even Codes

Since doubly even codes classify chromotopologies, it is useful to consider a few examples of such codes. For each N there is a trivial doubly even code $\{00\cdots 0\}$ with one element, which we call t_N; its generating set is the empty set. In addition, when $N = 4$, there is a code $\{0000, 1111\}$, called d_4. The generating set is $\{1111\}$. More generally, for every even $N \geq 4$, there is a doubly even code called d_N, of length N and with $\frac{N}{2} - 1$ generators, with generating set

$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,0 \cdots 0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0\,0 \cdots 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0 \cdots 0\,0\,0\,0\,0 \\ \vdots \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \cdots 0\,1\,1\,1\,1 \end{bmatrix}. \tag{5.1}$$

Note that this is a description of the generating set, so that the actual code has more codewords, including the null-vector and all those constructed by adding (bitwise, modulo 2) any number of these generators together. For example,

$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1 \end{bmatrix} \qquad \text{generates} \qquad \left\{ \begin{array}{rl} v_0 = & 0\,0\,0\,0\,0\,0\,0\,0 \\ v_1 = & 1\,1\,1\,1\,0\,0\,0\,0 \\ v_2 = & 0\,0\,1\,1\,1\,1\,0\,0 \\ v_3 = & 0\,0\,0\,0\,1\,1\,1\,1 \\ v_1 \boxplus v_2 = & 1\,1\,0\,0\,1\,1\,0\,0 \\ v_2 \boxplus v_3 = & 0\,0\,1\,1\,0\,0\,1\,1 \\ v_1 \boxplus v_3 = & 1\,1\,1\,1\,1\,1\,1\,1 \\ v_1 \boxplus v_2 \boxplus v_3 = & 1\,1\,0\,0\,0\,0\,1\,1 \end{array} \right\}. \tag{5.2}$$

Note that the same code, on the right-hand side of the display (5.2), is just as well generated by $\{v_1, v_2, (v_1 \boxplus v_3)\}$ and several other choices. For general N, the d_N code contains $2^{\frac{N}{2}-1}$ codewords.

When N is congruent to 7 or 8 modulo 8, there is an important doubly even code called e_N, the generating set of which is that of d_N (or $t_1 \oplus d_{N-1}$ when $N \equiv 7 \pmod 8$) augmented by an additional generator of the form $101010\cdots$. For instance,

$$e_7 : \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,0\,1\,0\,1 \end{bmatrix}, \qquad e_8 : \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1\,0 \end{bmatrix}, \tag{5.3}$$

and we then write:

$$e_{15}\ :\ \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \end{bmatrix}, \qquad e_{16}\ :\ \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0 \end{bmatrix}, \tag{5.4}$$

and so on.

These are famous codes: e_7 is known as the Hamming $[7, 3]$ code, and e_8 is the parity-extended Hamming code. Ref. [27], describes a so-called "Construction A", which determines a lattice as a subset of $\mathbb{Z}^N$ of all the points whose coordinates modulo 2 are in the code, and under this, we form the famous lattices e_7 and e_8. The points that are of closest distance to the origin form the root lattice for the Lie algebras E_7 and E_8, respectively. This correspondence is in fact more general [34], and may in particular also used to reconstruct the root lattices of the D_N Lie algebras.

Besides the trivial doubly even code t_N, $\{000\cdots0\}$, for any $N \equiv 0 \pmod 4$, there is an $[N, 1]$ doubly even code h_N consisting of $\{000\cdots0, 111\cdots1\}$, the generating set of which is $\{111\cdots1\}$.[4] Note that $h_4 = d_4$, but $h_N \subset d_N$ for $N = 8, 12, 16, \ldots$.

There are many other doubly even codes, and the number grows quickly as N becomes large; see Appendix C and Refs. [35,27].

5.2 Forgetting the Color of Edges, Permutation Equivalence, and R-Symmetry

It is also possible to permute the columns in a code. For instance, for e_7 we might swap the last two columns and obtain a generating set

$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1\,0\,1 \\ 1\,0\,1\,0\,1\,1\,0 \end{bmatrix}. \tag{5.5}$$

This is another doubly even code, and it is different from e_7 as given in Eq. (5.3). To verify this, one could write the 8 codewords in both cases and compare. More generally, any permutation of columns of a code will produce another code, which is sometimes the same code, sometimes not.

An example where a column-permutation results in precisely the same code again can be seen by taking the e_7 generating set (5.3), and swapping the first and third column, then swapping the second and fourth column. The result would be:

$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,0\,1\,0\,1 \end{bmatrix} \quad \longrightarrow \quad \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,0\,0\,1\,1\,0 \\ 1\,0\,1\,0\,1\,0\,1 \end{bmatrix}. \tag{5.6}$$

[4] This is the only code mentioned here not specifically named in Ref. [26].

The result does not look like the e_7 generating set, but it generates the same code. Indeed, replace the second generator with the sum of the first and the second generator, and we recover exactly the original generator set for e_7.

Yet all the so-obtained codes are in some sense similar, and we call codes related in this way *permutation-equivalent*. Note the distinction between code equality and their permutation equivalence, which is weaker than equality. It is convenient for the classification and naming of codes to give one name for the permutation equivalence class, and recognize the multiplicity of codes that the name represents.

Since the columns of a code correspond to the various Q_I, a permutation of the columns of the code corresponds to a permutation of the Q_I, *i.e.*, to an R-symmetry. For real N-extended supersymmetry, the group of R-symmetries is $O(N)$; the permutation equivalences describe the subgroup of this matrix group consisting of permutation matrices. Though this might suggest that the physically relevant question is permutation equivalence of codes, this is not necessarily so: It may well be possible to construct a theory with two types of supermultiplets, corresponding to two different but permutation-equivalent codes, coupled in a way that precludes rewriting the same theory in terms of only one type of supermultiplet. Although different in technical detail, the inextricable coupling of chiral and twisted-chiral supermultiplets discovered in Ref. [36] is a conceptual paradigm of this possibility.

The columns also correspond to the colors of the Adinkra, so permutation equivalence classes give rise to Adinkra topologies (without the edge colors). This raises the question: do permutation equivalence classes of doubly even codes classify connected Adinkra topologies? Certainly we have just described a map from the set of permutation equivalence class of doubly even codes to the set of connected Adinkra topologies. And certainly this map is surjective. But is it injective? That is, is it possible that two non-equivalent doubly even codes will give rise to the same Adinkra topology? The answer to this question is not clear, but luckily, in trying to classify Adinkras, we can leapfrog the issue of classifying Adinkra topologies and instead use the classification of Adinkra chromotopologies, where the issue is clear.

The set of column-permutations that do not change the (complete) code forms a group, called the automorphism group of the code, $\mathrm{Aut}(G)$. The number of codes permutation-equivalent to G is then $N!/|\mathrm{Aut}(G)|$, and is regarded the "mass" of the code. Below is a table listing the number of elements in $\mathrm{Aut}(G)$. Note that d_4, e_7 and e_8 do not fit the pattern for the other d_N or e_N.

It turns out to be possible to independently determine the total number of $[N, k]$ codes of various kinds, including those we need here. This total number must equal the sum of all "mass"-contributions, $N!/|\mathrm{Aut}(G_i)|$, of all $[N, k]$ codes, G_i. Such sum rules are called "mass formulae", and Gaborit [37] provides formulae to obtain all that we will need. These formulae are written by cases according to the congruence class of N modulo 8, and are given in Appendix C.

These numbers, even for moderate N such as $N = 11$, are intimidating. Nevertheless, in some cases, these are due to the many permutation-equivalent codes. Table 3 provides a listing of permutation equivalence classes for N up to 11, obtained by combining the results from Table 2 and accounting for permutations.

Beyond this, however, the number of permutation equivalence classes is still large: see Table 4

G	$\lvert\mathrm{Aut}(G)\rvert$
t_N	$N!$
d_4	24
d_{2m}, $m > 2$	$2^{m-1}m!$
e_7	168
e_8	1344
e_{8m-1}, $m > 1$	$2^{4m-1}(4m-1)!$
e_{8m}, $m > 1$	$2^{4m-1}(4m)!$
h_{4m}, $m > 1$	$(4m)!$

Table 2: Number of elements in $\mathrm{Aut}(G)$

and Appendix C for details. It is plainly impossible to list all the doubly even $[N, k]$-codes for $N \leq 32$ in journal publication.

N	$k=0$		$k=1$		$k=2$		$k=3$		$k=4$	
4	t_4	(1)	d_4	(1)						
5	t_5	(1)	$t_1 \oplus d_4$	(5)						
6	t_6	(1)	$t_2 \oplus d_4$	(15)	d_6	(15)				
7	t_7	(1)	$t_3 \oplus d_4$	(35)	$t_1 \oplus d_6$	(105)	e_7	(30)		
8	t_8	(1)	$t_4 \oplus d_4$	(70)	$t_2 \oplus d_6$	(420)	$t_1 \oplus e_7$	(240)	e_8	(30)
			h_8	(1)	$d_4 \oplus d_4$	(35)	d_8	(105)		
9	t_9	(1)	$t_5 \oplus d_4$	(126)	$t_3 \oplus d_6$	(1260)	$t_2 \oplus e_7$	(1080)	$t_1 \oplus e_8$	(270)
			$t_1 \oplus h_8$	(9)	$t_1 \oplus d_4 \oplus d_4$	(315)	$t_1 \oplus d_8$	(945)		
10	t_{10}	(1)	$t_6 \oplus d_4$	(210)	$t_4 \oplus d_6$	(3150)	$t_3 \oplus e_7$	(3600)	$t_2 \oplus e_8$	(1350)
			$t_2 \oplus h_8$	(45)	$t_6 * d_6$	(630)	$d_4 \oplus d_6$	(3150)	d_{10}	(945)
					$t_2 \oplus d_4 \oplus d_4$	(1575)	$t_2 \oplus d_8$	(4725)		
11	t_{11}	(1)	$t_7 \oplus d_4$	(330)	$t_5 \oplus d_6$	(6930)	$t_4 \oplus e_7$	(9900)	$t_3 \oplus e_8$	(4950)
			$t_3 \oplus h_8$	(165)	$t_1 \oplus t_6 * d_4$	(6930)	$t_1 \oplus d_4 \oplus d_6$	(34650)	$t_1 \oplus d_{10}$	(10395)
					$t_3 \oplus d_4 \oplus d_4$	(5775)	$t_3 \oplus d_8$	(17325)	$d_4 \oplus e_7$	(9900)
							$t_5 * d_6$	(13860)		

Table 3: A listing of permutation equivalence classes for N up to 11, with the number of codes in the permutation equivalence class given in in the parentheses. Here, $\oplus$ denotes a vector space direct sum, so that if $U \subset (\mathbb{Z}_2)^N$ and $V \subset (\mathbb{Z}_2)^M$, then $U \oplus V \subset (\mathbb{Z}_2)^N \oplus (\mathbb{Z}_2)^M \cong (\mathbb{Z}_2)^{N+M}$. The notation $t_M * C$ denotes the direct sum together with at least one additional "glue" codeword extending into the t_M summand, similar to how a e_N code is constructed from the corresponding d_N code.

5.3 Coset Enumerators and One-Hook Hanging Adinkras

Since we have concluded that any connected Adinkra chromotopology is a quotient of the N-cube by a doubly even $[N, k]$-code G, the vertices of the Adinkra (that is, the component fields of the supermultiplet) correspond to cosets of $\{0, 1\}^N$, thought of as $(\mathbb{Z}_2)^N$, by the subgroup G. This immediately implies that the Adinkra has 2^{N-k} nodes, where k is the dimension of G. Of these,

$N\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
4	1															
5	1															
6	1	1														
7	1	1	1													
8	2	2	2	1												
9	2	2	2	1												
10	2	3	3	2												
11	2	3	4	3												
12	3	5	7	7	2											
13	3	5	8	8	4											
14	3	7	12	14	9	4										
15	3	7	15	20	15	8	2									
16	4	10	23	38	36	23	9	2								
17	4	10	25	45	50	34	14	3								
18	4	13	34	72	94	79	35	9								
19	4	13	40	94	146	141	75	19								
20	5	17	57	158	295	353	231	84	10							
21	5	17	63	194	439	629	494	198	38							
22	5	21	83	298	812	1481	1465	740	187	25						
23	5	21	95	387	1287	2970	3811	2362	714	119	11					
24	6	27	129	607	2444	7287	12395	10048	3710	739	94	9				
25	6	27	141	755	3808	15177	35916	38049	16039	2973	309	22				
26	6	32	180	1114	6923	37455	128270	194626	103527	20206	1829	103				
27	6	32	202	1435	11320	86845	*	*	*	174809	13578	525				
28	7	39	263	2136	20812	*	*	*	*	*	*	7402	151			
29	7	39	287	2693	34233	*	*	*	*	*	*	*	1940			
30	7	46	359	3866	*	*	*	*	*	*	*	*	*	731		
31	7	46	400	4972	*	*	*	*	*	*	*	*	*	*	210	
32	8	55	506	*	*	*	*	*	*	*	*	*	*	*	*	85
N/k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**

Table 4: Number of distinct permutation classes of doubly even $[N, k]$ codes. The "*" entry indicates codes that are still being enumerated; see `http://www.rlmiller.org/de_codes/` for up-to-date results, including links to listings of the actual codes.

one half represents bosonic component fields, and the other half fermionic component fields in the corresponding supermultiplet. We thus have that the number of bosonic component fields, d_B, and the number of fermionic component fields, d_F, after taking this quotient satisfy

$$d_B \;=\; d_F \;=\; 2^{N-k-1}. \tag{5.7}$$

In Ref. [21], we described a notion of hanging a graph by a one or more sources. For instance, if we pick the vertex v_* and hang the graph by it, we let all the arrows on edges point from the vertices that are further away from v_* (as measured through the edge set) to vertices that are closer to v_*. Equivalently, for each vertex v we define the engineering degree of v to be

$$[v] = [v_*] - \frac{1}{2}\text{dist}(v, v_*), \tag{5.8}$$

where $\text{dist}(v, v_*)$ is the length of the shortest path from v to v_* in the edge set. Then the arrows are drawn in the direction of increasing engineering degree, upward.

For this, "one-hooked" Adinkra, we can examine how many component fields are in each engineering degree. By Eq. (5.8), this is equivalent to finding how many vertices are of a given distance

N	$\mathcal{N}$	$\min(d_B)$
4	1	4
8	2	8
12	3	64
16	4	128
20	5	1,024
24	6	2,048
28	7	16,384
32	8	32,768

Table 5: Minimal numbers for d_B and d_F

from the highest vertex, v_*. The corresponding notion in coding theory is the "coset weight enumerator" [26]. This is a polynomial of the form $\sum_\ell a_\ell\, x^\ell$ where a_ℓ is the number of cosets whose distance to the 0 coset is ℓ. Thus, the coset weight enumerator for a doubly even code can be used to find the number of component fields in each engineering dimension for a one-hooked Adinkra corresponding to that doubly even code.

The equation of (5.7) implies that in order to minimize the numbers of bosonic and fermionic fields in an Adinkra, one must maximize the value of k to $k = \varkappa(N)$. Thus the minimum number of bosons, $\min(d_B)$, and the minimum number of fermions, $\min(d_F)$, satisfy

$$\min(d_B) \;=\; \min(d_F) \;=\; 2^{N-\varkappa(N)-1}. \tag{5.9}$$

Since the ultimate application toward which all our efforts are aimed is a comprehensive understanding of the representation theory of spacetime supersymmetry in higher dimensions, it is perhaps useful to use this formula to gains some insights into the representations present in 4D, $\mathcal{N}$-extended supersymmetric theories (note: $\mathcal{N} := N/4$). For this purpose we note the results in Table 5, calculated using (5.9).

The lowest values are in perfect agreement with well known results. The value for $\mathcal{N} = 1$ corresponds to the chiral supermultiplet, the value for $\mathcal{N} = 2$ corresponds to the vector and tensor supermultiplets, and the value of $\mathcal{N} = 3, 4$ corresponds to the respective Weyl supermultiplets. Curiously enough and by very different arguments [38], the final result of $\min(d_B) = \min(d_F) = 32,768$ in this table appeared very early in our considerations of iso-spinning particles and "Garden Algebras" as the spectrum generating algebras of spacetime supersymmetry.

6 Toward a Classification of Adinkras and Supermultiplets

We have in this paper described the classification of chromotopologies of Adinkras: they are disjoint unions of connected chromotopologies, each of which is described by a doubly even binary linear code. The converse fact that every doubly even code can be used to form an Adinkra will be proved in Ref. [33]. This is done by taking an arbitrary doubly even code and constructing a supermultiplet whose Adinkra chromotopology comes from the code.

We have already addressed the issue of orienting the edges in Ref. [21]: given a chromotopology, consider all the ways to hang the vertices of the graph at various heights, subject to certain condi-

tions. In Ref. [21] we have also proven that all such height assignments can be obtained one from another through a sequence of 'vertex-raising/lowering'. In turn, the possible choices in dashing of edges, *i.e.*, assigning factors of -1 to the action of certain Q_I's on certain component fields (see Table 1), is closely related to the classification of representations of the Clifford algebras [39] and their iterated $\mathbb{Z}_2$-quotients, and is deferred to subsequent work.

As per Definition 3.2, a choice of a chromotopology together with a choice of orientations for all the edges and a choice of their dashedness fully specifies every Adinkra.

In light of all of the above, does this classify one-dimensional N-extended off-shell supermultiplets? Not quite. First, there is the question of whether every such supermultiplet comes from an Adinkra. Our investigations in this area indicate that there do indeed exist non-Adinkraic supermultiplets, but Adinkras turn out to be useful in their description. Second, we might ask whether two Adinkras may describe the same supermultiplet. This indeed occurs, and this issue will be settled in Ref. [33]. Suffice it to say here that, depending on the engineering dimensions of the component fields, distinct Adinkras usually correspond to distinct supermultiplets. Recall also that the number of possible such choices of engineering dimensions grows combinatorially with the number of component fields [21], which in turn grows exponentially with N.

We are now also in position to review a minor controversy from a more complete perspective. There is in the literature a work [17] the title of which leaves the impression of the existence of a prior complete classification of all off-shell representations of N-extended worldline supersymmetry without central charges. Ref. [23] then shows that only using the scheme described in [17], there exist counter-examples to this impression, which in turn caused two of the Authors of [17] to create a further 'refinement' [40] to their scheme.

Our current work shows that the problem of a complete classification of off-shell representations of N-extended worldline supersymmetry (and also Adinkras) includes the simpler problem of a complete classification of Adinkra chromotopologies. The latter problem is itself a formidable one as apparently the number of inequivalent chromotopologies grows combinatorially with N and k; see Table 4. The works of Refs. [17,40] include neither reference nor indication to this simpler problem of classifying chromotopologies or some equivalent thereof, and with it the emergence of the role played by doubly even binary linear error-correcting codes. Even with the proposed 'refinement' the works of Refs. [17,40] must then be regarded as leaving more work necessary for the presentation of a complete theory of the off-shell representations of N-extended worldline supersymmetry.

Finally, we wish to draw attention to the "degeneracy" in constructing even the minimal supermultiplets for various N, uncovered by the listing of permutation equivalence classes of doubly even binary linear codes in Table 4.

In Table 6, we list the dimensions of the minimal representations for $1 \leq N \leq 16$, and provide the doubly even binary linear codes which encode the projection of $I^N = \{0,1\}^N$ so as to obtain the corresponding Adinkra topologies. Starting with $N = 10$, there is more than one such permutation-inequivalent projection, and so more than one such minimal Adinkra topology. Each of these corresponds to at least one minimal supermultiplet, obtained by iteratively projecting an "unprojected" supermultiplet such as the Clifford supermultiplet, $\mathscr{M}^{=}_{I^N}$, defined in section 4.1.2, according to the provided permutation equivalence class of codes. "Unprojected" supermultiplets

$N=1$ (1|1)	$N=2$ (2|2)	$N=3$ (4|4)	$N=4$ (4|4)	$N=5$ (8|8)	$N=6$ (8|8)	$N=7$ (8|8)	$N=8$ (8|8)
			d_4 1111	$t_1 \oplus d_4$ 01111	d_6 001111 111100	e_7 0001111 0111100 1010101	e_8 00001111 00111100 11110000 01010101

$N=9$ (16|16)	$N=10$ (32|32)	$N=11$ (64|64)	$N=12$ (64|64)	$N=13$ (128|128)	$N=14$ (128|128)	$N=15$ (128|128)	$N=16$ (128|128)
$t_1 \oplus e_8$ 0 00001111 0 00111100 0 11110000 0 01010101	$t_2 \oplus e_8$ 00 00001111 00 00111100 00 11110000 00 01010101	$t_3 \oplus e_8$ 000 00001111 000 00111100 000 11110000 000 01010101	$d_4 \oplus e_8$ 0000 00001111 0000 00111100 0000 11110000 0000 01010101 1111 00000000	$t_1 \oplus d_4 \oplus e_8$ 00000 00001111 00000 00111100 00000 11110000 00000 01010101 01111 00000000	$d_6 \oplus e_8$ 000000 00001111 000000 00111100 000000 11110000 000000 01010101 001111 00000000 111100 00000000	$e_7 \oplus e_8$ 0000000 00001111 0000000 00111100 0000000 11110000 0000000 01010101 0001111 00000000 0111100 00000000 1010101 00000000	$e_8 \oplus e_8$ 00000000 00001111 00000000 00111100 00000000 11110000 00000000 01010101 00001111 00000000 00111100 00000000 11110000 00000000 01010101 00000000
	d_{10} 0000001111 0000111100 0011110000 1111000000	$t_1 \oplus d_{10}$ 0 0000001111 0 0000111100 0 0011110000 0 1111000000	d_{12} 000000001111 000000111100 000011110000 001111000000 111100000000	$t_1 \oplus d_{12}$ 0 000000001111 0 000000111100 0 000011110000 0 001111000000 0 111100000000	d_{14} 00000000001111 00000000111100 00000011110000 00001111000000 00111100000000 11110000000000	e_{15} 000000000001111 000000000111100 000000011110000 000001111000000 000111100000000 011110000000000 101010101010101	e_{16} 0000000000001111 0000000000111100 0000000011110000 0000001111000000 0000111100000000 0011110000000000 1111000000000000 1010101010101010
		$d_4 \oplus e_7$ 0000000 1111 0001111 0000 0111100 0000 1010101 0000		$d_6 \oplus e_7$ 0000000 001111 0000000 111100 0001111 000000 0111100 000000 1010101 000000	$e_7 \oplus e_7$ 0000000 0001111 0000000 0111100 0000000 1010101 0011110 0000000 1111000 0000000 1010101 0000000		
				e_{13} 0000000001111 0000000111100 0000011110000 0001111000000 1110101010101	e_{14} 00000000001111 00000000111100 00000011110000 00001111000000 00111100000000 11010101010101		

Table 6: Minimal supermultiplets of the indicated N-extended supersymmetry, with the stated number of (bosonic|fermionic) component fields and a generator set for the code, C, specifying the chromotopology as the $\{0,1\}^N/C$ quotient.

other than $\mathscr{M}^{=}_{I^N}$ can, in general, be quotiented only by a subcode of the listed codes, depending on the distribution of the component fields' engineering dimensions.

Notice that Table 6 details the cases represented only by the $N \leq 16$ right-most entries in Table 4, and the corresponding entries in Table 5. The remaining, $16 < N \leq 32$ right-most entries in Table 4 also correspond to topologically distinct minimal supermultiplets, but their sheer number prevents us from displaying them in the manner of Table 6. This clearly illustrates that this "degeneracy" amongst even the minimal supermultiplets grows extremely fast with N.

In turn, this provides a surprising wealth of building blocks for models with N-extended worldline supersymmetry as N is increased towards $N = 32$.

Mathematics is not a careful march down a well-cleared highway,
but a journey into a strange wilderness,
where explorers often get lost.
– W.S. Anglin

Acknowledgments: The research of S.J.G. is supported in part by the National Science Foundation Grant PHY-0354401. T.H. is indebted to the generous support of the Department of Energy through the grant DE-FG02-94ER-40854. The work described in Appendix C uses computer facilities supported by the National Science Foundation Grant No. DMS-0555776.

A Deferred Details on Supersymmetry and Adinkras

A.1 Real Coefficients

Real supermultiplets, $\mathscr{M} = (F_1(\tau), \cdots, F_m(\tau))$ consist of real component fields: $\big(F_A(\tau)\big)^\dagger = F_A(\tau)$, and supersymmetry is assumed to preserve this condition. In this section, we will show that this condition implies that the coefficients c in Eq. (3.4a) and Eq. (3.4b) are real.

We use the convention whereby $(XY)^\dagger = Y^\dagger X^\dagger$, regardless whether X and Y are bosonic (commuting) or fermionic (anticommuting) objects, as is standard in the physics literature.

If $\phi_A(\tau)$ is real, then its supersymmetry transform must also be real. Applying this to Eq. (3.4a) results in

$$\delta_Q(\epsilon)\,\phi_A(\tau) = -i\epsilon^I\, c\,\partial_\tau^{[\phi_A]+\frac{1}{2}-[\psi_B]}\psi_B(\tau). \tag{A.1}$$

Thus

$$\Big(\delta_Q(\epsilon)\,\phi_A(\tau)\Big)^\dagger \;=\; i\,\partial_\tau^{[\phi_A]+\frac{1}{2}-[\psi_B]}\psi_B^\dagger(\tau)\,c^*\,\epsilon^I \;=\; -\,i\epsilon^I c^*\,\partial_\tau^{[\phi_A]+\frac{1}{2}-[\psi_B]}\psi_B(\tau). \tag{A.2}$$

Comparing the right-hand sides of Eqs. (A.1) and (A.2), we find that

$$c^* = c. \tag{A.3}$$

Thus the coefficients c are real.

A.2 The Proof that Scaling Factors $c = \pm 1$ are the Only Ones Necessary

We now will show that the coefficients in an adinkraic supermultiplet can be chosen to be $c = \pm 1$ via a rescaling of fields by real numbers.

Proposition A.1 *Suppose we have an adinkraic supermultiplet, with $F_1(\tau), \cdots, F_{2m}(\tau)$ for component fields. There is a real rescaling of these component fields so that each non-zero coefficient c in Eqs. (3.4a) and (3.4b) is equal to 1 or -1.*

Proof: We suppose we have component fields $F_1(\tau), \cdots, F_{2m}(\tau)$,[5] and supersymmetry generators Q_I so that the supersymmetry transformation rules are all of the form

$$Q_I F_A = c\,\partial_\tau^\lambda F_B \tag{A.4}$$

where λ is either 0 or 1, and F_B is another component field, and c is a complex number (real if F_A is bosonic, pure imaginary if F_A is fermionic).

We can draw an Adinkra-like graph for this supermultiplet: we again create vertices $v_1, \cdots, v_{2m}$ to correspond to the component fields $F_1, \cdots, F_{2m}$—white for bosonic fields and black for fermionic fields. For each supersymmetry transformation like the one above, we draw an edge from the vertex v_A to v_B. The arrow goes from v_A to v_B if $\lambda = 0$, and the reverse if $\lambda = 1$. As before, this is consistent with the corresponding equation for $Q_I F_B$. Instead of choosing between a dashed or

[5] See footnote *3* in the proof of Theorem 4.1 about the use of this notation.

solid line, we label the edge by c if F_A is bosonic and i/c if it is fermionic. It is easy to check that this is consistent with the corresponding equation for $Q_I F_B$.

We can work on each connected component of this Adinkra-like graph separately. So for the following we assume our Adinkra-like graph is connected.

We wish to rescale the various component fields so that c has absolute value 1. A naive approach would be to simply rescale F_B by c, so that c disappears in Eq. (A.4). But remember that there are $2mN$ of these equations, and only $2m$ component fields. More visually, imagine the Adinkra-like graph with separate c-labels on each edge. We can imagine starting at one vertex, then going along an edge to the next vertex, rescaling the corresponding field by whatever it takes to make $c = 1$ on that edge. The trouble is that when we come back to a vertex we already saw, we are no longer free to rescale that field without messing up other c's. But as we will see, it turns out that the supersymmetry algebra guarantees that when we return to a previously-seen vertex, the edge will have $|c| = 1$.

We first begin by removing all the loops, using the following standard procedure from graph theory: if we choose an edge, and erase it from the graph, the graph may either separate or stay connected. We first find an edge so that erasing it keeps the graph connected. By successively finding such edges and erasing them, we obtain a minimal graph so that this graph is still connected. This resulting graph is a tree, so that for every pair of vertices v_i and v_j, there exists a unique, non-backtracking path from v_i to v_j in this tree (were it not unique, we could remove another edge). This tree is called a *spanning tree* for the original graph. The spanning tree for a graph is not unique, but we select one. See Figure 1.

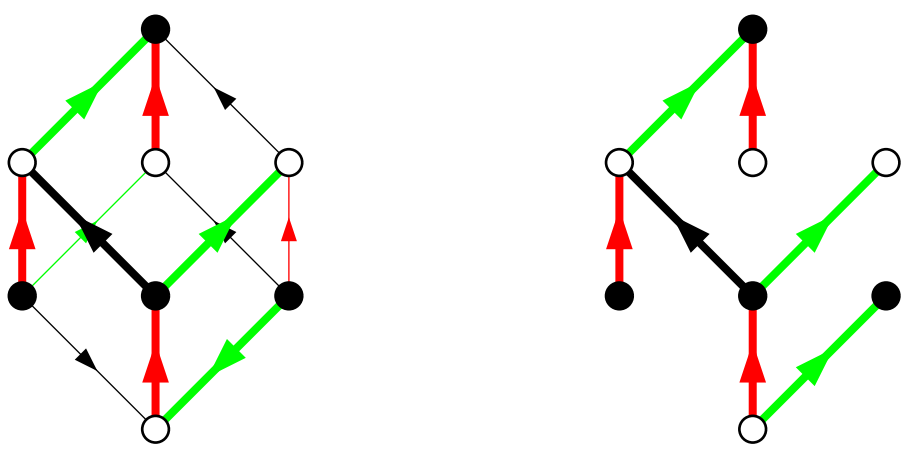


Figure 1: On the left is an $N = 3$ cubical Adinkra. The dashedness of the lines are suppressed for clarity. Some edges (shown with thin lines) can be deleted, until what is left is a tree, called the *spanning tree* (shown at right).

Now pick any vertex in our graph, denote it v_* and the corresponding component field $F_*(\tau)$. For simplicity of exposition we can assume v_* is bosonic. If v_i is any other vertex, there is a sequence

of edges in the tree connecting v_* with v_i, corresponding to a sequence of Q_I's, say $Q_{I_1}, \cdots, Q_{I_n}$. We then apply Eq. (A.4) iteratively to $F_*(\tau)$, obtaining

$$Q_{I_n} \cdots Q_{I_1} F_*(\tau) = C_i \, \partial_\tau^\lambda F_i(\tau) \tag{A.5}$$

for some non-negative integer λ (recording the number of arrows that point in the opposite direction of the path), and some non-zero coefficient C_i. This coefficient C_i is real if $n \equiv 0, 1 \pmod 4$, and pure imaginary if $n \equiv 2, 3 \pmod 4$. It is then possible to rescale $F_i(\tau)$ as

$$\tilde{F}_i(\tau) := |C_i| \, F_i(\tau), \tag{A.6}$$

and using $\tilde{F}_i(\tau)$ instead of $F_i(\tau)$ in the description of the supermultiplet. This rescaling is by a real number. Do so for every $v_i \neq v_*$. We have now done all the rescaling of the fields that we will do. The result is a new Adinkra-like graph, where each C_i has absolute value 1.

If instead we want to go from v_i back to v_* in the tree, we can do the supersymmetry transformations in reverse:

$$Q_{I_1} \cdots Q_{I_n} F_i = \frac{1}{C_i} \, \partial_\tau^{n-\lambda} F_*. \tag{A.7}$$

Now suppose we choose two vertices v_i and v_j of the graph. We construct a path P from v_j to v_i by first taking the non-retracing path in the spanning tree from v_j to v_*, followed by the non-retracing path in the spanning tree from v_* to v_i. See Figure 2. Note that P might be partially retracing, if the last few edges of the first path is retraced backward by a beginning segment of the second path. This retracing is immaterial for us.

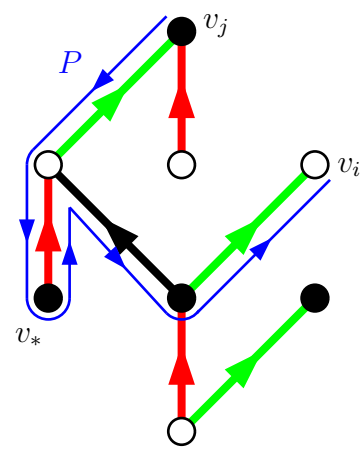


Figure 2: Using the same example as in Figure 1, suppose we are considering the vertices v_i and v_j as seen in the figure. The blue path shows a path P in the spanning tree going from v_j to v_*, followed by a path in the spanning tree going from v_* to v_i. One edge is retraced, but this is irrelevant for our purposes. The sequence of colors (black = 1, red = 2, green = 3) along P indicate that we should consider $Q_3 Q_1 Q_2 Q_2 Q_3 v_j$ which will be a constant multiple of $\partial_\tau^3 v_i$. Because of our rescaling in Eq. (A.6), we know this constant has absolute value 1.

If the path P involves the sequence $Q_{J_1} \cdots Q_{J_k}$ of supersymmetry generators, we then use Eq. (A.5) and Eq. (A.7) to get

$$Q_{J_k} \cdots Q_{J_1} F_j = \frac{C_i}{C_j} \, \partial_\tau^\mu F_i, \tag{A.8}$$

where μ is some non-negative integer, measuring the number of arrows pointing against the path P. Note that C_i/C_j has absolute value 1.

In particular, if we consider any edge of our Adinkra that is in the spanning tree, its label c now has absolute value 1.

We now examine an edge of the Adinkra that is not in the spanning tree. Suppose it connects a bosonic vertex v_i to a fermionic vertex v_j, *i.e.*, there is a Q_I so that

$$Q_I v_i = c\,\partial_\tau^\lambda v_j \tag{A.9}$$

for some real c and $\lambda = 0$ or 1. Now take the path P in the spanning tree described above, namely, the one that goes from v_j to v_* then to v_i. This corresponds to a sequence of supersymmetry generators $Q_{I_1}, \cdots, Q_{I_n}$. We now close the path P into a closed loop, P' by adding to P the new edge going back from v_i to v_j.

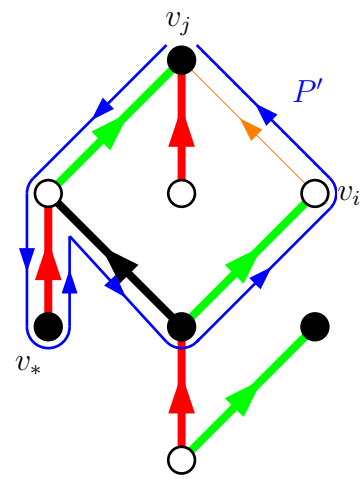


Figure 3: Suppose we consider the orange edge from v_i to v_j, which is in the original Adinkra but not in the spanning tree. We draw that path P from v_j to v_i in the spanning tree as in Figure 2, and then tack on the orange edge. The resulting path P', shown in blue, is a loop from v_j to itself.

Using Eq. (A.9) together with Eq. (A.8) we get, for some $r \in \mathbb{N}$,

$$Q_I Q_{I_n} \cdots Q_{I_1} F_j(\tau) = c(-i)^r \frac{C_i}{C_j}\,\partial_\tau^{\lambda+\mu} F_j. \tag{A.10}$$

As this is a closed loop, we can apply this equation to itself, and get

$$\left(Q_I Q_{I_n} \cdots Q_{I_1} \right)^2 F_j(\tau) = (-1)^r c^2 \left(\frac{C_i}{C_j} \right)^2 \partial_\tau^{2(\lambda+\mu)} F_j(\tau). \tag{A.11}$$

Using Eq. (3.1) we can anti-commute the Q_I's past each other, contracting the repeated Q_I's, until we get

$$\pm (i\partial_\tau)^{n+1} F_j(\tau) = (-1)^r c^2 \left(\frac{C_i}{C_j} \right)^2 \partial_\tau^{2(\lambda+\mu)} F_j(\tau). \tag{A.12}$$

The $\pm$ sign is determined by n and how many repetitions there are in the Q_I sequence. Note that $n+1$ is even, since each edge connects vertices of opposite statistics, and a sequence of $n+1$ of

these edges go from v_j to itself. Thus, the left hand side of Eq. (A.12) is actually real. We can then match coefficients in Eq. (A.12) to get

$$c^2 = \pm(-1)^{r+(n+1)/2}\frac{C_j^2}{C_i^2} = \pm 1. \tag{A.13}$$

From this we see that c must have complex absolute value 1. Since c is real, $c = \pm 1$. Furthermore, we also note from Eq. (A.12) that $2(\lambda + \mu) = n + 1$, indicating that the loop P' involved precisely $(n + 1)/2$ arrows going against the path, and thus, $(n + 1)/2$ arrows going along the path. This implies that the absence of central charge excludes Escheric loops [13]. ☑

This also proves that Adinkraic supermultiplets with no central charge are *engineerable* [21]: it is possible to assign engineering dimensions to all component fields, consistently with the action Eqs. (3.4) of the supersymmetry algebra (3.1), and without having to introduce parameters of nonzero engineering dimension in either of these.

B Complex Adinkras

Everything in the paper up to now has dealt with supermultiplets thinking of each field as real. If we wish to represent complex supermultiplets, we can introduce a kind of Adinkra that is analogous to the Adinkras we have been considering up to now, only using vertices to describe complex fields instead of real fields. Many of the same ideas continue to hold, except that instead of the coefficients c being real, they might also be complex.

If an adinkraic supermultiplet is complex, then the proof of Proposition A.1 in Appendix A.2 can be modified in the following way: when the fields are rescaled by a real number to make $|C_i| = 1$, we can actually rescale the fields by a complex number to make each $C_i = 1$. Then, Eq. (A.13) shows that the $c^2 = \pm 1$, so that c could be ± 1, or it could also be $\pm i$. So instead of edges being either solid or dashed, there should be two other ways to decorate the edge to denote the additional cases of i and $-i$.

For examples of complex Adinkras, we can take the cubical Adinkras, and simply reinterpret the vertices as representing complex component fields, instead of real ones. Thus, for $N = 1$, there is the base Adinkra and the Klein-flipped base Adinkra, just as in the real case. Note that these have edges labelled only by ± 1, so it is no surprise that there are other complex Adinkras that are not described in this way.

As in the real case, connected Adinkras can be obtained by taking the quotient of a cube by a binary linear block code (the same proof applies), but this time the condition that the code is doubly even is replaced by the condition that the code is even and self-orthogonal. That is, codewords need not have weights that are multiple of 4; rather, we require simply that their weights be even. In this case, we must also impose self-orthogonality, since unlike doubly even codes, even codes are not always self-orthogonal.

For instance, the $N = 2$ exterior Adinkra splits. To be specific, the $N = 2$ exterior supermultiplet involves two bosons ϕ_1, ϕ_2 and two fermions ψ_1, ψ_2.

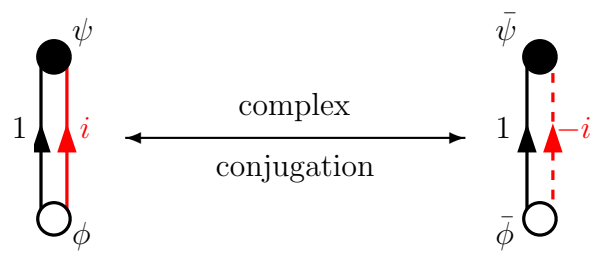
$$Q_1\phi_1 = \quad \psi_1, \qquad\qquad Q_2\phi_1 = -\psi_2, \tag{B.1}$$

$$Q_1\phi_2 = \quad \psi_2, \qquad\qquad Q_2\phi_2 = \quad \psi_1, \tag{B.2}$$
$$Q_1\psi_1 = -i\,\partial_\tau\phi_1, \qquad\qquad Q_2\psi_1 = -i\,\partial_\tau\phi_2, \tag{B.3}$$
$$Q_1\psi_2 = -i\,\partial_\tau\phi_2, \qquad\qquad Q_2\psi_2 = \quad i\,\partial_\tau\phi_1. \tag{B.4}$$

To project out, we define $\phi = \phi_1 + i\phi_2$, $\bar{\phi} := \phi_1 - i\phi_2$, $\psi = \psi_1 + i\psi_2$, and $\bar{\psi} := \psi_1 - i\psi_2$. The supermultiplet $(\phi_1, \phi_2|\psi_1, \psi_2)$ may then be rewritten as a complex supermultiplet, $(\phi|\psi)$, and its conjugate, $(\bar{\phi}|\bar{\psi})$, with the supersymmetry transformations:

$$Q_1\phi = \quad \psi, \qquad\qquad Q_2\phi = \quad i\psi, \tag{B.5}$$
$$Q_1\psi = -i\,\partial_\tau\phi, \qquad\qquad Q_2\psi = -\partial_\tau\phi; \tag{B.6}$$
$$Q_1\bar{\phi} = \quad \bar{\psi}, \qquad\qquad Q_2\bar{\phi} = -i\bar{\psi}, \tag{B.7}$$
$$Q_1\bar{\psi} = -i\,\partial_\tau\bar{\phi}, \qquad\qquad Q_2\bar{\psi} = \quad \partial_\tau\phi. \tag{B.8}$$

The (complex) $\mathbb{C}$-Adinkras for $(\phi|\psi)$ and for $(\bar{\phi}|\bar{\psi})$ exhibit a relationship between the two complex supermultiplets:



(B.9)

The two disagree only on the values of c. The unlabeled topologies are the same, with a double edge between two vertices.

We note that one of these two complex supermultiplets (B.9) can be regarded as the "chiral" supermultiplet, while the other is its conjugate, the "anti-chiral" supermultiplet. Also, $Q_2 \neq \pm i\,Q_1$; we note that $Q_2 = +iQ_1$ when acting on $\phi(\tau)$ and $\bar{\psi}(\tau)$, but $Q_2 = -iQ_1$ when acting upon $\bar{\phi}(\tau)$ and $\psi(\tau)$. Thus, Q_1 and Q_2 really are independent.

C Computing Doubly Even Codes

For some classification problems, there are theorems that answer the question once and for all. For example, the theorem on the classification of finitely generated abelian groups describes the isomorphism class of such a group via a simple sequence of integers. In most cases in combinatorics, one cannot hope for such a theorem. Instead one must settle for an exhaustive enumeration up to a certain size. It turns out that there are more than 1.1 trillion doubly even, binary $[32, k]$ codes, which without compression will take some number of terabytes to store. Using methods including those described in this appendix, we have already computed over 60,000 of these codes.

The first section of this appendix gives the formula describing the number of codes we are interested in and a lower bound for the number of permutation-isomorphism classes of codes. The next section explains the computations necessary for computing the automorphism group and canonical representative of a binary code, which are critical for the exhaustive enumeration. In the third

section, some optimizations for the particular situation at hand are discussed, which are used in an implementation written specifically for this purpose. The fourth section describes the overarching algorithm designed to produce the intended enumeration.

The methods used to tackle this problem, namely partition refinement and canonical augmentation, were originally developed by McKay [41,42], who describes an effective method for computing the automorphism group of a graph and for generating a unique representative for each isomorphism class. These methods were adapted by Leon in Ref. [43] to tackle similar problems, including several group theory questions (which are graph isomorphism complete) as well as the problem at hand, namely computing the automorphism group of a linear code. McKay and Leon have both written optimized software packages implementing these methods. The first software package, which remains the standard program for determining graph isomorphism, is called `nauty`, and is available online through McKay's website [44]. The second is now licensed under the GPL, thanks to the efforts of David Joyner and Vera Pless, and is available as a part of GUAVA [45], a GAP package for coding theory. GPL implementations for the cases of graphs of any size and binary codes of length up to 32 (or 64 on many machines) can be found in Sage [46] (`sage.graphs.graph_isom` and `sage.coding.binary_code`). For more details about the latter, written specifically for the classification of Adinkra topologies, the Reader can consult Section C.3. The second and fourth sections are essentially a summary of the techniques described in Refs. [41] and [42], respectively, which provide the full story. The Reader should also be aware of the improvements to this algorithm in the sparse case in Ref. [47].

C.1 The Mass Formula

The following formula, due to Gaborit [37], gives the number $\sigma(N,k)$ of distinct doubly even binary $[N,k]$ codes, depending on the residue of N modulo 8:

$$\sigma(N,k) = \begin{cases} \displaystyle\prod_{i=0}^{k-1} \frac{2^{N-2i-2} + 2^{\lfloor \frac{N}{2} \rfloor - i - 1} - 1}{2^{i+1}-1}, & \text{if } N \equiv 1,7 \pmod 8, \\ \displaystyle\prod_{i=0}^{k-1} \frac{\left(2^{\frac{N}{2}-i-1}+1\right)\left(2^{\frac{N}{2}-i-1}-1\right)}{2^{i+1}-1}, & \text{if } N \equiv 2,6 \pmod 8, \\ \displaystyle\prod_{i=0}^{k-1} \frac{2^{N-2i-2} - 2^{\lfloor \frac{N}{2} \rfloor - i - 1} - 1}{2^{i+1}-1}, & \text{if } N \equiv 3,5 \pmod 8, \\ \displaystyle\prod_{i=0}^{k-2} \frac{2^{N-2i-2} + 2^{\frac{N}{2}-i-1} - 2}{2^{i+1}-1} \cdot \varpi^{+}(N,k), & \text{if } N \equiv 0 \pmod 8, \\ \displaystyle\prod_{i=0}^{k-2} \frac{2^{N-2i-2} + 2^{\frac{N}{2}-i-1} - 2}{2^{i+1}-1} \cdot \varpi^{-}(N,k), & \text{if } N \equiv 4 \pmod 8, \end{cases} \tag{C.1}$$

$$\varpi^{\pm}(N,k) := \frac{1}{2^{k-1}} + \frac{2^{N-2k} \pm 2^{\frac{N}{2}-k} - 2}{2^k - 1}\,, \tag{C.2}$$

We are interested in those codes with $N \leq 32$. According to the mass formula above, we find the most such codes when $N = 32$ and $k = 10$. In this case,

$$\begin{aligned}\sigma(32,10) &= 162,953,548,221,364,911,292,708,847,668,107,902,745,573,601,875 \\ &\approx 1.6 \times 10^{47}.\end{aligned} \tag{C.3}$$

However, we are interested not in the codes themselves, but rather their equivalence classes under permutations of the columns. To establish a lower bound on the number of equivalence classes, suppose that every such code has a trivial automorphism group and thus its orbit is as large as possible, namely 32!. Hence there are at least

$$\left\lceil \frac{\sigma(32,10)}{32!} \right\rceil = 619,287,158,132 \tag{C.4}$$

distinct classes of $[32, 10]$ codes. If we carry out a similar computation for each case of interest, we find that there are at least

$$\sum_{k=0}^{\varkappa(32)} \left\lceil \frac{\sigma(32,k)}{32!} \right\rceil = 1,117,005,776,858 \;\approx\; 1.1 \times 10^{12} \tag{C.5}$$

distinct classes (note that for smaller N, we simply add zero columns to obtain an equivalent code of degree 32). Since the automorphism group of a generic code is small, this lower bound is a reasonable estimate for the actual number of codes.

Regardless of the speed at which an individual processor can generate codes, the procedure for generating the codes can be run in many parallel processes. We can not only divide by the rate at which we can generate codes per processor, but we can also divide by the number of processors working on the job. This is what makes the computation feasible; speed increases almost linearly as a function of the number of processors.

C.2 Computing the Automorphism Group

Computing the automorphism group of a binary code is closely related to the graph isomorphism problem. Given a binary block code $C \subset \{0,1\}^N$, define a bipartite graph $G(C)$, with the vertices partitioned into "left" and "right", as follows: The words of the code are the left vertices, and the set $\{1, 2, ..., N\}$ forms the right set of vertices. Given a word w on the left, and a number j on the right, there is an edge between the two if and only if w has a 1 in the jth place. Consider the automorphism group $\mathrm{Aut}(G(C))$ of the graph, but allow only those permutations that map left vertices to left vertices and right vertices to right vertices. This gives a subgroup $\mathrm{Aut}(G(C))_b$ which is isomorphic in a canonical way to the permutation automorphism group of the code, simply by considering each permutation's action on the right set of vertices, which is identified with the set of columns of the code. Our approach is to think of the codes in terms of their corresponding bipartite graphs, with additional structure coming from the linear, self-orthogonal code. The algorithm described in this section computes the automorphism group of the code, as well as a canonical representative of the code, which is an arbitrary but fixed representative of the isomorphism class.

Suppose G is a graph, and Π is a partition of the vertices $V(G)$. The partition Π is called an *equitable partition* if for every pair of cells C_1, C_2 in Π, the number of edges $\{u, v\}$ such that

$v \in C_2$ is constant as u ranges over vertices in C_1. By considering the orbits of vertices of the graph under any subgroup of the automorphism group, one obtains a partition which is always equitable. However, the converse is not always true; there are equitable partitions which do not arise in this way. One partition Π_1 is *coarser* than another partition Π_2 (or Π_2 is *finer* than Π_1) if every cell of Π_2 is a subset of a cell of Π_1. The *discrete partition* is the one where every cell is of size one, and the *unit partition* is the one where there is only one cell.

Given a graph G and a partition Π of $V(G)$, denote by $E_G(\Pi)$ the coarsest equitable partition of G which is finer than Π. In particular, if Π is equitable with respect to G, then $E_G(\Pi) = \Pi$. The algorithm described in this section takes as input a graph G and a partition Π_0 of $V(G)$, and it returns the subgroup $\mathrm{Aut}(G)_{\Pi_0}$ of the automorphism group consisting of permutations that respect Π_0, *i.e.*, that do not carry any vertex of one cell of Π_0 into another. From here on, consider all partitions to be ordered, so that for example $(\{1,2,3\},\{4,5\}) \neq (\{4,5\},\{1,2,3\})$. In particular, a discrete ordered partition is simply an ordering on the vertices of G. For $v \in V(G)$, if $\Pi = (C_1, ..., C_r)$ and $v \in C_i$, define $R(G,\Pi,v) = E_G(\Pi')$, where $\Pi' = (C_1, ..., C_{i-1}, \{v\}, C_{i-1} \setminus \{v\}, C_{i+1}, ..., C_r)$. This defines $R(G,\Pi,v)$ up to reordering of the cells of the partition. For full details, see Ref. [41], Algorithm 2.5, which defines the ordering.

Define a rooted tree $T = T(G,\Pi_0)$ consisting of equitable partitions of G finer than Π_0 as follows: The root of T is the partition $E_G(\Pi_0)$, and for any node Π of T, its children are the partitions $R(G,\Pi,v)$ for which v is not yet in a singleton cell of Π. The group $\mathrm{Aut}(G)_{\Pi_0}$ acts on the tree T by taking a sequence of nested partitions $\Pi_0, ..., \Pi_k$ to the resulting nested sequence of partitions by letting $\mathrm{Aut}(G)_{\Pi_0}$ act on the elements of the cells; since this sequence is defined by a sequence $v_1, ..., v_k$, the sequence $\gamma(v_1), ..., \gamma(v_k)$ defines the image under $\gamma \in \mathrm{Aut}(G)_{\Pi_0}$. Further, the subgroup of permutations respecting a partition Π acts on any subtree of T rooted at Π. Because this action is faithful, the structure of T can be used to calculate a set of generators for $\mathrm{Aut}(G)_{\Pi_0}$.

The algorithm itself is a backtrack algorithm that successively refines the partitions to explore the tree T. Any leaf of the tree is a discrete ordered partition $(\{v_1\},\{v_2\},...,\{v_n\})$, which defines an ordering of the vertices of G. Given two leaves of the tree T, one has two orderings $v_1, ..., v_n$ and $v'_1, ..., v'_n$, which we think of as a permutation defined by $v_i \mapsto v'_i$. This is the means by which the algorithm finds automorphisms, and it uses the presence of automorphisms to deduce when different parts of the tree are equivalent. At this point the algorithm backtracks towards the root until there is a new part of the tree to explore which is not yet known to be equivalent to a part of the tree already traversed. In practice, the part of the tree traversed is much smaller than the entire tree, and once one backtracks off of the root, one has a set of generators for $\mathrm{Aut}(G)_{\Pi_0}$. Invariants and orderings can also be used to find a leaf of the tree T which is maximal amongst the nodes with largest invariants, which is uniquely defined independently of reordering the inputs. This leaf is a "canonical label" for the pair (G,Π_0). In particular, if $\gamma \in \mathrm{Aut}(G)_{\Pi_0}$, then the canonical label for (G^γ, Π_0^γ) is the same as that of (G,Π_0). For details, the Reader is once again directed to the theory in Ref. [41] and the code in Ref. [46].

The outcome is an algorithm to efficiently compute the group $\mathrm{Aut}(C) = \mathrm{Aut}(G(C))_b$ (here "=" means canonically isomorphic) as well as a unique representative of each isomorphism class which we will denote $c(C)$, both of which will be used in the algorithm to generate all the permutation classes of doubly even codes. Also note that by the use of the bipartite graph construction, not

only is $c(C)$ a binary code, but it also contains an ordering of its words. In practice, $c(C)$ will be output as an actual generator matrix.

C.3 New Software for Special Circumstances

It is a curious coincidence that the limit on N for this application is 32. This implies that the size of the words in a code are at most the size of the machine words on a 32-bit system. The analogy goes further: the operation of adding two vectors in a code becomes the single clock tick of taking an `XOR` on the machine words representing them.

When analyzing codes in terms of their corresponding graphs, we can take advantage of the linear orthogonal structure of the codes to optimize standard graph algorithms. For example, given an equitable partition Π of the bipartite graph $G(C)$, the set of words appearing in their own singleton cells of Π is closed under binary addition. This is immediate from the definition of equitable partition and of binary addition, and this fact greatly reduces the expected height of the tree to be searched in computing $c(C)$. Further, every permutation acts linearly, so to check if some permutation is indeed an automorphism, we need only check it on a basis. It is also possible to use the linear structure to derive variants on the refinement procedure, depending on the size and type of binary codes being generated.

In the generation algorithm, since we are interested only in self-orthogonal codes, we can use more linear algebra to efficiently enumerate the possible children of a code. We start by writing the code in standard form, $C = [I_k|C']$, and then extend the basis given by the rows to a basis for $C^\perp$. This basis can be taken so that the first k positions of each additional vector are zero, and we can also shuffle the pivot columns to the front, so that we obtain this basis as the rows of a matrix of the form

$$\left[\begin{array}{c|c|c} I_k & \multicolumn{2}{c}{C''} \\ \hline 0 & I_{N-2k} & * \end{array}\right],$$

where C'' is a rearrangement of the columns of C'. The rowspan of the bottom part of the matrix, $\begin{bmatrix} 0 & I_{N-2k} & * \end{bmatrix}$, then forms a set of unique coset representatives of the original code. Furthermore, (2.2) implies the following:

Corollary C.1 *Suppose C is a binary code spanned by $\{v_1, \cdots, v_k\}$, and that* $\mathrm{wt}(v_i) \equiv 0 \pmod 4$ *for each $i = 1, \ldots, k$. If either of the following hold for all $i \neq j$, then C is doubly even:*

a) $\langle v_i, v_j \rangle = 0$,

b) $\mathrm{wt}(v_i + v_j) \equiv 0 \pmod 4$.

Proof: These results are special to binary codes, since every vector in C is the sum of distinct vectors in $\{v_1, \cdots, v_k\}$. Equation (2.2) implies that both conditions are equivalent, so we use $b)$. Suppose $x, y, z \in C$ are such that

$$\mathrm{wt}(x) \equiv \mathrm{wt}(y) \equiv \mathrm{wt}(z) \equiv \mathrm{wt}(x+y) \equiv \mathrm{wt}(x+z) \equiv \mathrm{wt}(y+z) \equiv 0 \pmod 4.$$

Then again by (2.2),

$$\mathrm{wt}(x+y+z) \equiv \mathrm{wt}(x+y) + \mathrm{wt}(z) - 2\langle x+y, z \rangle$$

$$
\begin{aligned}
&\equiv -2\langle x,z\rangle - 2\langle y,z\rangle \\
&\equiv -2\,\mathrm{wt}(x) - 2\,\mathrm{wt}(z) + 2\,\mathrm{wt}(x+z) \\
&\quad -2\,\mathrm{wt}(y) - 2\,\mathrm{wt}(z) + 2\,\mathrm{wt}(y+z) \\
&\equiv 0 \pmod 4.
\end{aligned}
$$

By induction on the number of basis elements in a linear combination, C is doubly even. ☑

Going back to the situation at hand, this means that we need only examine the doubly even vectors in the rowspan of $\begin{bmatrix} 0 & I_{N-2k} & * \end{bmatrix}$, since the code we already have is doubly even and self-orthogonal, and any vector to be considered is already orthogonal to the code. These vectors are in one-to-one correspondence with the set of doubly even codes of one dimension higher containing (our rearrangement of) C . We can do even better by starting with the even subcode of $\begin{bmatrix} 0 & I_{N-2k} & * \end{bmatrix}$, since the sum of two even words is even.

C.4 Exhaustively Generating the Codes

Here we use another algorithm developed by McKay [42], called canonical augmentation. The ingredients for this are a question like the one we are considering, a canonical labeling function as described, a way of computing the automorphism group of a code, and a hereditary structure on the objects to be generated.

We obtain a hereditary structure by thinking of a certain code's *children* as the set of codes constructible by adding a single word not already in the code. Thus the dimension of a code is one less than the dimension of all its children, and every code can be built up from the zero dimensional code by a limited number of augmentations. In our application, this puts all the desired codes on a tree of height 16. At this point one can consider the very naive algorithm of generating all the children for each code, keeping a list and throwing out isomorphs. However, the obvious problem with this approach is the excessive isomorphism computation, which is expensive. This problem is not really ameliorated by storing the canonical representative of each code, as the canonical representative calculation is almost as expensive as isomorphism testing.

The idea behind canonical augmentation is that instead of requiring the objects generated be in canonical form, we require simply that the augmentation itself be canonical. The definition of an *augmentation* is simply an ordered pair (C, C'), where C' is a child of C. An isomorphism of augmentations is a permutation γ such that $(C^\gamma, C'^\gamma) = (D, D')$. For example, $(C, C') \cong (C, C'')$ implies that there is a $\gamma \in \mathrm{Aut}(C)$ such that $C'^\gamma = C''$.

Suppose we have a function p which takes codes C of dimension $k > 0$ to codes $p(C)$ of dimension $k-1$, such that C is a child of $p(C)$. Suppose further that p satisfies the following property: if $C \cong D$, then $(p(C), C) \cong (p(D), D)$. If we have such a function, we can define that an augmentation (C, D) is canonical if $(C, D) \cong (p(D), D)$. Now suppose that we have two augmentations (C, C') and (D, D') such that $C' \cong D'$. If they were both canonical augmentations then we would have

$$(C, C') \cong \big(p(C'), C'\big) \cong \big(p(D'), D'\big) \cong (D, D').$$

In other words, if both augmentations are canonical, then $C' \cong D'$ implies that $C \cong D$. Thus any repeated isomorphs would have to be due to the parents being repeated. If we assume that there

are no isomorphs on the parent level, then this implies that $C = D$, and $(C, C') \cong (C, D')$. In other words, under the framework of canonical augmentation, repeated isomorphs arise only due to automorphisms of the parents. Thus we have Algorithm 1 below.

Algorithm 1 Generate all doubly even binary codes of degree 32, isomorph-free.

```
C_0 := the dimesion zero code, of degree 32
traverse(C_0)
procedure traverse(C):
    report C
    children := the children of C
    representatives := {}
    for D in children:
        if D is minimal in its orbit under Aut(C):
            add D to representatives
    for D in representatives:
        if (C, D) is a canonical augmentation:
            traverse(D)
```

The only remaining question is to produce such a function p. This is where we use the canonical label defined in the last section. If C is a code of dimension k, let γ be the permutation taking C to $c(C)$. Recall that $c(C)$ came with a particular generator matrix determined by the ordering of the words of the code. Removing the last row of that generator matrix gives a code C' of the same dimension as C and applying γ^{-1} to that, we arrive at $p(C)$. Suppose that $C \cong D$, which in particular implies that $c(C) = c(D)$. Let γ_C, γ_D be the permutations taking C, D to $c(C), c(D)$, respectively. These are isomorphisms, so in fact, $\gamma_D^{-1} \circ \gamma_C$ is an isomorphism from C to D taking $p(C)$ to $p(D)$ by the definition of p. This proves the property that $(p(C), C) \cong (p(D), D)$. Further, since we have already done the computation $c(C)$, we can obtain not only the canonical label but also generators for the automorphism group $\mathrm{Aut}(C)$ for free. Thus when we are checking whether $(C', C) \cong (p(C), C)$, we can simply look for an element of $\mathrm{Aut}(C)$ that takes C' to $p(C)$.

The final task of enumerating the codes for storage to disk will approximate the experience of a harvest. Many separate worker processes will be working in parallel, each examining the pairs (C, D) as in the second to last line of Algorithm 1, for a fixed C and many D. Each worker will receive one code at a time, and perform all the steps in Algorithm 1, only instead of recursively calling the function on the last line, they will record the augmented codes that succeed in a table, which will each in turn become the fodder for another worker. As codes are given to workers to augment on, they will be flagged as searched, and once all the codes have been flagged and all the workers are done, all the desired codes will be in the table. The intended end result of all this is a searchable online database which categorizes the codes by N and k, automorphism group size, weight distribution, and perhaps other parameters.

References

[1] S. J. Gates, Jr., L. Rana, *Ultramultiplets: A new representation of rigid 2-d, N=8 supersymmetry*, Phys. Lett. B342 (1995) 132–137. hep-th/9410150

[2] S.-T. Yau (Ed.), *Mirror Manifolds*, International Press, 1990.

[3] B. Greene, S.-T. Yau (Eds.), *Mirror Manifolds II*, International Press, 1996.

[4] D. H. Phong, L. Vinet, S.-T. Yau (Eds.), *Mirror Manifolds III*, American Mathematical Society, Providence, RI, 1999.

[5] K. Hori, S. Katz, A. Klemm, R. Pandharipande, R. Thomas, C. Vafa, R. Vakil, E. Zaslow, *Mirror symmetry*, Vol. 1 of *Clay Mathematics Monographs*, American Mathematical Society, Providence, RI, 2003.

[6] S. K. Donaldson, *An application of gauge theory to the topology of four-manifolds*, Journal of Differential Geometry 18 (1983) 269–316.

[7] E. Witten, *Topological quantum field theory*, Communications in Mathematical Physics 117 (1988) 353–386.

[8] E. Witten, *Monopoles and four-manifolds*, Mathematical Research Letters 1 (1994) 769–796.

[9] K. Iga, *What do topologists want from Seiberg-Witten theory?*, Internat. J. Modern Phys. A 17 (30) (2002) 4463–4514.

[10] S. J. Gates, Jr., L. Rana, *A theory of spinning particles for large N-extended supersymmetry*, Phys. Lett. B 352 (1-2) (1995) 50–58. hep-th/9504025

[11] S. J. Gates, Jr., L. Rana, *A theory of spinning particles for large N-extended supersymmetry. II*, Phys. Lett. B 369 (3-4) (1996) 262–268. hep-th/9510151

[12] S. J. Gates, Jr., W. D. Linch, III, J. Phillips, *When superspace is not enough*. hep-th/0211034

[13] M. Faux, S. J. Gates, Jr., *Adinkras: A graphical technology for supersymmetric representation theory*, Phys. Rev. D (3) 71 (2005) 065002. hep-th/0408004

[14] M. Faux, D. Kagan, D. Spector, *Central charges and extra dimensions in supersymmetric quantum mechanics*. hep-th/0406152

[15] M. Faux, D. Spector, *Duality and central charges in supersymmetric quantum mechanics*, Phys. Rev. D (3) 70 (8) (2004) 085014, 5. hep-th/0311095

[16] M. Faux, D. Spector, *A bps interpretation of shape invariance*, J. Phys. A37 (2004) 10397–10407. quant-ph/0401163

[17] Z. Kuznetsova, M. Rojas, F. Toppan, *Classification of irreps and invariants of the N-extended supersymmetric quantum mechanics*, JHEP 03 (2006) 098. hep-th/0511274

[18] S. Bellucci, S. Krivonos, A. Marrani, E. Orazi, *'Root' action for N=4 supersymmetric mechanics theories*, Phys. Rev. D 73 (2006) 025011. `hep-th/0511249`

[19] S. Bellucci, S. J. Gates, Jr., E. Orazi, *A journey through garden algebras*, Lect. Notes Phys. 698 (2006) 1–47. `hep-th/0602259`

[20] S. Bellucci, S. Krivonos, O. Lechtenfeld, A. Shcherbakov, *Superfield formulation of nonlinear N=4 supermultiplets*. `arXiv:0710.3832[hep-th]`

[21] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *On graph-theoretic identifications of Adinkras, supersymmetry representations and superfields*, Int. J. Mod. Phys. A22 (2007) 869–930. `math-ph/0512016`

[22] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *Adinkras and the dynamics of superspace prepotentials*, Adv. S. Th. Phys. 2 (3) (2008) 113–164. `hep-th/0605269`

[23] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *A counter-example to a putative classification of 1-dimensional, N-extended supermultiplets*, Adv. S. Th. Phys. 2 (3) (2008) 99–111. `hep-th/0611060`

[24] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *On the matter of $N = 2$ matter*, Phys. Lett. B 659 (2008) 441–446. `arXiv/0710.5245`

[25] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[26] W. C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge Univ. Press, 2003.

[27] J. H. Conway, V. Pless, N. J. A. Sloane, *The binary self-dual codes of length up to 32: A revised enumeration*, J. Combinatorial Theory, Series A 60 (1992) 183–195.

[28] T. Hübsch, *Haploid* $(2,2)$*-superfields in* 2*-dimensional space-time*, Nucl. Phys. B555 (3) (1999) 567–628.

[29] R. Q. Almukahhal, T. Hübsch, *Gauging Yang-Mills symmetries in* (1+1)*-dimensional space-time*, Internat. J. Modern Phys. A 16 (29) (2001) 4713–4768.

[30] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *Super-zeeman embedding models on N-supersymmetric world-lines*. `arXiv/0803.3434`

[31] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *Off-shell supersymmetry and filtered Clifford supermodules*. `math-ph/0603012`

[32] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, *Relating doubly-even error-correcting codes, graphs, and irreducible representations of N-extended supersymmetry*, *in: F. Liu, et al. (Eds.), Discrete and Computational Mathematics*, Nova Science Publishers, Inc., Hauppauge, NY, 2008.

[33] C. F. Doran, M. G. Faux, S. J. Gates, Jr., T. Hübsch, K. M. Iga, G. D. Landweber, R. L. Miller, *Adinkras for clifford algebras, and some one-dimensional supermultiplets*. `forthcoming`

[34] N. D. Elkies, *Lattices, linear codes, and invariants, Part II*, Notices of the AMS 47 (11) (2000) 1382–1391.

[35] R. T. Bilous, G. H. J. van Rees, *An enumeration of binary self-dual codes of length 32*, Designs, Codes and Cryptography 26 (1-3) (2002) 66–86.

[36] S. J. Gates, Jr., C. M. Hull, M. Roček, *Twisted multiplets and new supersymmetric nonlinear sigma models and new supersymmetric nonlinear sigma models*, Nucl. Phys. B248 (1984) 157.

[37] P. Gaborit, *Mass formulas for self-dual codes over Z_4 and $F_q + uF_q$ rings*, IEEE Trans. Inform. Theory 42 (4) (1996) 1222–1228.

[38] S. J. Gates, Jr., W. Linch, J. Phillips, L. Rana, *The fundamental supersymmetry challenge remains*, Gravit. Cosmol. 8 (1-2) (2002) 96–100. `hep-th/0109109`

[39] H. B. Lawson, Jr., M.-L. Michelsohn, *Spin geometry*, Vol. 38 of *Princeton Mathematical Series*, Princeton University Press, Princeton, NJ, 1989.

[40] Z. Kuznetsova, F. Toppan, *Refining the classification of the irreps of the 1d n- extended supersymmetry*. `hep-th/0701225`

[41] B. McKay, *Practical graph isomorphism*, Congr. Numer. 30 (30) (1981) 45–87.

[42] B. McKay, *Isomorph-free exhaustive generation*, Journal of Algorithms 26 (1998) 306–324.

[43] J. S. Leon, *Permutation group algorithms based on partition, I: Theory and algorithm*, Journal of Symbolic Computation 12 (1991) 533–583.

[44] B. McKay, *The nauty page*, `http://cs.anu.edu.au/~bdm/nauty/` (2008).

[45] R. Baart, T. Boothby, J. Cramwinckel, W. Joyner, R. Miller, E. Minkes, E. Roijackers, L. Ruscio, C. Tjhai, *GAP package GUAVA*, `http://opensourcemath.org/guava/` (2008).

[46] W. Stein, *Sage: Open Source Mathematical Software (Version 3.0.2)*, The Sage Group, `http://www.sagemath.org` (2008).

[47] P. Darga, M. Liffiton, K. Sakallah, I. Markov, *Exploiting structure in symmetry detection for CNF*, Design Automation Conference, 2004. Proceedings. 41st (2004) 530–534. `http://vlsicad.eecs.umich.edu/BK/SAUCY/`