

Codes and supersymmetry in one dimension

Charles F. Doran¹, Michael G. Faux², Sylvester James
Gates, Jr.³, Tristan Hübsch⁴, Kevin M. Iga⁵,
Gregory D. Landweber⁶ and Robert L. Miller^{7,8}

¹Department of Mathematical and Statistical Sciences, University of
Alberta, Edmonton, Alberta T6G 2G1, Canada
doran@math.ualberta.ca

²Department of Physics, State University of New York, Oneonta,
NY 13825, USA
fauxmg@oneonta.edu

³Center for String and Particle Theory, Department of Physics, University
of Maryland, College Park, MD 20472, USA
gatess@wam.umd.edu

⁴Department of Physics and Astronomy, Howard University, Washington,
DC 20059, USA
thubsch@howard.edu

⁵Natural Science Division, Pepperdine University, Malibu, CA 90263, USA
Kevin.Iga@pepperdine.edu

⁶Mathematics Program, Bard College, Annandale-on-Hudson, NY
12504-5000, USA
gregland@bard.edu

⁷Department of Mathematics, University of Washington, Seattle, WA
98105, USA
rlmill@math.washington.edu

⁸Quid, Inc., 733 Front Street, C1A, San Francisco, CA 94111, USA
rmiller@quid.com

Abstract

Adinkras are diagrams that describe many useful supermultiplets in $D = 1$ dimensions. We show that the topology of the Adinkra is uniquely determined by a doubly even code. Conversely, every doubly even code produces a possible topology of an Adinkra. A computation of doubly even codes results in an enumeration of these Adinkra topologies up to $N = 28$, and for minimal supermultiplets, up to $N = 32$.

1 Introduction, review and synopsis

Although many supersymmetric theories have been known since the 1970s, there is still no overarching classification of supermultiplets, even in one dimension (time). In fact, supersymmetry in one dimension has been the subject of several investigations, for instance the development of the $\mathcal{GR}(d, N)$ algebras [1–3], the development of Adinkras [4], and to other efforts [5–14].

An Adinkra is a directed graph 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 in one dimension [4]. The main purpose of the present work is to determine the kinds of graphs (i.e., the *topology*) that can be used for Adinkras.

The class of supermultiplets described by Adinkras is wide enough to contain many noteworthy superfields. [4, 15] This paper will clarify the conditions for a supermultiplet in one dimension to be described using an Adinkra. Beyond this, by showing that this class is large, we thereby show that the class of supermultiplets is also large, and thus establish that the classification problem of supersymmetry in one dimension is much more intricate than it might appear at first. We also thereby add many new supermultiplets to the literature that were previously unknown, and it is possible that some of these newly discovered supermultiplets may be useful or interesting in their own rights.

Finally, we hope that classifying Adinkras may help one better understand the conditions under which a superfield has an off-shell description. Subject to a particular set of assumptions about dynamics, Siegel and Roček [16] had previously shown that not all supermultiplets have off-shell descriptions. On the other hand, superfields described by Adinkras are off-shell supermultiplets. Therefore, understanding the range of what Adinkras can describe may shed some light on the question of which supermultiplets admit an off-shell description.

1.1 Adinkras

In one dimension, with N supersymmetry generators Q_1, \dots, Q_N , the supersymmetry algebra is

$$\{Q_I, Q_J\} = 2i\delta_{IJ}\partial_\tau, \quad (1.1)$$

where ∂_τ is the derivative in the time direction.

An Adinkra is a finite directed graph, with every vertex colored either white or black, and with every edge colored one of N colors (each color corresponds to one of the supersymmetries Q_I), and each edge drawn with either a solid or a dashed line. The vertices correspond to the component fields (black for fermions, white for bosons) and the edges correspond to the action of each of the Q_I , in a way that is reminiscent of the Cayley diagram of a finitely generated group, or even more analogously, the Schreier diagram of the set of cosets of a subgroup. Details of Adinkras and how they correspond to supermultiplets can be found in [4, 17]. The 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 [17]).
- (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.
- (3) Determine the ways in which edges may be chosen as dashed or solid. This is closely related to the well-known theory of Clifford algebras, 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 [17], 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 codes. Much work has already been done in this area [18–20], and the work described in this paper goes even further in developing this classification; see Appendix B.

We emphasize that we focus here on the representation theory, not the dynamics. This is natural, 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. For instance, 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 [21, 22]. Herein, we defer the task of finding Lagrangians involving the supermultiplets considered in this paper. In [23, 24], 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. [17, 25].

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:

$$\varkappa(N) := \begin{cases} 0 & \text{for } N < 4, \\ 1 & \text{for } N = 4, 5, \\ 2 & \text{for } N = 6, \\ 3 & \text{for } N = 7, \\ 4 + \varkappa(N-8) & \text{for } N \geq 8, \text{ recursively.} \end{cases} \quad (1.2)$$

- (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) There is a one-to-one correspondence between possible chromotopologies of connected Adinkras and doubly even codes of length N .
 - (a) Each connected chromotopology has, associated to it, a doubly even code of length N and dimension $k \leq \varkappa(N)$ that records which paths connect a vertex to itself.
 - (b) The chromotopology is then the quotient of the colored N -dimensional cube by this code. (The colored N -cube is the set of vertices and edges of the N -dimensional cube $[0, 1]^N$, with colors on the

- edges determined by which axis it is parallel to, and colors of vertices according to the number, modulo 2, of coordinates that are 1).
- (c) This quotient can be viewed as an iterated k -fold \mathbb{Z}_2 -quotient.
 - (d) These chromotopologies really do come from supermultiplets in $D = 1$ dimension, and if the arrows are chosen properly (one-hooked) we can arrange it so that it is easy to see that different codes give rise to different supermultiplets.
 - (e) Permuting the columns of a code corresponds to permuting the colors of the chromotopology, which in turn describes R -symmetries of the supermultiplet.
 - (f) There are an enormous multitude of distinct doubly-even codes for $N \leq 32$, even when counting permutation equivalent codes as the same code. Thus, there is an enormous multitude of Adinkra chromotopologies.

This 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 first major result, in Section 4, is that each Adinkra chromotopology gives rise to a doubly even code. It will be convenient to provide a few classes of examples of doubly even codes for our discussions, and to give a sense for how many doubly even codes there are, so this is done in Section 5. We then turn to the second major result: that every doubly even code actually arises as the code for an Adinkra chromotopology for a supermultiplet. This is done in Sections 6 and 7. Sections 8 and 9 discuss some consequences and directions for further research.

2 Codes

We begin with a brief introduction to the theory of codes. For a more thorough introduction to the subject, see [18–20].

We think of $\{0, 1\}$ as a group with the operation \boxplus , which is addition modulo 2, i.e., the group \mathbb{Z}_2 . For the purposes of this paper, a *code* of length N means a subgroup of $\{0, 1\}^N$.¹ Although the standard notation for an element of a cartesian product is (x_1, x_2, \dots, 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 word. This word is called a *codeword* if it is in the code.

¹In the coding literature, there are sometimes other more general definitions of codes. What we have described is a linear binary block code of length N .

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. Codes are then linear subspaces of $\{0, 1\}^N$. Every code has a basis, called a *generating set*, g_1, \dots, g_k , so that every codeword can be written uniquely as a sum

$$\sum_{i=1}^k x_i g_i, \quad (2.1)$$

where the coefficients x_1, \dots, x_k are each either 0 or 1. The number k is the same for every generating set for a given code, and is called the *dimension* of the 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 $k \times N$ matrix, where each row is an element of the generating set.

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

A 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 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 i th bit of $v \& w$ is 1 if and only if the i th bit of v and the i th bit of w are both 1. A basic fact in $\{0, 1\}^N$ is

$$\text{wt}(v \boxplus w) = \text{wt}(v) + \text{wt}(w) - 2 \text{wt}(v \& w). \quad (2.2)$$

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

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

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 $\langle v, v \rangle \equiv \text{wt}(v) \pmod{2}$, and thus, when $\text{wt}(v)$ is even, v is orthogonal to itself. Also note that $\langle v, w \rangle \equiv \text{wt}(v \& w) \pmod{2}$. One important consequence for us is that if $\text{wt}(v)$ and $\text{wt}(w)$ are multiples of 4, then (2.2) implies that $\text{wt}(v \boxplus w)$ 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, \dots, Q_N , satisfying the following supersymmetry relations:

$$\{Q_I, Q_J\} = 2i \delta_{IJ} \partial_\tau, \quad [\partial_\tau, Q_I] = 0, \quad I, J = 1, \dots, N. \quad (3.1)$$

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 \mathcal{M} is a real, 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*, $\phi_1(\tau), \dots, \phi_m(\tau)$ and $\psi_1(\tau), \dots, \psi_n(\tau)$, respectively; each component field is a function of time, τ . The supersymmetry transformations, generated by the Hermitian operators Q_1, \dots, Q_N , act linearly on \mathcal{M} while satisfying equations (3.1). The supermultiplet is *off-shell* if no differential equation is imposed on it². The number of bosons as fermions is then the same, guaranteed by supersymmetry.

3.2 Building supermultiplets from Adinkras

The authors of [4, 17, 23] introduced and then studied Adinkras, diagrams that encode the transformation rules of the component fields under the action of the supersymmetry generators Q_1, \dots, Q_N .

Supermultiplets that can be described by Adinkras have a collection of bosonic and fermionic component fields and a collection of supersymmetry

²Logically, 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 [15].

generators Q_1, \dots, Q_N , so that: (1) Given a bosonic field ϕ and a supersymmetry generator Q_I , the transformation rule for Q_I of ϕ is of the form

$$\text{either } Q_I \phi = \pm \psi, \quad (3.2)$$

$$\text{or } Q_I \phi = \pm \partial_\tau \psi, \quad (3.3)$$

for some fermionic field ψ . (2) Given instead a fermionic field η and a supersymmetry generator Q_I , the transformation rule of Q_I on η is of the form

$$\text{either } Q_I \eta = \pm i B, \quad (3.4)$$

$$\text{or } Q_I \eta = \pm i \partial_\tau B, \quad (3.5)$$

for some bosonic field B . In particular, these supersymmetry generators act linearly using first-order differential operators. Furthermore, the supersymmetry algebra requires that

$$Q_I \phi = \pm \psi \iff Q_I \psi = \pm i \partial_\tau \phi, \quad (3.6)$$

and

$$Q_I \psi = \pm \partial_\tau \phi \iff Q_I \phi = \pm i \psi, \quad (3.7)$$

and where the \pm signs are correlated to preserve equations (3.1).





More generally, suppose we label the bosons ϕ_1, \dots, ϕ_m and the fermions ψ_1, \dots, ψ_m . Choose an integer I with $1 \leq I \leq N$, and an integer A with $1 \leq A \leq m$. For each such pair of integers, we consider the transformation rules for Q_I on the boson ϕ_A , and we might expect that these will be of the form

$$Q_I \phi_A(\tau) = c \partial_\tau^\lambda \psi_B(\tau), \quad (3.8)$$

where $c = \pm 1$, $\lambda = 0$ or 1 , and B is an integer with $1 \leq B \leq m$, so that ψ_B is some fermion; each of c, λ, B will, in general, depend on I and A . Note that

$$\lambda = [\phi_A] - [\psi_B] + \frac{1}{2}, \quad (3.9)$$

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

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 I^{th} color.

for ϕ_A and ψ_B to have a definite engineering dimension — provided the transformation rules had only dimensionless constants as we assume throughout. For each such transformation rule, we will get a corresponding transformation rule for the Q_I on the fermion $\psi_B(\tau)$ that looks like this:

$$Q_I \psi_B(\tau) = \frac{i}{c} \partial_\tau^{1-\lambda} \phi_A(\tau). \tag{3.10}$$

Equations(3.8)–(3.10) constitute all of the transformation rules on the bosons and fermions, respectively.

Definition 3.1. A supermultiplet, \mathcal{M} , is *adinkraic* if all of its supersymmetric transformation rules are of the form (3.8) and (3.10).

For each adinkraic supermultiplet, its *Adinkra*, $\mathcal{A}_\mathcal{M}$, is a directed graph, consisting of a set of vertices, V , a set of edges, E , a coloring C of the edges, a set of their orientations, O , and a labeling D of each edge corresponding to whether or not it is dashed.

Each component field of \mathcal{M} is represented by a vertex in $\mathcal{A}_\mathcal{M}$: white for bosonic fields and black for fermionic ones, thus equipartitioning the vertex set $V \rightarrow W$. Every transformation rule of the form (3.8) is depicted by an edge connecting the vertex corresponding to ϕ_A to the vertex corresponding to ψ_B , and color the edge with the I^{th} color. We use a dashed edge if $c = -1$, and oriented it from ϕ_A to ψ_B if $\lambda = 0$ and the other way around if $\lambda = 1$.

Table 1 illustrates the four possibilities for an edge.

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 (3.2)–(3.5).

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

In particular, from Definition 3.1 of an Adinkra, we forget the bipartition (black or white) of the vertices and all additional information associated to the edges, namely, the color of the edges (which records the Q_I corresponding to that edge), dashedness ($c = \pm 1$), and direction (the exponent of ∂_τ in (3.2)–(3.5)).

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 3.1 of an Adinkra, we forget the dashedness of the edges and the direction of the arrow along the edge.

Suppose we have an adinkraic multiplet with component fields $(F_1(\tau), \dots, F_{2m}(\tau))$,³ and let \mathcal{A} be its Adinkra. We wish to consider the chromotopology. That is, we wish to ignore the arrows on the edges and ignore whether an edge is dashed. We can capture this information by taking the vertex set of the Adinkra $V = \{v_1, \dots, v_{2m}\}$, the coloring of these vertices, and the edge set E with its coloring. Since this graph is inherited from an Adinkra, for every vertex and every $I \in \{1, \dots, N\}$ 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, \dots, q_N from the vertex set $V = \{v_1, \dots, v_{2m}\}$ to itself, such that whenever A , B , and I are

³Previously, we labeled the bosons ϕ_1, \dots, ϕ_m and the fermions ψ_1, \dots, ψ_m . Here, it will be convenient for notation to treat them on the same footing.

such that there is an equation of type (3.8) or (3.10),

$$Q_I F_A(\tau) = c \partial_\tau^\lambda F_B, \quad \implies \quad q_I(v_A) = v_B. \quad (4.1)$$

Note 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:

$$q_I^2 = \mathbf{1}, \quad \text{i.e.,} \quad q_I(q_I(v)) = v, \text{ for all } v \in V, \quad (4.2)$$

and

$$q_I q_J = q_J q_I, \quad \text{i.e.,} \quad q_I(q_J(v)) = q_J(q_I(v)), \text{ for all } I, J, \text{ and all } v \in V. \quad (4.3)$$

The vertex set $V = \{v_1, \dots, v_{2^N}\}$, the partition of V into bosons and fermions, and the maps $q_I : V \rightarrow V$ then encode the chromotopology.

4.1 The colored N -cube chromotopology

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 $\vec{p} = (p_1, \dots, p_N) \in \mathbb{R}^N$, 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 $\text{wt}(\vec{p})$, equals the number of $J \in \{1, \dots, N\}$ for which $p_J = 1$.

There is a natural chromotopology associated to the N -cube: we color vertices of even weight white, and vertices of odd weight we color black. 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 I th coordinate with the I th color. The result is called the colored N -cube.

Using the notation from the previous section, we determine q_1, \dots, q_N as follows:

$$q_I(p_1, \dots, p_{I-1}, p_I, p_{I+1}, \dots, p_N) = (p_1, \dots, p_{I-1}, 1 - p_I, p_{I+1}, \dots, p_N). \quad (4.4)$$

There are indeed Adinkras of supermultiplets that have this chromotopology, as we will see in Section 7.

4.2 The quotient of a colored cube by a code

We now introduce another chromotopology: given a colored N -cube and a code C of length N , we will create a chromotopology called $[0, 1]^N/C$.

Before we do so, recall that C is a subset of $\{0, 1\}^N$, and so codewords (elements of C) are vectors, which we can write as \vec{g} . Likewise, the vertices of the colored N -cube are elements of $\{0, 1\}^N$ as well.

For each vertex $\vec{v} \in \{0, 1\}^N$, we identify it with all vertices of the form $\vec{v} \boxplus \vec{g}$ for $\vec{g} \in C$. The resulting vertex set we call V' . Likewise, consider an edge e colored I , connecting \vec{v} to \vec{w} (note that this means the \vec{v} and \vec{w} are identical except in bit I , where they differ). We identify e with the other edges colored I that connect $\vec{v} \boxplus \vec{g}$ with $\vec{w} \boxplus \vec{g}$ (note that $\vec{v} \boxplus \vec{g}$ and $\vec{w} \boxplus \vec{g}$ also differ precisely at bit I , and so there is a unique edge connecting them, and it is colored I). One way to view this is that the group C acts on the vertex set and the edge set by \boxplus , and we are taking the orbit space of this group action.

Now note that we have thereby created a graph with vertices and edges, and the colors of the edges is determined by this construction. In order to consistently label the colors of the vertices, we must only identify vertices when they are of the same color (boson or fermion). Since the color of a vertex in the colored N -cube is determined by its weight modulo 2, and since equation (2.2) says that

$$\text{wt}(v \boxplus g) = \text{wt}(v) + \text{wt}(g) - 2 \text{wt}(v \& g), \quad (4.5)$$

we see that $\text{wt}(v \boxplus g)$ agrees with $\text{wt}(v)$ modulo 2 for all g if and only if $\text{wt}(g)$ is always even. Therefore, if C is an even code, this quotient of a colored N -cube gives a consistent coloring of the vertices.

This quotient may not necessarily be the chromotopology of the Adinkra of a supermultiplet: we will see in the next section that for this, the code must be doubly even. We will then see in Section 7 that this is sufficient.

4.3 The code of an Adinkra chromotopology

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. So if we can determine the chromotopologies of connected Adinkras, the general chromotopology of an Adinkra

will be a finite disjoint union of such. Thus we focus our attention to connected Adinkras.

Construction 4.1. *Given a connected Adinkra with N edge colors, we construct a code of length N as follows.*

Let V be the set of vertices. Define $q_I : V \rightarrow V$ as above. Pick any vertex $v_ \in V$ and fix it. Now define*

$$C = \{(x_1, \dots, x_N) \mid q_1^{x_1}(\dots q_N^{x_N}(v_*)) = v_*\}. \tag{4.6}$$

Proposition 4.1. *The set C defined in this construction is a code.*

Proof. We need to prove that the elements of C form a subgroup of $\{0, 1\}^N$. To prove closure, let $\vec{x} = (x_1, \dots, x_N)$ and $\vec{y} = (y_1, \dots, y_N)$ be elements of C . By equation (4.3), we have

$$q_1^{x_1 \boxplus y_1} \dots q_N^{x_N \boxplus y_N}(v_*) = q_1^{x_1} \dots q_N^{x_N}(q_1^{y_1} \dots q_N^{y_N}(v_*)) = v_* \tag{4.7}$$

and so $\vec{x} \boxplus \vec{y} \in C$.

To prove identity, note that $q_1^0 \dots q_N^0(v_*) = v_*$.

To prove inverse, note that in $\{0, 1\}^N$, every element is its own inverse. \square

Proposition 4.2. *The code C is independent of the choice of vertex v_* .*

Proof. Let v_0 be any other vertex in the Adinkra. Since the Adinkra is connected, there is a sequence of edges connecting v_* to v_0 . If the colors of these edges are I_1, \dots, I_m , then by the definition of q_I , we have

$$q_{I_1} \dots q_{I_m}(v_*) = v_0. \tag{4.8}$$

Using equation (4.3), we can rearrange the q_{I_i} so that they are in increasing order, and we can furthermore cancel pairs of q_{I_i} so that each q_I appears at most once. In this way, we obtain a vector $\vec{t} = (t_1, \dots, t_N) \in \{0, 1\}^N$ so that

$$q_1^{t_1} \dots q_N^{t_N}(v_*) = v_0. \tag{4.9}$$

Now suppose \vec{g} is a codeword obtained by applying Construction 4.1 using v_* . Then

$$q_1^{g_1} \dots q_N^{g_N}(v_*) = v_*. \tag{4.10}$$

Apply $q_1^{t_1} \cdots q_N^{t_N}$ to both sides, using equation (4.3) to commute the q_I past each other, and we get

$$q_1^{g_1} \cdots q_N^{g_N} (q_1^{t_1} \cdots q_N^{t_N} (v_*)) = q_1^{t_1} \cdots q_N^{t_N} (v_*). \quad (4.11)$$

Then we plug in equation (4.9) and get

$$q_1^{g_1} \cdots q_N^{g_N} (v_0) = v_0. \quad (4.12)$$

Therefore, any codeword for v_* is a codeword for v_0 . Applying the argument with v_0 and v_* switched proves the converse. \square

Remark 4.1. Thus we can determine the code of an Adinkra as follows: first pick any vertex v_* . Then for each $\vec{g} = (g_1, \dots, g_N) \in \{0, 1\}^N$, write down the set S of all colors I for which $g_I = 1$, in some order. Then take the path that starts at v_* and goes along edges according to the colors in the set S . If we return to v_* , then \vec{g} is in the code.

For example, consider the Adinkra in (7.9) in Section 7 below. If we take $v_* = \phi_{(0000)}$, we can follow the colors black (color 1), then red (color 2), then green (color 3), then blue (color 4). This takes us on a path that goes to $\psi_{(1000)}$ then $\phi_{(1100)}$, then $\psi_{(1110)}$, then back to $\phi_{(0000)}$. From this we see that 1111 is in the code. On the other hand, suppose we took red (color 2), then blue (color 4). This takes us from $\phi_{(0000)}$ to $\psi_{(0100)}$ to $\phi_{(1010)}$, which is not $\phi_{(0000)}$. So we see that 0101 is not in the code. You can check that the code is $\{0000, 1111\}$ no matter which starting vertex v_* we use.

Theorem 4.3. *Every connected Adinkra chromotopology is isomorphic to a quotient of a colored N -dimensional cube by the code of the chromotopology.*

Proof. Suppose we have a connected Adinkra with N edge colors, with vertex set V and edge set E . Pick any one bosonic vertex $v_* \in V$ and fix it. As in Construction 4.1, we define

$$C = \{(g_1, \dots, g_N) \mid q_1^{g_1} \cdots q_N^{g_N} (v_*) = v_*\}. \quad (4.13)$$

We then take the quotient of the colored N -cube by C . Let the vertex set of this quotient be called W , and let the edge set be called F . Now we produce an isomorphism from the chromotopology of the quotient of the colored N -cube to the chromotopology of the original Adinkra. By this we mean a bijection from W to V that preserves the colors of the vertices, and a bijection from F to E that preserves the colors of the edges and the relationship of which edges are incident with which vertices.

To define the bijection $f : W \rightarrow V$, let $w \in W$ be any vertex in W . It comes from a quotient of an N -cube, meaning it is the result of identifying vectors in $\{0, 1\}^N$, so there is at least one vector $(w_1, \dots, w_N) \in \{0, 1\}^N$ that was used in the identification to produce w . We define

$$f(w) = q_1^{w_1} \cdots q_N^{w_N}(v_*). \tag{4.14}$$

We note that any other choice of N -tuple that produces w would be $(w_1, \dots, w_N) \boxplus \vec{g}$ where $\vec{g} \in C$, and

$$q_1^{w_1 \boxplus g_1} \cdots q_N^{w_N \boxplus g_N}(v_*) = q_1^{w_1} \cdots q_N^{w_N}(q_1^{g_1} \cdots q_N^{g_N}(v_*)) = q_1^{w_1} \cdots q_N^{w_N}(v_*) = f(w), \tag{4.15}$$

so that f does not depend on the choice of (w_1, \dots, w_N) .

We now prove f is injective. Suppose $f(w) = f(x)$. Then

$$q_1^{w_1} \cdots q_N^{w_N}(v_*) = q_1^{x_1} \cdots q_N^{x_N}(v_*). \tag{4.16}$$

By using equation (4.3), we obtain

$$q_1^{w_1 \boxplus x_1} \cdots q_N^{w_N \boxplus x_N}(v_*) = v_*, \tag{4.17}$$

which implies $(w_1, \dots, w_N) \boxplus (x_1, \dots, x_N) = (g_1, \dots, g_N)$ is in the code C . Then $(x_1, \dots, x_N) = (w_1, \dots, w_N) \boxplus (g_1, \dots, g_N)$, so that the vertices (w_1, \dots, w_N) and (x_1, \dots, x_N) are identified in the construction of the quotient. Therefore $w = x$.

To see that f is surjective, let $v \in V$ be any vertex. Recall that we are assuming 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, \dots, I_k , when tracing from the vertex v to v_* . If we then apply to v_* a corresponding sequence of q_I 's, we obtain:

$$q_{I_1} \cdots q_{I_k}(v_*) = v. \tag{4.18}$$

Using the commutativity of the q 's to put them in numerical order and equation (4.2) to eliminate those that appear more than once, we can write this as

$$q_1^{x_1} \cdots q_N^{x_N}(v_*) = v. \tag{4.19}$$

However, the left hand side is just $f(x_1, \dots, x_N)$. Therefore, f is surjective.

We now examine the edges. Let e be an edge in F with color I . It connects two vertices v and w , so that $q_I(v) = w$. We wish to show that there is an edge in E colored I connecting $f(v)$ to $f(w)$; that is, that $q_I(f(v)) = f(w)$.

Choose a vector $(v_1, \dots, v_N) \in \{0, 1\}^N$ that gets used in the identification to make v . For (w_1, \dots, w_N) we take

$$(w_1, \dots, w_N) = (v_1, \dots, v_{I-1}, 1 - v_I, v_{I+1}, \dots, v_N), \tag{4.20}$$

which is possible because this is the edge of color I coming from (v_1, \dots, v_N) in the original N -cube before quotienting. Now we apply q_I to $f(v)$:

$$q_I(f(v)) = q_I(q_1^{v_1} \cdots q_N^{v_N}(v_*)) \tag{4.21}$$

$$= q_1^{v_1} \cdots q_{I-1}^{v_{I-1}} q_I^{1-v_I} q_{I+1}^{v_{I+1}} \cdots q_N^{v_N}(v_*) \tag{4.22}$$

$$= q_1^{w_1} \cdots q_N^{w_N}(v_*) \tag{4.23}$$

$$= f(w). \tag{4.24}$$

Therefore, there is an edge colored I connecting $f(v)$ to $f(w)$.

The fact that π sends bosons to bosons and fermions to fermions can now be seen by the fact that it sends the boson $(0, \dots, 0) \in \{0, 1\}^N$ 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. \square

We next consider what kinds of codes C can appear in Adinkras. We have already noted in the previous section that in order to get a consistent vertex-coloring, C must be an even code. We now show that the edge-dashing of the original Adinkra (which we have been ignoring in studying chromotopologies) will force C to be *doubly* even; that is, for each $\vec{g} \in C$, $\text{wt}(\vec{g})$ must be a multiple of 4.

Theorem 4.4. *The code for an Adinkra must be doubly even.*

Proof. To prove this, 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 $\vec{x} \in C$ means that

$$q_1^{x_1} (\cdots q_N^{x_N}(v_*)) = v_*, \tag{4.25}$$

which means, if F_* is the component field that is represented by v_* , then

$$Q_1^{x_1} \cdots Q_N^{x_N} F_*(\tau) = c \partial_\tau^{\text{wt}(\vec{x})/2} F_*(\tau) \tag{4.26}$$

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 $\text{wt}(\vec{x})$ is even, so that $\frac{1}{2} \text{wt}(\vec{x})$ is indeed an integer and equation (4.26) is well defined.

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

$$Q_1^{x_1} \cdots Q_N^{x_N} \cdot Q_1^{x_1} \cdots Q_N^{x_N} F_*(\tau) = c^2 \partial_\tau^{\text{wt}(\vec{x})} F_*(\tau). \tag{4.27}$$

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

$$(-1)^{\binom{\text{wt}(\vec{x})}{2}} Q_1^{2x_1} \cdots Q_N^{2x_N} F_*(\tau) = c^2 \partial_\tau^{\text{wt}(\vec{x})} F_*(\tau), \tag{4.28}$$

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

$$(-1)^{\binom{\text{wt}(\vec{x})}{2}} i^{\text{wt}(\vec{x})} \partial_\tau^{\text{wt}(\vec{x})} F_*(\tau) = c^2 \partial_\tau^{\text{wt}(\vec{x})} F_*(\tau). \tag{4.29}$$

Comparing the exponents of ∂_τ in equation (4.29) confirms the exponent of ∂_τ in equation (4.26). Comparing the numerical coefficients produces:

$$c^2 = (-1)^{\binom{\text{wt}(\vec{x})}{2}} i^{\text{wt}(\vec{x})} = i^{\text{wt}(\vec{x})(\text{wt}(\vec{x})-1)+\text{wt}(\vec{x})} = i^{\binom{\text{wt}(\vec{x})}{2}}. \tag{4.30}$$

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

Using (3.8) and (3.10) repeatedly, we see that c is ± 1 if $\text{wt}(\vec{x}) = 0 \pmod{4}$, and $\pm i$ if $\text{wt}(\vec{x}) = 2 \pmod{4}$. But $c^2 = 1$ implies that $c = \pm 1$, and this can happen only if $\text{wt}(\vec{x}) = 0 \pmod{4}$. □

Remark 4.2. The converse is also true. That is, if G is a doubly even 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. This will be done in Section 7.

5 Finding doubly even codes

5.1 Examples of doubly even codes

Since doubly even codes classify chromotologies, 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

$$\begin{bmatrix} 111100000 \cdots 00000 \\ 001111000 \cdots 00000 \\ 000011110 \cdots 00000 \\ \vdots \\ 000000000 \cdots 01111 \end{bmatrix}. \tag{5.1}$$

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,

$$\begin{bmatrix} 11110000 \\ 00111100 \\ 00001111 \end{bmatrix} \text{ generates } \left\{ \begin{array}{l} v_0 = 00000000 \\ v_1 = 11110000 \\ v_2 = 00111100 \\ v_3 = 00001111 \\ v_1 \boxplus v_2 = 11001100 \\ v_2 \boxplus v_3 = 00110011 \\ v_1 \boxplus v_3 = 11111111 \\ v_1 \boxplus v_2 \boxplus v_3 = 11000011 \end{array} \right\}. \tag{5.2}$$

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... For instance,

$$e_7 : \begin{bmatrix} 1111000 \\ 0011110 \\ 1010101 \end{bmatrix}, \quad e_8 : \begin{bmatrix} 11110000 \\ 00111100 \\ 00001111 \\ 10101010 \end{bmatrix}, \quad (5.3)$$

and we then write:

$$e_{15} : \begin{bmatrix} 1111000000000000 \\ 0011110000000000 \\ 0000111100000000 \\ 0000001111000000 \\ 0000000011110000 \\ 0000000000111100 \\ 0000000000011110 \\ 1010101010101010 \end{bmatrix}, \quad e_{16} : \begin{bmatrix} 11110000000000000 \\ 00111100000000000 \\ 00001111000000000 \\ 00000011110000000 \\ 00000000111100000 \\ 00000000001111000 \\ 00000000000111100 \\ 00000000000011110 \\ 1010101010101010 \end{bmatrix}, \quad (5.4)$$

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. Conway *et al.* [20], describe what it calls “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. The previously defined d_N codes likewise relate to the root lattice for the Lie algebras D_N .

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

There are many other doubly even codes, and the number grows quickly as N becomes large; see Appendix B and [20, 26].

⁴This is the only code mentioned here not specifically named in [19].

5.2 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

$$\begin{bmatrix} 1111000 \\ 0011101 \\ 1010110 \end{bmatrix}. \quad (5.5)$$

This is another doubly even code, and it is different from e_7 as given in equation (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:

$$\begin{bmatrix} 1111000 \\ 0011110 \\ 1010101 \end{bmatrix} \longrightarrow \begin{bmatrix} 1111000 \\ 1100110 \\ 1010101 \end{bmatrix}. \quad (5.6)$$

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 .

If a permutation of the columns sends one code to another, the codes are said to be *permutation-equivalent*. 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 [27] 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 chromotologies, where the issue is clear.

We have created computer algorithms to find and count permutation equivalence classes of doubly even codes. Table 2 provides a list of such

Table 2: 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.

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: 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.

$N \setminus k$	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	464579	1103023	817167	174809	13578	525				
28	7	39	263	2136	20812	224825	1917212	7631323	8948070	2550127	203178	7402	151			
29	7	39	287	2693	34233	555804	8084014	*	*	*	4837471	133563	1940			
30	7	46	359	3866	61871	1477074	*	*	*	*	*	*	70744	731		
31	7	46	400	4972	*	*	*	*	*	*	*	*	*	25497	210	
32	8	55	506	*	*	*	*	*	*	*	*	*	*	*	7689	85
N/k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

for $N \leq 11$, but beyond this the list is too long to include here. Instead, we provide table 3, which provides the number of permutation equivalence classes of doubly even codes for $N \leq 32$, except for a few cases where we do not yet know the answer. Details on how these tables were created are in Appendix B.

6 Supersymmetry and Clifford algebras

It has been understood for a long time that the supersymmetry algebra (3.1) has a formal similarity with the Clifford algebra generated by the Dirac Γ_I matrices⁵:

$$\{\Gamma_I, \Gamma_J\} = 2 \delta_{IJ} \mathbb{1}. \tag{6.1}$$

⁵In (3.1), $I, J = 1, \dots, N$ count *fermionic* dimensions. Accordingly, the quadratic form, δ_{IJ} , occurring in the right-hand side of equation (6.1) is positive definite. This is unlike

One manifestation of this was the study of the spinning particle by Gates and Rana [1, 2], resulting in the Scalar Supermultiplet (which we herein rename *Isoscalar Supermultiplet*), defined in terms of the $\mathcal{GR}(d, N)$ algebra, a form of the Clifford algebra. We will review their work first, and more explicitly describe how it relates to Clifford algebras; this will motivate the more general construction of relating supermultiplets to a representation of the Clifford algebra.

The main idea relating supermultiplets and Clifford representations is to note that the supersymmetry algebra in one dimension and the Clifford algebra differ only in that in the former, there is a factor of i and a derivative ∂_τ . So, to turn a supermultiplet into a representation of the Clifford algebra, we can simply forget the factors of i and the derivatives ∂_τ . This construction is in a sense the same construction quotienting by the ideal $(H - 1)$ in Ref. [25], where $H = i\partial_\tau$. This also removes the \mathbb{Z} -grading afforded by the notion of engineering dimension. Conversely, to take a representation of the Clifford Algebra, we must re-insert factors of i and ∂_τ . The factor of i must go on the right-hand side of the transformation rules of the fermions (see Appendix A.1) and if we declare that the ∂_τ goes in that same place, we end up with a supermultiplet called the Isoscalar supermultiplet.

6.1 The isoscalar supermultiplet and the clifford algebra

The Isoscalar supermultiplet consists of component fields $(\phi_1, \dots, \phi_m | \psi_1, \dots, \psi_m)$, written as column vectors:

$$\Phi = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_m \end{bmatrix} \quad \text{and} \quad \Psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_m \end{bmatrix}. \tag{6.2}$$

and with supersymmetry transformations given by the following Ansatz:

$$Q_I \Phi = \mathbb{L}_I \Psi, \tag{6.3}$$

$$Q_I \Psi = i \mathbb{R}_I \partial_\tau \Phi, \tag{6.4}$$

the *typical* field theory use of the Clifford/Dirac algebra, where I, J would count *bosonic* dimensions of spacetime, implying the Lorentzian signature for the quadratic form on the right-hand side of equation (6.1). Another point is that traditionally there would be a minus sign on the right-hand side; but this difference is equivalent to changing the metric from positive definite to negative definite. In the language of [28], this leads to the Clifford algebra $\text{Cl}(0, N)$ instead of $\text{Cl}(N) = \text{Cl}(N, 0)$.

where \mathbb{L}_I and \mathbb{R}_I are $m \times m$ real matrices to be determined. The supersymmetry algebra (3.1) then implies

$$\mathbb{L}_I \mathbb{R}_J + \mathbb{L}_J \mathbb{R}_I = 2 \delta_{IJ} \mathbb{1}, \tag{6.5}$$

$$\mathbb{R}_I \mathbb{L}_J + \mathbb{R}_J \mathbb{L}_I = 2 \delta_{IJ} \mathbb{1}, \tag{6.6}$$

where $\mathbb{1}$ is the $m \times m$ identity matrix. The $I = J$ cases of these equations imply that $\mathbb{R}_I = \mathbb{L}_I^{-1}$.

Remark 6.1. The additional requirement $\mathbb{L}_I = \pm \mathbb{R}_I^T$ was also made in [1], but this is needed only when writing Lagrangians and will play no rôle in this paper.

We create, for each I , a $2m \times 2m$ real matrix Γ_I of the form

$$\Gamma_I := \left[\begin{array}{c|c} \mathbf{0} & \mathbb{L}_I \\ \hline \mathbb{R}_I & \mathbf{0} \end{array} \right]. \tag{6.7}$$

Equations (6.5) and (6.6) then imply that these $\Gamma_1, \dots, \Gamma_N$ indeed satisfy equation (6.1), the algebra of the Dirac gamma matrices in dimension N , whence the notation.

We do not insist that m be minimal here (a minimality criterion was used in Refs. [1, 2]), nor do we adopt any particular convention for the type of the Γ_I matrices, except that the entries must be real.

The property of being block off-diagonal corresponds to the existence of a fermion number operator $(-1)^F$, which anti-commutes with the Γ_I . If we write our fields listing the bosons first, followed by the fermions, then $(-1)^F$ will be a diagonal matrix of the form

$$(-1)^F = \left[\begin{array}{c|c} \mathbb{1} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{1} \end{array} \right]. \tag{6.8}$$

If we define Γ_0 to be $(-1)^F$, then Γ_0 anticommutes with the remaining Γ_I :

$$\Gamma_0^2 = +\mathbb{1}, \quad \{\Gamma_0, \Gamma_I\} = 0. \tag{6.9}$$

The formal algebra generated by $\Gamma_0, \Gamma_1, \dots, \Gamma_N$, is then defined by the anti-commutation relations:

$$\{\Gamma_I, \Gamma_J\} = 2 \delta_{IJ} \mathbb{1}, \quad \text{for } I, J = 0, 1, \dots, N \tag{6.10}$$

and therefore $\{\Gamma_0, \dots, \Gamma_N\}$ satisfy (6.1). The algebra generated by the $\Gamma_1, \dots, \Gamma_N$ is the *Clifford algebra* $\text{Cl}(0, N)$, and the inclusion of Γ_0 extends

$\text{Cl}(0, N)$ into the Clifford algebra $\text{Cl}(0, N+1)$. Any set of real matrices representing these $\Gamma_0, \Gamma_1, \dots, \Gamma_N$ such that the above algebra closes is called a *real Clifford representation* [28].

Now suppose we are given a real Clifford representation. We can split the representation into the +1 and -1 eigenvalues for Γ_0 , and if we choose a basis that respects this splitting, then Γ_0 will be of the form given above, in (6.8). The Ansatz (6.3)–(6.4) then gives the transformation rules for the Isoscalar multiplet corresponding to the Clifford representation.

7 Constructing Adinkras from codes

The main construction in this paper is to take a description of an N -dimensional cube $[0, 1]^N$ and a doubly even $[N, k]$ -code C , and construct a real Clifford representation. The Isoscalar supermultiplet corresponding to this real Clifford representation will then be a supermultiplet for N -extended supersymmetry in one dimension, and will have an Adinkra with a chromotopology given by $[0, 1]^N/C$.

7.1 Clifford supermultiplets and cubical Adinkras

We start with the N -dimensional cube itself. To obtain an Adinkra with chromotopology the N -cube, we take the Clifford Algebra itself, $\text{Cl}(0, N)$, as a Clifford representation. The Clifford algebra $\text{Cl}(0, N)$ is a real 2^N -dimensional vector space, spanned by products of the form $\Gamma_{I_1} \cdots \Gamma_{I_k}$, where $I_1 < \cdots < I_k$ [28]. As a vector space, it splits as a direct sum of two vector spaces: the even and odd parts. The even (resp. odd) part is the subspace spanned by products of even (resp. odd) numbers of Γ_I matrices.

The Clifford algebra $\text{Cl}(0, N+1)$ acts on $\text{Cl}(0, N)$ in the following way: for every $1 \leq I \leq N$, Γ_I acts by multiplication on the left. The operator Γ_0 multiplies the even Γ -monomials by 1 and the odd Γ -monomials by -1. It is then easy to see that the Clifford algebra (6.10) holds.

We take as a basis for $\text{Cl}(0, N)$ the products $\Gamma_{I_1} \cdots \Gamma_{I_k}$ as above. Specifically, for every vertex of the cube $\vec{x} = (x_1, \dots, x_N) \in \{0, 1\}^N$, we define a component field⁶

$$e_{\vec{x}} = \Gamma_1^{x_1} \cdots \Gamma_N^{x_N}. \tag{7.1}$$

⁶Elements of a Clifford algebra can be used to realize the component fields of a supermultiplet, by allowing the element of the Clifford algebra to be a function of the time-like coordinate τ . This construction was explicitly carried out in equation (67) of [3].

We note that $\Gamma_I e_{\vec{x}} = \Gamma_I \Gamma_1^{x_1} \dots \Gamma_N^{x_N}$ can be transformed into the form (7.1), with perhaps an overall minus sign: we use the anticommutation of the Γ_I 's and, if $x_I = 1$, the fact that $\Gamma_I^2 = 1$. This results in

$$\Gamma_I \cdot e_{\vec{x}} = \pm \Gamma_1^{x_1} \dots \Gamma_{I-1}^{x_{I-1}} \Gamma_I^{1-x_I} \Gamma_{I+1}^{x_{I+1}} \dots \Gamma_N^{x_N}. \tag{7.2}$$

The sign is +1 if the number of J with $x_J = 1$ and $J < I$ is even, and is -1 otherwise. If we define $(-1)^{|\vec{x} < I|}$ to be that sign, and define $\vec{x} \boxplus I$ to be the vector $(x_1, \dots, x_{I-1}, 1 - x_I, x_{I+1}, \dots, x_N)$, then this equation becomes

$$\Gamma_I \cdot e_{\vec{x}} = (-1)^{|\vec{x} < I|} e_{\vec{x} \boxplus I}. \tag{7.3}$$

We will also need the function $\text{wt}(\vec{x})$, which equals the number of 1's in \vec{x} .

Thus, $\text{Cl}(0, N)$ is a representation of the Clifford algebra $\text{Cl}(0, N+1)$. It corresponds to an Isoscalar supermultiplet, by replacing $e_{\vec{x}} \mapsto \phi_{\vec{x}}(\tau)$ when the weight of \vec{x} is even, and $e_{\vec{x}} \mapsto \psi_{\vec{x}}(\tau)$ when the weight of \vec{x} is odd. Following the construction of the Isoscalar supermultiplet, we define \mathbb{L}_I and \mathbb{R}_I to be

$$\mathbb{L}_I \psi_{\vec{x}} = (-1)^{|\vec{x} < I|} \phi_{\vec{x} \boxplus I}, \quad \mathbb{R}_I \phi_{\vec{x}} = (-1)^{|\vec{x} < I|} \psi_{\vec{x} \boxplus I}, \tag{7.4}$$

and thus define

$$Q_I \psi_{\vec{x}} = (-1)^{|\vec{x} < I|} \phi_{\vec{x} \boxplus I}, \quad Q_I \phi_{\vec{x}} = (-1)^{|\vec{x} < I|} i \partial_{\tau} \psi_{\vec{x} \boxplus I}. \tag{7.5}$$

The Adinkra for this is an N -dimensional cube. To see this, we note that the basis elements are labeled by $\vec{x} \in \{0, 1\}^N$, the vertices of the N -dimensional cube. The edges colored I connect \vec{x} to $\vec{x} \boxplus I$, which changes the I^{th} coordinate. The supermultiplet specified by equations (7.5) was called the ‘‘base bosonic Clifford Algebra superfield’’ in [4]. To emphasize its chromotopology, we will refer to this as the ‘‘Colored N -cube Clifford supermultiplet’’.

7.2 The $N = 4, D_4$ projection

In considering Adinkras that are not cubes, but rather quotients of cubes, it is useful to first consider a few examples. First, we consider the $N = 4$ example obtained by quotienting a four-dimensional cube by identifying antipodal points. This Adinkra was first described in Ref. [4], where it was identified as the dimensional reduction of the $D = 4$ chiral superfield. Here, we present this example in a way that will motivate the general construction to quotient cubes.

7.2.1 One D_4 projection

Consider the Clifford representation $\text{Cl}(0, 4)$. Define the element

$$g = \Gamma_1 \Gamma_2 \Gamma_3 \Gamma_4. \tag{7.6}$$

Define the two linear transformations $\pi_+, \pi_- : \text{Cl}(0, N) \rightarrow \text{Cl}(0, N)$ to be

$$\pi_{\pm}(v) = v \cdot \frac{1 \pm g}{2}, \quad v, \pi_{\pm}(v) \in \text{Cl}(0, N). \tag{7.7}$$

Note that in the definition (7.7), we multiply v by a factor on its right. This is important. It implies that for every Γ_I and $v \in \text{Cl}(0, N)$, we have that $\Gamma_I(\pi_{\pm}(v)) = \pi_{\pm}(\Gamma_I(v))$.

The fact that g is even means that π_{\pm} preserves the bosonic and fermionic statistics. The fact that $g^2 = 1$ implies that $\pi_+^2 = \pi_+$, $\pi_-^2 = \pi_-$, $\pi_+ \pi_- = \pi_- \pi_+ = 0$, and $\pi_+ + \pi_- = \mathbf{1}$. This means that π_+ and π_- are a complete set of projection operators, so that $\text{Cl}(0, N) = \text{Im}(\pi_+) \oplus \text{Im}(\pi_-)$ as Clifford representations, and π_+ and π_- project onto their corresponding components. Neither of these components are zero, since $(1 + g)/2$ and $(1 - g)/2$ are themselves non-zero elements of $\text{Cl}(0, N)$, which are in $\text{Im}(\pi_+)$ and $\text{Im}(\pi_-)$, respectively.

This kind of projection is nothing new: the matrix g is, up to a scalar factor, the matrix known as γ_5 in four-dimensional field theory, and the projection π_{\pm} corresponds to the familiar projection to chiral spinors: the left- and right-handed halves of the Dirac spinor.

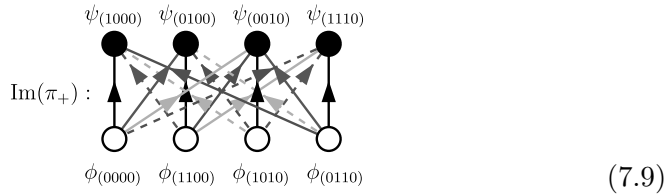
It is also true that $\ker(\pi_+) = \text{Im}(\pi_-)$ and vice-versa, so that we can also describe these representations in terms of constraints: as “ v such that $v \cdot (1 \mp g) = 0$ ”. Then we can realize the Clifford representation as a subspace of $\text{Cl}(0, N)$, rather than as a quotient. This accords with the idea that $\text{Cl}(0, N)$, being a representation for $\text{Cl}(0, N+1)$, decomposes as a direct sum into irreducibles.

We take the standard basis for $\text{Cl}(0, 4)$ mentioned above, $\{e_{\vec{x}} : \vec{x} \in \{0, 1\}^N\}$, and apply π_+ (resp. π_-) on it. The result spans $\text{Im}(\pi_+)$ (resp. $\text{Im}(\pi_-)$), but there are duplications (up to sign). For instance, in $\text{Im}(\pi_+)$, a vertex $\pi_+(e_{\vec{x}})$ and $\pi_+(e_{\vec{x}})g$ will be identified. Since, up to an overall sign,

$$\pi_+(e_{(x_1, x_2, x_3, x_4)}) \cdot g \propto \pi_+(e_{(1-x_1, 1-x_2, 1-x_3, 1-x_4)}), \tag{7.8}$$

vertices are accordingly identified pairwise.

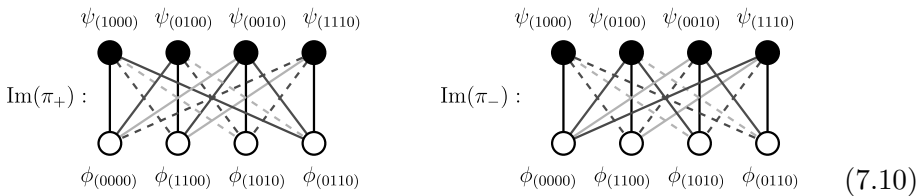
Constructing the Isoscalar supermultiplet from this, we get the following Adinkra:



The result (7.9) is the four-dimensional cube with opposite corners identified, as in [4]. This Adinkra is a quotient of the four-dimensional cube by the code d_4 generated by 1111. This corresponds to the fact that by either doing nothing, or by reversing all four bits of a vertex, we return to the same vertex (with perhaps a minus sign). In reference to the code name, the topology of the four-dimensional cube with opposite corners identified will be called D_4 .

7.2.2 The two inequivalent D_4 quotients

Similarly, we can find the Adinkra for the image of π_- , using $1 - \Gamma_1\Gamma_2\Gamma_3\Gamma_4$ instead. When we do so, we see that the image of π_- and of π_+ look similar — indeed, they have the same topology, the 4-cube with opposite corners identified. But they have different patterns of dashed edges. These patterns cannot be made to coincide even when we redefine some of the vertices by replacing them with their negatives. There are, in fact, two distinct irreducible representations of $Cl(0, 5)$, and these are the two. Nevertheless, we can easily describe the relationship between them: by replacing $\Gamma_4 \mapsto -\Gamma_4$, for instance, which corresponds to replacing Q_4 with $-Q_4$. This, in turn, results in reversing the sign associated to each edge with $I = 4$. The Adinkras for these two are as follows:



We have suppressed the directions of the arrows: they are assumed to always point upward. The replacement of Q_4 by $-Q_4$ is seen in the two Adinkras

above by noting that all solid blue lines in the left hand Adinkra are replaced by dashed blue lines in the in the right hand Adinkra (and vice versa).

Since the dimensional reduction of the $N = 1$ chiral superfield in dimension $D = 4$ down to dimension $D = 1$ results in the image of π_+ , Readers might be tempted to think that the image of π_- arises from the dimensional reduction of the $N = 1$, $D = 4$ antichiral superfield. This, however, is not the case. Rather, it distinguishes different ways of reconstituting the Q_1, \dots, Q_4 into Q_α and $Q_{\dot{\alpha}}^\dagger$, with $\alpha, \dot{\alpha} = 1, 2$, as usual. To go from one to the other Adinkra (7.10), we must complex conjugate not both of the components of Q_α , but only *half* of them — which is impossible without violating Lorentz symmetry in four dimensions. If we complex conjugate all of Q_α , we do swap chiral with antichiral superfields, but we reverse *both* Q_3 and Q_4 . This change can be reversed by a redefinition of the real component fields, which swaps their complex combinations into the complex conjugates. Graphically, the action of complex conjugation would require that *two* colors must be used to implement the dashed/solid exchanges. So for example, both solid blue lines *and* solid green lines in the left hand Adinkra are replaced by dashed blue lines *and* dashed green lines in the in the right hand Adinkra (and vice-versa).

The distinction between the two non-isomorphic supermultiplets depicted by the Adinkras (7.10) is thus more subtle. In fact, in more than 2-dimensional spacetimes, the Q_I 's are not Lorentz-invariant, and the sign of only one of them cannot be changed without violating Lorentz symmetry. In two-dimensional spacetime the supermultiplets depicted in (7.10) are called chiral and twisted-chiral [27], and we adopt this nomenclature also for the worldline $N = 4$ supersymmetry. In $(2, 2)$ -supersymmetric theories in 2-dimensional spacetime, the transformation between the two supermultiplets (7.10) has been identified [29, 30] as the root of mirror symmetry [31–33].

Now, any Lagrangian term involving only one of these types of supermultiplets can just as well be written in terms of only the other type; in this sense they may be regarded as equivalent. However, these two supermultiplets may well mix in a Lagrangian, and in a way that prevents rewriting the Lagrangian in terms of only one or the other type of supermultiplet, as has been done in [27]. This feature makes the two representations of supersymmetry, corresponding to two distinct irreducible representations of $\text{Cl}(0, 5)$ and depicted by the Adinkras (7.10), *usefully distinct*.

Note that such two non-isomorphic irreducible representations of $\text{Cl}(0, N + 1)$ exist precisely when $N = 0 \pmod{4}$, according to table 5, discussed in Section 7.6.

7.3 Projecting twice: the D_6 isoscalar supermultiplet

For $N = 6$, define the two elements

$$g_1 = \Gamma_1\Gamma_2\Gamma_3\Gamma_4, \tag{7.11}$$

$$g_2 = \Gamma_3\Gamma_4\Gamma_5\Gamma_6. \tag{7.12}$$

As before, $g_1^2 = g_2^2 = 1$. Also note that g_1 and g_2 commute, and in fact,

$$g_1g_2 = g_2g_1 = -\Gamma_1\Gamma_2\Gamma_5\Gamma_6. \tag{7.13}$$

Analogously to the D_4 example, we define the four projection operators

$$\pi_{1\pm}(v) = v \cdot \frac{1 \pm g_1}{2}, \tag{7.14}$$

$$\pi_{2\pm}(v) = v \cdot \frac{1 \pm g_2}{2} \tag{7.15}$$

and note that since g_1 and g_2 commute, so do π_{1+} and π_{2+} . We will now project twice: once using π_{1+} (or π_{1-}) and then again, using π_{2+} (or π_{2-}) — a total of four choices. Using π_{1+} and π_{2+} (for instance) produces the Clifford representation $\text{Im}(\pi_{1+} \circ \pi_{2+})$. In this, and in what follows, we will use π_{1+} and π_{2+} for notational definiteness, but it is to be understood that these may be replaced by π_{1-} or π_{2-} , respectively and independently, *mutatis mutandis*.

The composition is

$$(\pi_{1+} \circ \pi_{2+})(v) = v \cdot \frac{1 + g_2}{2} \cdot \frac{1 + g_1}{2} = \frac{1}{4}v \cdot (1 + g_1 + g_2 + g_2g_1) \tag{7.16}$$

$$= \frac{1}{4}v \cdot (1 + \Gamma_1\Gamma_2\Gamma_3\Gamma_4 + \Gamma_3\Gamma_4\Gamma_5\Gamma_6 - \Gamma_1\Gamma_2\Gamma_5\Gamma_6). \tag{7.17}$$

It is straightforward to prove that $\text{Im}(\pi_{1+} \circ \pi_{2+}) = \text{Im}(\pi_{1+}) \cap \text{Im}(\pi_{2+})$ and that this is a Clifford representation. Writing this as $\ker(\pi_{1-}) \cap \ker(\pi_{2-})$, we can see that for all v in this image, $v = v \cdot g_1 = v \cdot g_2 = v \cdot g_1g_2$. In particular, when g_1 and g_2 commute with v , we have that $v = g_1 v = g_2 v = g_1g_2 v$.

Define

$$e_0 := (\pi_{1+} \circ \pi_{2+})(1) = \frac{1}{4}(1 + \Gamma_1\Gamma_2\Gamma_3\Gamma_4 + \Gamma_3\Gamma_4\Gamma_5\Gamma_6 - \Gamma_1\Gamma_2\Gamma_5\Gamma_6). \tag{7.18}$$

Note that e_0 commutes with g_1 and g_2 , and so $e_0 = g_1e_0 = g_2e_0 = g_1g_2e_0$. We successively apply the various $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5$ to e_0 on the left and we get a collection of 16 fields that span $\text{Im}(\pi_{1+} \circ \pi_{2+})$, corresponding to vectors

of the form $e_{(x_1, x_2, x_3, 0, x_5, 0)}$. Applying Γ_4 will not generate any new vectors because $\Gamma_4 = g_1\Gamma_1\Gamma_2\Gamma_3$, and so for every $e_{\vec{x}}$,

$$\Gamma_4 e_{\vec{x}} = g_1\Gamma_1\Gamma_2\Gamma_3 \cdot e_{\vec{x}} \tag{7.19}$$

$$= g_1\Gamma_1\Gamma_2\Gamma_3\Gamma_1^{x_1} \cdots \Gamma_N^{x_N} \cdot e_0 \tag{7.20}$$

$$= \pm\Gamma_1\Gamma_2\Gamma_3\Gamma_1^{x_1} \cdots \Gamma_N^{x_N} g_1 \cdot e_0 \tag{7.21}$$

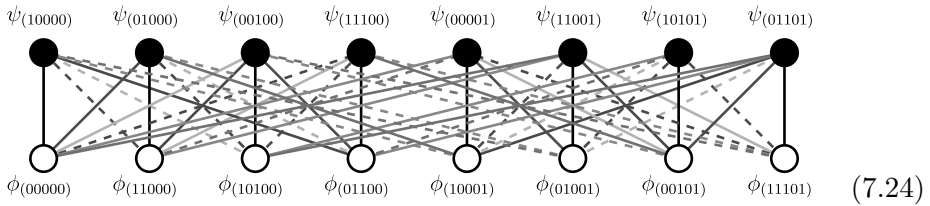
$$= \pm\Gamma_1\Gamma_2\Gamma_3\Gamma_1^{x_1} \cdots \Gamma_N^{x_N} \cdot e_0 \tag{7.22}$$

$$= \pm\Gamma_1\Gamma_2\Gamma_3 \cdot e_{\vec{x}}. \tag{7.23}$$

Similarly, Γ_6 will not generate new vectors, using $\Gamma_6 = g_2\Gamma_3\Gamma_4\Gamma_5$.

The Isoscalar supermultiplet corresponding to this will have 8 bosons and 8 fermions. The $e_{\vec{x}}$ vectors correspond to the various $\phi_{\vec{x}}$ and $\psi_{\vec{x}}$.

If we do this, we get an Adinkra whose topology we call D_6 , which is a six-dimensional cube projected twice: once according to $g_1 = \Gamma_1\Gamma_2\Gamma_3\Gamma_4$ and then according to $g_2 = \Gamma_3\Gamma_4\Gamma_5\Gamma_6$:



Here the colors are as before, with orange for Q_5 and purple for Q_6 .

The name D_6 derives from the code d_6 , generated by the $k = 2$ codewords $c_1 = 111100$ and $c_2 = 001111$; to denote this, we assemble the generator codewords as a matrix, called a *generator matrix* in coding theory:

$$\begin{bmatrix} 111100 \\ 001111 \end{bmatrix}, \tag{7.25}$$

but here we simply regard it as a collection of the codewords listed in its rows. The first codeword corresponds to $\Gamma_1\Gamma_2\Gamma_3\Gamma_4$, the second with $\Gamma_3\Gamma_4\Gamma_5\Gamma_6$, and the sum of these codewords modulo 2, 110011, corresponds to $-\Gamma_1\Gamma_2\Gamma_5\Gamma_6$.

More generally, addition in the code turns into multiplication of the corresponding products of Γ_I matrices, perhaps with a minus sign, because the Γ_I can be anticommutated past each other, and when a particular Γ_I appears in both g_1 and g_2 , once anticommutation is done so that the two Γ_I 's are adjacent, these two simplify to $\Gamma_I^2 = 1$.

7.4 Constructing an isoscalar supermultiplet from a code

The procedure illustrated in the previous two examples can be generalized to the following construction, first applied to Clifford representations by Dimakis. [34].

Construction 7.1. *Suppose we are given N and a doubly-even code $C \subset (\mathbb{Z}/2)^N$ of length N , given by a generating set $\{c_1, \dots, c_k\} \subset C$. Writing each c_i as $(x_{i1}, \dots, x_{iN}) \in \{0, 1\}^N$, we associate to it $g_i := \Gamma_1^{x_{i1}} \dots \Gamma_N^{x_{iN}}$. For instance, $c_1 = 101100100$ would produce $g_1 = \Gamma_1\Gamma_3\Gamma_4\Gamma_7$.*

The $\text{wt}(c_i)$ being even translates into g_i being even. When $\text{wt}(c_i)$ is even, $\text{wt}(c_i)$ being a multiple of 4 is equivalent to $g_i^2 = 1$. At the end of Section 2 we presented the standard fact that any two elements of the codes are orthogonal (that is, share an even number of 1s). As a result, all of the g_1, \dots, g_k commute with each other. They generate a group G under multiplication that is isomorphic as a group to the code C . The isomorphism is done analogously to converting c_i to g_i , except that a minus sign is sometimes required.⁷

For each g_i we have $\pi_{i\pm} : Cl(0, N) \rightarrow Cl(0, N)$ defined as

$$\pi_{i\pm}(v) = v \cdot \frac{1 \pm g_i}{2}. \tag{7.26}$$

As before, the operators $\pi_{i\pm}$ are homomorphisms. The evenness of g_i implies that $\pi_{i\pm}$ preserves the fermionic and bosonic statistics. The fact that $g_i^2 = 1$ implies that π_{i+} and π_{i-} are projection operators.

The fact that the g_i all commute implies that the π_{i+} and π_{i-} all commute.

The following table summarizes how the properties of the generators of the code relate to the properties of the $\pi_{i\pm}$.

c_i (Codeword)	g_i	$\pi_{i\pm}$ (Projector)	
<i>Even weight</i>	<i>Even</i>	<i>Preserves statistics</i>	(7.27)
<i>weight is 0 (mod 4)</i>	$g_i^2 = 1$	<i>projection</i>	
<i>pairwise orthogonal</i>	<i>commute</i>	<i>commute</i>	

⁷This construction is due to Wood, who used it to classify two-elementary abelian subgroups of the spin groups [35].

We define⁸

$$\pi_C := \pi_{1+} \circ \cdots \circ \pi_{k+}, \tag{7.28}$$

and then as in the previous example,

$$\text{Im}(\pi_C) = \text{Im}(\pi_{1+}) \cap \cdots \cap \text{Im}(\pi_{k+}) = \ker(\pi_{1-}) \cap \cdots \cap \ker(\pi_{k-}) \tag{7.29}$$

is the Clifford representation we want. If we define

$$e_0 = \pi_C(1) = \frac{1 + g_1}{2} \cdots \frac{1 + g_k}{2} = \frac{1}{2^k} \sum_{g \in G} g, \tag{7.30}$$

and then successively apply the various Γ_I on the left to e_0 , we obtain a set of elements of $Cl(0, N)$. Using the Dirac relations (6.1) shows that many of these are the same, up to an overall sign. Furthermore, for all g_i , we have $g_i e_0 = e_0$, and more generally this is true of all elements of G . It therefore follows that if we begin with an element v represented by a d_v -dimensional matrix, the application of π_C results in a quantity representable by a $2^{-k} d_v$ -dimensional matrix.

In fact, this will result in 2^{N-k+1} different elements of $\text{Im}(\pi_C)$, occurring in \pm pairs. If we arbitrarily choose one from each \pm pair, the result is a collection of 2^{N-k} vectors that form a basis for $\text{Im}(\pi_C)$.

7.5 Other examples

Example 7.1. The case $N = 8$ allows the code e_8 , which has the following generator set:

$$e_8 : \begin{bmatrix} 11110000 \\ 00111100 \\ 00001111 \\ 10101010 \end{bmatrix}. \tag{7.31}$$

⁸Strictly speaking, the notation (7.28)–(7.29) should specify for each g_i which sign is being used. For illustrative purposes, herein we only use π_{i+} 's. This choice will not affect chromotopology, but will affect the dashedness of the edges. Precisely which of these 2^k distinct choices in $\pi_{1\pm} \circ \cdots \circ \pi_{k\pm}$ provide (non-)isomorphic representations akin to (7.10) is a question we defer to a subsequent effort.

The corresponding g_i are:

$$g_1 = \Gamma_1\Gamma_2\Gamma_3\Gamma_4, \tag{7.32}$$

$$g_2 = \Gamma_3\Gamma_4\Gamma_5\Gamma_6, \tag{7.33}$$

$$g_3 = \Gamma_5\Gamma_6\Gamma_7\Gamma_8, \tag{7.34}$$

$$g_4 = \Gamma_1\Gamma_3\Gamma_5\Gamma_7. \tag{7.35}$$

These produce the following group elements:

$$g_1g_2 = -\Gamma_3\Gamma_4\Gamma_5\Gamma_6, \quad g_1g_3 = \Gamma_1\Gamma_2\Gamma_3\Gamma_4\Gamma_5\Gamma_6\Gamma_7\Gamma_8, \tag{7.36a}$$

$$g_1g_4 = \Gamma_2\Gamma_4\Gamma_5\Gamma_7, \quad g_2g_3 = -\Gamma_3\Gamma_4\Gamma_7\Gamma_8, \tag{7.36b}$$

$$g_2g_4 = \Gamma_1\Gamma_4\Gamma_6\Gamma_7, \quad g_3g_4 = \Gamma_1\Gamma_3\Gamma_6\Gamma_8, \tag{7.36c}$$

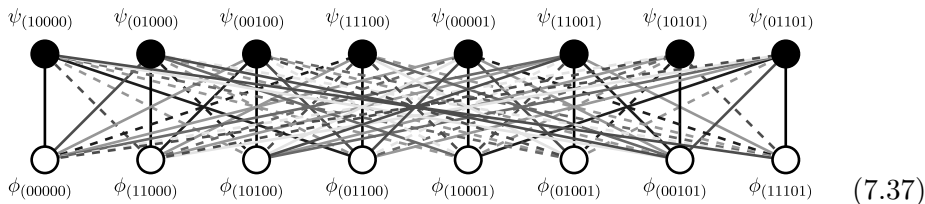
$$g_1g_2g_3 = \Gamma_3\Gamma_4\Gamma_7\Gamma_8, \quad g_1g_2g_4 = -\Gamma_1\Gamma_4\Gamma_6\Gamma_7, \tag{7.36d}$$

$$g_1g_3g_4 = \Gamma_2\Gamma_4\Gamma_6\Gamma_8, \quad g_2g_3g_4 = -\Gamma_1\Gamma_3\Gamma_4\Gamma_8, \tag{7.36e}$$

$$g_1g_2g_3g_4 = \Gamma_1\Gamma_4\Gamma_5\Gamma_8. \tag{7.36f}$$

Recall that each g must be a product of a doubly even number of Γ_I 's, explaining why are we only now seeing g 's with differing numbers of Γ_I 's.

The Adinkra is below. This has the feature that every boson is connected to every fermion. So this is a $K(8,8)$ graph, and is denoted E_8 . This was introduced in [36,37] as the $N = 8$ spinning particle in relation to what was described as a ‘‘supergravity surprise’’. The new color, brown, corresponds to Γ_8 .



As in the $N = 4$ case, there are two irreducible representations, and one is obtained as above, while the other is obtained by reversing the sign on one of the g_i . The result is the same Adinkra topology, but with various signs on the edges reversed. For instance, we can reverse the sign on g_1 by reversing the signs on edges corresponding to Γ_4 . This preserves the sign on $g_2, g_3,$ and g_4 . Reversing various Γ_I yields apparently different Adinkras, but all of these must fall into just two isomorphism classes.

There are $\frac{8!}{1344} = 30$ codes that are permutation equivalent to e_8 .

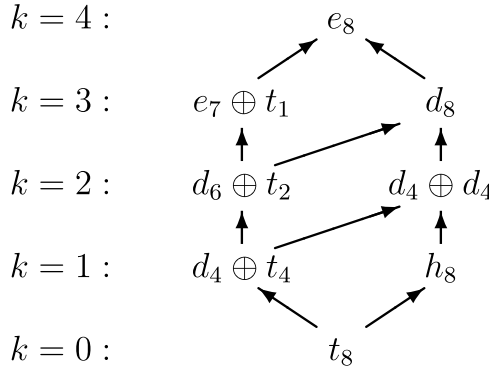
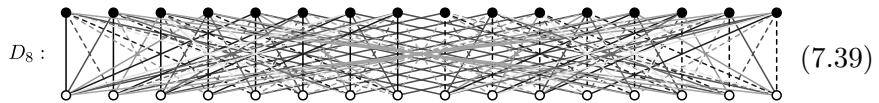
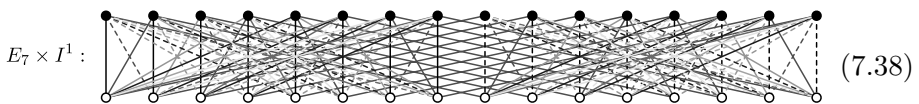


Figure 1: $N = 8$ doubly even codes and their subset relationships: The arrows connecting two codes are injections (possibly after permutation): the code on the lower level is a subcode of the higher one. The trivial code t_8 is at the bottom, and e_8 is the unique maximal doubly even code for $N = 8$ and is drawn at the top. Every doubly even code in $N = 8$ is a subcode of e_8 . The code h_8 is the one generated by 11111111.

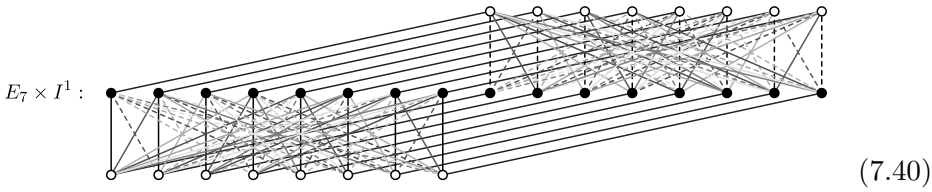
As before, we can take a subgroup of e_8 , but in this case, the situation is a bit more interesting: there are multiple inequivalent choices for which generator to remove. Figure 1 shows all the $N = 8$ doubly even codes, up to permutation equivalence.

For instance, for $k = 3$, we could choose $e_7 \oplus t_1$, which results in an Adinkra topology $E_7 \times I^1$, or we could choose d_8 , which results in a different Adinkra topology D_8 :

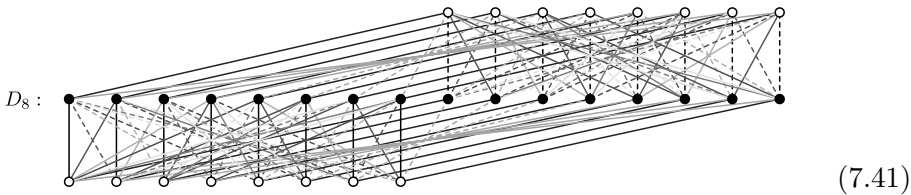


The only difference between these two Adinkras is the way one of the supersymmetries acts, the one represented here by orange edges. In the $E_7 \times I^1$ -Adinkra (7.38), it acts within each of the two halves, leaving the Adinkra *1-color-decomposable*: only the brown edges span the whole Adinkra, and it decomposes into two identical $N = 7$ Adinkras if the brown edges are erased. Each of the halves has the E_7 topology, except that edges of one

color (black) have their dashing reversed. This becomes clearer upon rearranging the nodes a little:



In turn, in the D_8 -Adinkra (7.39), the orange edges span the whole Adinkra together with the brown ones. For the sake of comparison with the $E_7 \times I^1$ -Adinkra (7.40), we also rearrange the nodes of the D_8 -Adinkra:



To decompose this second Adinkra, one would have to erase the edges of at least two colors; we say it is two-color-decomposable. This n -color-decomposability property correlates with the fact that the $e_7 \oplus t_1$ code has one column of zeros, whereas d_8 does not:

$$E_7 \times I^1 \iff e_7 \oplus t_1 : \begin{bmatrix} 0000 & 1111 \\ 0011 & 1100 \\ 0101 & 0101 \end{bmatrix}, \quad vs. \quad D_8 \iff d_8 : \begin{bmatrix} 0000 & 1111 \\ 0011 & 1100 \\ 1111 & 0000 \end{bmatrix}. \tag{7.42}$$

Finally, both Adinkras (7.38) and (7.39) admit precisely one supersymmetry-preserving \mathbb{Z}_2 symmetry: for the former, it is encoded as 11110000 and generated by $\Gamma_1\Gamma_2\Gamma_3\Gamma_4$, while the latter is symmetric with respect to the action of $\Gamma_2\Gamma_4\Gamma_6\Gamma_8$, encoded as 01010101. The projection of each Adinkra by its respective symmetry then produces the E_8 -Adinkra (7.37).

Similar relations exist between the $D_6 \times I^2$ - vs. $D_4 \times D_4$ -Adinkras, and the $D_4 \times I^4$ - versus H_8 -Adinkra corresponding to the $k = 2$ and $k = 1$ rows in the diagram in figure 1.

To summarize, table 4 gives the possible topologies for each N up to $N = 10$. The cartesian product \times refers to taking the cartesian product of the vertex set, and drawing edges between (v_1, w_1) and (v_1, w_2) whenever there is an edge between w_1 and w_2 , and between (v_1, w_1) and (v_2, w_1) whenever there is an edge between v_1 and v_2 . Exponentiation means iterated

Table 4: Adinkra topologies up to $N = 10$: The number of nodes in an Adinkra with the indicated topology is shown to the right of each topology, in the form $(n_B|n_F)$, where n_B is the number of bosons and n_F is the number of fermions in the supermultiplet.

N	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	I (1 1)				
2	I^2 (2 2)				
3	I^3 (4 4)				
4	I^4 (8 8)	D_4 (4 4)			
5	I^5 (16 16)	$D_4 \times I^1$ (8 8)			
6	I^6 (32 32)	$D_4 \times I^2$ (16 16)	D_6 (8 8)		
7	I^7 (64 64)	$D_4 \times I^3$ (32 32)	$D_6 \times I^1$ (16 16)	E_7 (8 8)	
8	I^8 (128 128)	$D_4 \times I^4$ (64 64)	$D_6 \times I^2$ (32 32)	$E_7 \times I^1$ (16 16)	E_8 (8 8)
		H_8 (64 64)	$D_4 \times D_4$ (32 32)	D_8 (16 16)	
9	I^9 (256 256)	$D_4 \times I^5$ (128 128)	$D_6 \times I^3$ (64 64)	$E_7 \times I^2$ (32 32)	$E_8 \times I^1$ (16 16)
		$H_8 \times I^1$ (128 128)	$D_4 \times D_4 \times I^1$ (64 64)	$D_8 \times I^1$ (32 32)	
10	I^{10} (512 512)	$D_4 \times I^6$ (256 256)	$D_6 \times I^4$ (128 128)	$E_7 \times I^3$ (64 64)	$E_8 \times I^2$ (32 32)
		$H_8 \times I^1$ (256 256)	$D_4 \times D_4 \times I^2$ (128 128)	$D_8 \times I^2$ (64 64)	D_{10} (32 32)
			$(D_4 *_1 I^6)^\dagger$ (128 128)	$D_4 \times D_6$ (64 64)	

[†] Herein, the code name “ $D_4 *_1 I^6$ ” denotes that the D_4 code is padded by six zeroes and augmented by one additional, “glue” generator spanning the positions of the six added zeroes and a sufficient number (here, two) of 1’s within D_4 so as to generate a doubly even code; it is generated by {1111 0000 00, 0011 1111 11}.

cartesian products. We note the Adinkra topologies D_4 , D_6 , D_8 , and D_{10} from the codes d_4 , d_6 , d_8 , and d_{10} , as well as E_7 and E_8 from e_7 and e_8 , respectively. In addition, there turns up a code with no standard name, generated by {1111000000, 0011111111}, which we tentatively call $D_4 *_1 I^6$.

7.6 Comparison with Clifford representations

Since Clifford representations are already classified, it is worthwhile comparing what we have just found with the known classification of Clifford representations. The abstract algebras $Cl(0, N+1)$ are known for all N . The following can be found on any standard text on Clifford algebras, such as Lawson and Michelsohn’s *Spin Geometry* [28].

The Clifford algebras $Cl(0, N)$ and $Cl(0, N+1)$ are given in table 5. The description of these algebras has a modulo 8 sort of periodicity, so it is convenient to write $N = 8m + s$, where m and s are integers and $0 \leq s \leq 7$. The notation $\mathbb{R}(n)$, $\mathbb{C}(n)$, and $\mathbb{H}(n)$ denotes the algebra of $n \times n$ matrices with real, complex, and quaternionic coefficients, respectively.

Note from table 5 that every Clifford algebra is either a matrix algebra, or a direct sum of two matrix algebras. It is a classical result that every finite-dimensional real representation of such an algebra decomposes into irreducibles. For $\mathbb{R}(n)$, $\mathbb{C}(n)$, and $\mathbb{H}(n)$, there is up to isomorphism only

Table 5: Clifford algebras $\text{Cl}(0, N)$ decomposed into irreducible representations of $\text{Cl}(0, N+1)$: Here $N = 8m + s$ where m and s are integers and $0 \leq s \leq 7$.

s	$\text{Cl}(0, N)$	$\text{Cl}(0, N+1)$	$\dim_{\mathbb{R}}$ Irrep.	$\dim_{\mathbb{R}}$ $\text{Cl}(0, N)$	$\#$ (Irrep.) in $\text{Cl}(0, N)$	max k
0	$\mathbb{R}(16^m)$	$\mathbb{R}(16^m) \oplus \mathbb{R}(16^m)$	16^m	16^{2m}	16^m	$4m$
1	$\mathbb{R}(16^m) \oplus \mathbb{R}(16^m)$	$\mathbb{R}(2 \cdot 16^m)$	$2 \cdot 16^m$	$2 \cdot 16^{2m}$	16^m	$4m$
2	$\mathbb{R}(2 \cdot 16^m)$	$\mathbb{C}(2 \cdot 16^m)$	$4 \cdot 16^m$	$4 \cdot 16^{2m}$	16^m	$4m$
3	$\mathbb{C}(2 \cdot 16^m)$	$\mathbb{H}(2 \cdot 16^m)$	$8 \cdot 16^m$	$8 \cdot 16^{2m}$	16^m	$4m$
4	$\mathbb{H}(2 \cdot 16^m)$	$\mathbb{H}(2 \cdot 16^m) \oplus \mathbb{H}(2 \cdot 16^m)$	$8 \cdot 16^m$	$16 \cdot 16^{2m}$	$2 \cdot 16^m$	$4m + 1$
5	$\mathbb{H}(2 \cdot 16^m) \oplus \mathbb{H}(2 \cdot 16^m)$	$\mathbb{H}(4 \cdot 16^m)$	$16 \cdot 16^m$	$32 \cdot 16^{2m}$	$2 \cdot 16^m$	$4m + 1$
6	$\mathbb{H}(4 \cdot 16^m)$	$\mathbb{C}(8 \cdot 16^m)$	$16 \cdot 16^m$	$64 \cdot 16^{2m}$	$4 \cdot 16^m$	$4m + 2$
7	$\mathbb{C}(8 \cdot 16^m)$	$\mathbb{R}(16 \cdot 16^m)$	$16 \cdot 16^m$	$128 \cdot 16^{2m}$	$8 \cdot 16^m$	$4m + 3$

one irreducible real representation: \mathbb{R}^n , \mathbb{C}^n , or \mathbb{H}^n , respectively. For $\mathbb{R}(n) \oplus \mathbb{R}(n)$, there are two non-isomorphic irreducible representations: one that ignores the second summand and is the standard representation on the first, and the other that ignores the first summand. Likewise for $\mathbb{C}(n) \oplus \mathbb{C}(n)$ and $\mathbb{H}(n) \oplus \mathbb{H}(n)$.

The fourth column in table 5 counts the real dimension of the irreducible representation. Based on the real dimension of $\text{Cl}(0, N)$, this determines the number of copies of the irreducible representation found in $\text{Cl}(0, N)$, which is shown in column six.

The last column of the table is labeled “max k ”, and refers to the maximal dimension k of a doubly even code. This can be computed using Gaborit’s mass formula, described in Appendix B.1 [38], which gives the number of doubly even codes for a given N and k . This number vanishes when k is larger than the numbers given in the table. To find this value of k , recall that in (1.2) we defined the following function of N :

$$\varkappa(N) := \begin{cases} 0 & \text{for } N < 4, \\ 1 & \text{for } N = 4, 5, \\ 2 & \text{for } N = 6, \\ 3 & \text{for } N = 7, \\ 4 + \varkappa(N-8) & \text{for } N \geq 8, \text{ recursively.} \end{cases} \tag{7.43}$$

Note that if we write $N = 8m + s$ where $0 \leq s \leq 7$, then

$$\varkappa(N) = \begin{cases} 4m & \text{for } s = 0, 1, 2, \text{ or } 3, \\ 4m + 1 & \text{for } s = 4 \text{ or } s = 5, \\ 4m + 2 & \text{for } s = 6, \\ 4m + 3 & \text{for } s = 7. \end{cases} \tag{7.44}$$

and Gaborit’s mass formula implies that for any doubly even code of length N and dimension k , $0 \leq k \leq \varkappa(N)$.

Suppose we take an N -dimensional cubical Adinkra. It has 2^N nodes. If we quotient by a doubly even code of dimension k , so that the code has 2^k elements, we have 2^{N-k} nodes in the quotient Adinkra. The minimal Adinkra then occurs when k is as large as possible; that is, for $k = \varkappa(N)$. Thus, a minimal Adinkra for a given N has $2^{N-\varkappa(N)}$ nodes: half bosons, and the other half of them fermions. In turn, this gives rise to a representation of the Clifford algebra, which in general will be some number of copies of the irreducible representations. The fact that the real dimension of the irreducible representation, found in column 4, coincides with the real dimension of the maximally quotiented Adinkra, $2^{N-\varkappa(N)}$, indicates that the irreducible representations of the Clifford algebra can actually be obtained by quotienting by maximal codes. In particular, the corresponding isoscalar multiplet has a description in terms of Adinkras.

Incidentally, it may concern the reader that this table shows that there is a unique irreducible representation for the Clifford algebra when N is not a multiple of 4, and that there are precisely two irreducible representations for the Clifford algebra when N is a multiple of 4. This seems strange in contrast to the multitude of codes of maximal k in table 4. This indicates that for Isoscalar Adinkras at least, there are hidden isomorphisms between these supermultiplets. The one-hooked versions, however, have no isomorphisms, and so are immune to this problem. This issue will be investigated in greater detail in a future work.

7.7 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 C , 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 C . This immediately implies that the Adinkra has 2^{N-k} nodes, where k is the dimension of C . Of these, 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

$$d_B = d_F = 2^{N-k-1}. \tag{7.45}$$

In [17], 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

$$[v] = [v_*] - \frac{1}{2} \text{dist}(v, v_*), \quad (7.46)$$

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 every doubly even code C , use the Construction above to find an Adinkra. Then choose one bosonic field and create a one-hooked Adinkra hooked on that field. Call this Adinkra \mathcal{A}_C . In this way, we have assigned to every doubly even code C an Adinkra \mathcal{A}_C . Now given two different doubly even codes C_1 and C_2 , we have two distinct Adinkras, \mathcal{A}_1 and \mathcal{A}_2 . They are not isomorphic as chromotopologies because it is a different set of sequences of colors that take us from a vertex to itself in both cases. But because these are one-hooked, we can say something stronger: the supermultiplets they describe will not be isomorphic.

Theorem 7.1. *Let C_1 and C_2 be different doubly even codes, and suppose \mathcal{A}_1 and \mathcal{A}_2 are one-hooked connected Adinkras with codes C_1 and C_2 , respectively. Then the supermultiplets from \mathcal{A}_1 and from \mathcal{A}_2 are not isomorphic.*

Proof. For suppose there were such an isomorphism. Suppose ϕ_1 is the hooked bosonic field for \mathcal{A}_1 and ϕ_2 is the hooked bosonic field for \mathcal{A}_2 . Any isomorphism preserves the number of degrees of freedom in each mass dimension, and since there are no fields with mass dimension less than $[\phi_1]$ or $[\phi_2]$, we must have $[\phi_1] = [\phi_2]$. Any isomorphism must send some linear combination of component fields of \mathcal{A}_1 and their derivatives to ϕ_2 . But ϕ_1 is of lowest mass dimension, and derivatives increase mass dimension. So the only candidate for what must be sent to ϕ_2 is ϕ_1 perhaps multiplied by a unit-less constant c .

The fact that an isomorphism commutes with Q_I means that for each I , $cQ_I\phi_1$ gets sent to $Q_I\phi_2$. We continue in this fashion, until the isomorphism of supermultiplets produces an isomorphism of the corresponding chromotopologies.

By Proposition 4.2, the code is an invariant of the chromotopology, and therefore, this isomorphism of chromotopologies implies $C_1 = C_2$. \square

One final remark about one-hooked Adinkras: for such Adinkras, we can examine how many component fields are in each engineering degree. By

equation (7.46), this is equivalent to finding how many vertices are of a given distance from the highest vertex, v_* . The corresponding notion in coding theory is the “coset weight enumerator” [19]. 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.

8 Application to four- and higher-dimensional theories

We expect to find, among $D = 1$ off-shell theories, the dimensional reductions of the off-shell theories of higher-dimensional supersymmetric theories. For instance, off-shell theories in four dimensions with \mathcal{N} -extended supersymmetry will dimensionally reduce to one-dimensional theories with $N = 4\mathcal{N}$ supersymmetries, since the smallest irreducible spinor in four-dimensional spacetime has $N = 4$ components. In higher dimensions the smallest irreducible spinors have dimension that is an integral multiple of this.

Multiplets with minimal numbers of degrees of freedom are of interest, not only because they presumably provide the simplest theories, but also because they give us an idea of a lower bound to the size of a theory. Minimal multiplets correspond to the irreducible representations of the Clifford Algebra, which are described in table 5. As was pointed out in Section 7.6, they also correspond to maximal codes for each N , and these are listed in table 3 for $N \leq 32$. In any case, the number of component fields is $2^{N-\varkappa(N)}$ so that the number of bosons is $2^{N-\varkappa(N)-1}$ and likewise for the number of fermions.

The results are shown in table 6.

Table 6 can be used to make an argument about the size of the smallest irreducible off-shell representation for a given value of \mathcal{N} . If the smallest indecomposable representation coincides with the smallest irreducible representation, then for a given value of \mathcal{N} , the table above determines the smallest off-shell representation. For each value of \mathcal{N} , it is possible to consider the representation that appears at the lowest level of that column. All of these topologies are shown together with the number of bosonic and fermionic nodes in table 6.

For the cases of $\mathcal{N} = 1$ and 2, the number of bosonic and fermionic degrees of freedom are in agreement with the known minimal off-shell

Table 6: Minimal off-shell 4D, $\mathcal{N} \leq 8$ supermultiplets.

\mathcal{N}	N	#(Bosons)	#(Fermions)	#(Adinkra topologies)
1	4	4	4	1 D_4
2	8	8	8	1 E_8
3	12	64	64	2 $D_{12}, E_8 \times D_4$
4	16	128	128	2 $E_{16}, E_8 \times E_8$
5	20	1,024	1,024	10 ...*
6	24	2,048	2,048	9 ...*
7	28	16,384	16,384	151 ...*
8	32	32,768	32,768	85 ...*

*For $\mathcal{N} \geq 5$ there are too many Adinkra topologies to be shown here; see http://www.rlmiller.org/de_codes/ for an up-to-date table with links to actual codes.

supersymmetrical representations. The smallest 4D, $\mathcal{N} = 1$ off-shell representations do indeed consist of 4 bosons and 4 fermions. In a similar manner, the smallest 4D, $\mathcal{N} = 2$ off-shell representations do indeed consist of 8 bosons and 8 fermions. The case of 4D, $\mathcal{N} = 3$ off-shell representations is not so widely known. Nevertheless, Siegel has presented an argument about the off-shell structure of conformal 4D, $\mathcal{N} = 3$ supergravity that indicates it describes 64 bosons and 64 fermions [39]. There exist, also, one known off-shell example of a 4D, $\mathcal{N} = 4$ supermultiplet in Salam–Strathdee superspace. It is the conformal 4D, $\mathcal{N} = 4$ supergravity supermultiplet field strength [40] and it consists of precisely 128 bosons and 128 fermions. All of this agrees with the first four ‘data’ points on table 6.

Precisely when N is a multiple of 8, the maximum value of k is $N/2$, and these codes are self-dual (where the orthogonal space of the code equals the code; doubly even implies that these codes are self-orthogonal). Thus, these relate to even unimodular lattices. Indeed, the case $\mathcal{N} = 2$, or $N = 16$, provides the two lattices, well known to string theorists: $E_8 \times E_8$, and $SO(32)$, which we call E_{16} , since D_{16} fits in the sequence of codes D_{2n} such that $D_{16} \subset E_{16}$:

$$D_{16} \iff d_{16} : \begin{bmatrix} 1111 & 0000 & 0000 & 0000 \\ 0011 & 1100 & 0000 & 0000 \\ 0000 & 1111 & 0000 & 0000 \\ 0000 & 0011 & 1100 & 0000 \\ 0000 & 0000 & 1111 & 0000 \\ 0000 & 0000 & 0011 & 1100 \\ 0000 & 0000 & 0000 & 1111 \end{bmatrix} \text{ versus}$$

$$E_{16} \iff e_{16} : \begin{bmatrix} 1111\ 0000\ 0000\ 0000 \\ 0011\ 1100\ 0000\ 0000 \\ 0000\ 1111\ 0000\ 0000 \\ 0000\ 0011\ 1100\ 0000 \\ 0000\ 0000\ 1111\ 0000 \\ 0000\ 0000\ 0011\ 1100 \\ 0000\ 0000\ 0000\ 1111 \\ 0101\ 0101\ 0101\ 0101 \end{bmatrix}. \tag{8.1}$$

It is of interest to consider the final case above, with $\mathcal{N} = 8$, or $N = 32$ supersymmetries. For this case we have $2^{N-16} = 2^{16} = 65,536$ total nodes. The Adinkra associated with this topology has 32,768 bosonic nodes and 32,768 fermionic nodes. By very different arguments [37], 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.

There are 85 different maximal codes in $N = 32$ (up to permutation equivalence). The attempt to classify these began with Conway and Pless [41] in 1980, though a correction was found to be necessary by Conway *et al.*, in 1990 [20]. Bilous and van Rees [26] in 2007 replicated these results by performing a more systematic search and provided the list of all 85 codes on the web site [42]. It is amusing to note that this was achieved not long ago, that $N = 32$ is the upper limit of what is known currently about self-dual doubly even codes, and that 32 is the maximal N needed in applications to superstrings and their M - and F -theory extensions. Five of these 85 codes have minimal weight 8, and it would be interesting to see if these five play a special role in $\mathcal{N} = 8$ supersymmetries in four dimensions.

9 Toward a classification of Adinkras and supermultiplets

An Adinkra is determined by its chromotopology, a hanging of the vertices, and a choice of which edges are dashed. 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 code. Conversely, every doubly even code can be used to form an Adinkra. As per Definition 3.1, to go from this to specifying an Adinkra would require a choice of orientations for all the edges and a choice of dashing of each edge.

Concerning orienting the edges, we have discussed in this paper two such choices that are always available: Isoscalar Adinkras and one-hooked

Adinkras. More generally, in [17], we have shown that the choices in orienting edges correspond to the ways of “hanging” the vertices of the graph at various heights, subject to certain conditions. The Isoscalar Adinkra results from hanging the Adinkra on all of the bosons, which are placed all at the same level. The one-hooked Adinkra results from hanging the Adinkra from a single boson.

So if we describe the various ways of dashing the edges (an effort which will be described in another paper), would this then 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 even for these supermultiplets, Adinkras turn out to be useful in their description. [15] Second, we might ask whether two Adinkras may describe the same supermultiplet. But if we insist that the Adinkras are one-hooked, then the corresponding representations are isomorphic if and only if they have the same code, as we saw in Theorem 7.1. So by one-hooking the Adinkras, we can be sure that these are all different supermultiplets. Even considering these two issues, what is clear is that there is a surprising wealth of supermultiplets in one dimension as N is increased towards $N = 32$.

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 codes in table 3. Entries in this table that are greater than 1 indicate situations where there is more than one permutation class of doubly even codes that are possible for that value of N and k . This is true even if we restrict our attention to minimal supermultiplets, which have maximal k , when $N \geq 10$. Note that as N increases, this degeneracy increases, certainly for k non-maximal, but also for k maximal. This clearly illustrates that this “degeneracy” amongst even the minimal supermultiplets grows extremely fast with N .

Acknowledgments

The research of S.J.G. is supported by the endowment of the John S. Toll Professorship, the University of Maryland Center for String & Particle Theory, and 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, as well as the Department of Physics, University of Central Florida, Orlando FL, and the Physics Department of the Faculty of Natural Sciences of the University of Novi Sad, Serbia,

for recurring hospitality and resources. The work described in Appendix B uses computer facilities supported by the National Science Foundation grant no. DMS-0555776. Some Adinkras were drawn with the aid of the *Adinkramat* © 2008 by G. Landweber. A special thanks goes to Martin Mueller, Erik Duamine, Morteza Zadimoghaddam, Richard Stanley, all at M.I.T., and especially Yan Zhang, Ricard Stanley’s student, for fruitful discussions.

Appendix A Deferred details on supersymmetry and Adinkras

A.1 Real coefficients

Real supermultiplets, $\mathcal{M} = (F_1(\tau), \dots, F_m(\tau))$ consist of real component fields: $(F_A(\tau))^\dagger = F_A(\tau)$, and supersymmetry is assumed to preserve this condition. In this section, we will show that this condition implies that the coefficients c in equation (3.8) and equation (3.10) 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

$$\delta_Q(\epsilon) := -i\epsilon^I Q_I \tag{A.1}$$

must also be real. Applying this to equation (3.8) results in

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

Thus

$$\left(\delta_Q(\epsilon) \phi_A(\tau)\right)^\dagger = i \partial_\tau^{[\phi_A] + \frac{1}{2} - [\psi_B]} \psi_B^\dagger(\tau) c^* \epsilon^I = -i\epsilon^I c^* \partial_\tau^{[\phi_A] + \frac{1}{2} - [\psi_B]} \psi_B(\tau). \tag{A.3}$$

Comparing the right-hand sides of equations (A.2) and (A.3), we find that

$$c^* = c. \tag{A.4}$$

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 $F_1(\tau), \dots, F_{2m}(\tau)$ for component fields. There is a real rescaling of these component fields so that each non-zero coefficient c in equations (3.8) and (3.10) is equal to 1 or -1 .*

Proof. We suppose we have component fields $F_1(\tau), \dots, F_{2m}(\tau)$,⁹ and supersymmetry generators Q_I so that the supersymmetry transformation rules are all of the form

$$Q_I F_A = c \partial_\tau^\lambda F_B, \quad (\text{A.5})$$

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, \dots, v_{2m} to correspond to the component fields F_1, \dots, 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 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 equation (A.5). 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

⁹See footnote in the proof of Theorem 4.3 about the use of this notation.

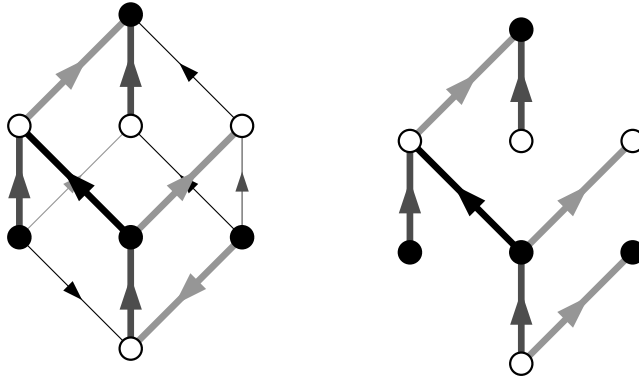


Figure 2: On the left is an $N = 3$ cubical Adinkra. The dashedness of the lines are suppressed for clarity. Some edges (shown with thin lines) can be deleted, until what is left is a tree, called the *spanning tree* (shown at right).

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 2.

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}, \dots, Q_{I_n} . We then apply equation (A.5) iteratively to $F_*(\tau)$, obtaining

$$Q_{I_n} \cdots Q_{I_1} F_*(\tau) = C_i \partial_\tau^\lambda F_i(\tau) \tag{A.6}$$

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 $n \equiv 2, 3$

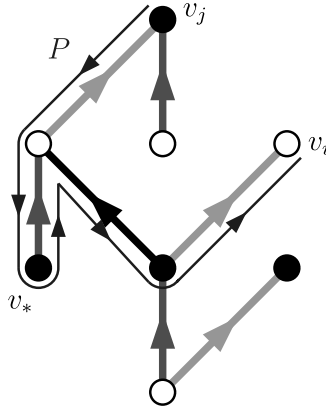


Figure 3: Using the same example as in figure 2, suppose we are considering the vertices v_i and v_j as seen in the figure. The blue path shows a path P in the spanning tree going from v_j to v_* , followed by a path in the spanning tree going from v_* to v_i . One edge is retraced, but this is irrelevant for our purposes. The sequence of colors (black = 1, red = 2, green = 3) along P indicate that we should consider $Q_3Q_1Q_2Q_2Q_3v_j$ which will be a constant multiple of $\partial_\tau^3 v_i$. Because of our rescaling in equation (A.7), we know this constant has absolute value 1.

(mod 4). It is then possible to rescale $F_i(\tau)$ as

$$\tilde{F}_i(\tau) := |C_i| F_i(\tau), \tag{A.7}$$

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:

$$Q_{I_1} \cdots Q_{I_n} F_i = \frac{1}{C_i} \partial_\tau^{n-\lambda} F_*. \tag{A.8}$$

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 3. 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.

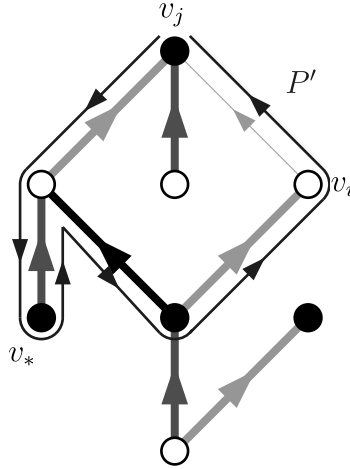


Figure 4: Suppose we consider the orange edge from v_i to v_j , which is in the original Adinkra but not in the spanning tree. We draw that path P from v_j to v_i in the spanning tree as in figure 3, and then tack on the orange edge. The resulting path P' , shown in blue, is a loop from v_j to itself.

If the path P involves the sequence $Q_{J_1} \cdots Q_{J_k}$ of supersymmetry generators, we then use equation (A.6) and equation (A.8) to obtain

$$Q_{J_k} \cdots Q_{J_1} F_j = \frac{C_i}{C_j} \partial_\tau^\mu F_i, \tag{A.9}$$

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

$$Q_I v_i = c \partial_\tau^\lambda v_j \tag{A.10}$$

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}, \dots, 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 (see figure 4).

Using equation (A.10) together with equation (A.9) we get, for some $r \in \mathbb{N}$,

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

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

$$(Q_I Q_{I_n} \cdots Q_{I_1})^2 F_j(\tau) = (-1)^r c^2 \left(\frac{C_i}{C_j}\right)^2 \partial_\tau^{2(\lambda+\mu)} F_j(\tau). \tag{A.12}$$

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

$$\pm (i\partial_\tau)^{n+1} F_j(\tau) = (-1)^r c^2 \left(\frac{C_i}{C_j}\right)^2 \partial_\tau^{2(\lambda+\mu)} F_j(\tau). \tag{A.13}$$

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 equation (A.13) is actually real. We can then match coefficients in equation (A.13) to get

$$c^2 = \pm (-1)^{r+(n+1)/2} \frac{C_j^2}{C_i^2} = \pm 1. \tag{A.14}$$

From this we see that c must have complex absolute value 1. Since c is real, $c = \pm 1$. Furthermore, we also note from equation (A.13) 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 [4]. □

This also proves that Adinkraic supermultiplets with no central charge are *engineerable* [17]: it is possible to assign engineering dimensions to all component fields, consistently with the formulas for the supersymmetry generator action of the form equations (3.2)– (3.5) of the supersymmetry algebra (3.1), and without having to introduce parameters of non-zero engineering dimension in either of these.

Appendix B 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 $[32, k]$ 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 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 [43, 44], 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 [45] 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 [46]. 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` [47], a GAP package for coding theory. GPL implementations for the cases of graphs of any size and codes of length up to 32 (or 64 on many machines) can be found in Sage [48] (`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 B.3. Sections 2 and 4 are essentially a summary of the techniques described in [43, 44], respectively, which provide the full story. The Reader should also be aware of the improvements to this algorithm in the sparse case in [49].

B.1 The mass formula

The following formula, due to Gaborit [38], gives the number $\sigma(N, k)$ of distinct doubly even $[N, k]$ codes, depending on the residue of N modulo 8:

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

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

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,

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

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

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

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

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

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.

B.2 Computing the automorphism group

Computing the automorphism group of a code is closely related to the graph isomorphism problem. Given a 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 $\{1, 2, \dots, N\}$ 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 j th place. Consider the automorphism group $\text{Aut}(G(C))$ 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 $\text{Aut}(G(C))_b$ 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 $\text{Aut}(G)_{\Pi_0}$ 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 $\Pi = (C_1, \dots, C_r)$ and $v \in C_i$, define $R(G, \Pi, v) = E_G(\Pi')$, where $\Pi' = (C_1, \dots, C_{i-1}, \{v\}, C_{i-1} \setminus \{v\}, C_{i+1}, \dots, C_r)$. This defines $R(G, \Pi, v)$ up to reordering of the cells of the partition. For full details, see Ref. [43], 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 $\text{Aut}(G)_{\Pi_0}$ acts on the tree T by taking a sequence of nested partitions Π_0, \dots, Π_k to the resulting nested sequence of partitions by letting $\text{Aut}(G)_{\Pi_0}$ act on the elements of the cells; since this sequence is defined by a sequence v_1, \dots, v_k , the sequence $\gamma(v_1), \dots, \gamma(v_k)$ defines the image under $\gamma \in \text{Aut}(G)_{\Pi_0}$. 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 $\text{Aut}(G)_{\Pi_0}$.

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 $(\{v_1\}, \{v_2\}, \dots, \{v_n\})$, which defines an ordering of the vertices of G . Given two leaves of the tree T , one has two orderings v_1, \dots, v_n and v'_1, \dots, 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 $\text{Aut}(G)_{\Pi_0}$. 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 $\gamma \in \text{Aut}(G)_{\Pi_0}$, 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. [43] and the code in Ref. [48].

The outcome is an algorithm to efficiently compute the group $\text{Aut}(C) = \text{Aut}(G(C))_b$ (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 code, but it also contains an ordering of its words. In practice, $c(C)$ will be output as an actual generator matrix.

B.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 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

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

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

Corollary B.1. *Suppose C is a code spanned by $\{v_1, \dots, v_k\}$, and that $\text{wt}(v_i) \equiv 0 \pmod{4}$ for each $i = 1, \dots, k$. 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) $\text{wt}(v_i + v_j) \equiv 0 \pmod{4}$.

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

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

Then again by (2.2),

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

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 $\left[\begin{array}{ccc} 0 & I_{N-2k} & * \end{array} \right]$, 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 $\left[\begin{array}{ccc} 0 & I_{N-2k} & * \end{array} \right]$, since the sum of two even words is even.

B.4 Exhaustively generating the codes

Here we use another algorithm developed by McKay [44], 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 $\gamma \in \text{Aut}(C)$ 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

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

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 codes of degree 32, isomorph-free.

```

C_0 := the dimension 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 $\text{Aut}(C)$ for free. Thus when we are checking whether $(C', C) \cong (p(C), C)$, we can simply look for an element of $\text{Aut}(C)$ 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. and L. Rana, *A theory of spinning particles for large N -extended supersymmetry*, Phys. Lett. B **352**(1–2) (1995), 50–58, [arXiv:hep-th/9504025].
- [2] S.J. Gates Jr. and L. Rana, *A theory of spinning particles for large N -extended supersymmetry. II*, Phys. Lett. B **369**(3–4) (1996), 262–268, [arXiv:hep-th/9510151].
- [3] S.J. Gates Jr. W.D. Linch and J. Phillips, III, *When superspace is not enough*, [arXiv:hep-th/0211034]
- [4] M. Faux and S.J. Gates Jr., *Adinkras: a graphical technology for supersymmetric representation theory*, Phys. Rev. D **71**(3) (2005), 065002, [arXiv:hep-th/0408004].
- [5] M. Faux, D. Kagan and D. Spector, *Central charges and extra dimensions in supersymmetric quantum mechanics*, [arXiv:hep-th/0406152].
- [6] M. Faux and D. Spector, *Duality and central charges in supersymmetric quantum mechanics*, Phys. Rev. D (3) **70**(8) (2004), 085014, 5, [arXiv:hep-th/0311095].
- [7] M. Faux and D. Spector, *A BPS interpretation of shape invariance*, J. Phys. A **37** (2004), 10397–10407, [arXiv:quant-ph/0401163].
- [8] Z. Kuznetsova, M. Rojas and F. Toppan, *Classification of irreps and invariants of the N -extended supersymmetric quantum mechanics*, J. High Energy Phys. **03** (2006), 098, [arXiv:hep-th/0511274].
- [9] S. Bellucci, S. Krivonos, A. Marrani and E. Orazi, *‘Root’ action for $N = 4$ supersymmetric mechanics theories*, Phys. Rev. D **73** (2006), 025011, [arXiv:hep-th/0511249].
- [10] S. Bellucci, J. Gates, S. James and E. Orazi, *A journey through garden algebras*, Lect. Notes Phys. **698** (2006), 1–47, [arXiv:hep-th/0602259].
- [11] S. Bellucci, S. Krivonos, O. Lechtenfeld and A. Shcherbakov, *Superfield formulation of nonlinear $N = 4$ supermultiplets*, [arXiv:0710.3832 [hep-th]].

- [12] Z. Kuznetsova and F. Toppan, *Refining the classification of the irreps of the 1d n -extended supersymmetry*, Mod. Phys. Lett. A **23** (2008), 37–51, [arXiv:0701225].
- [13] M. Gonzales, M. Rojas and F. Toppan, *One-dimensional sigma-models with $n = 5, 6, 7, 8$ off-shell supersymmetries*, Int. J. Mod. Phys. A **24** (2009), 4317–4326, [arXiv:0812.3042].
- [14] Z. Kuznetsova and F. Toppan, *Extended supersymmetric quantum mechanics of Fierz and Schur type*, [arXiv:1012.1005].
- [15] C.F. Doran, M.G. Faux, S.J. Gates Jr., T. Hübsch, K.M. Iga and G.D. Landweber, *On the matter of $n = 2$ matter*, Phys. Lett. B **659** (2008), 441–446, [arXiv:0710.5245].
- [16] W. Siegel and M. Roček, *On off-shell supermultiplets*, Phys. Lett. B **105**(4) (1981), 275–277.
- [17] C.F. Doran, M.G. Faux, S.J. Gates Jr., T. Hübsch, K.M. Iga and G.D. Landweber, *On graph-theoretic identifications of Adinkras, supersymmetry representations and superfields*, Int. J. Mod. Phys. A **22** (2007), 869–930, [arXiv:math-ph/0512016].
- [18] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, Amsterdam, 1977.
- [19] W.C. Huffman and V. Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, 2003.
- [20] J.H. Conway, V. Pless and N.J.A. Sloane, *The binary self-dual codes of length up to 32: a revised enumeration*, J. Comb. Theory, Series A **60** (1992), 183–195.
- [21] T. Hübsch, *Haploid $(2, 2)$ -superfields in 2-dimensional space-time*, Nucl. Phys. B **555**(3) (1999), 567–628.
- [22] R.Q. Almkahhal and T. Hübsch, *Gauging Yang-Mills symmetries in $(1+1)$ -dimensional space-time*, Int. J. Mod. Phys. A **16**(29) (2001), 4713–4768.
- [23] C.F. Doran, M.G. Faux, S.J. Gates Jr., T. Hübsch, K.M. Iga and G.D. Landweber, *Adinkras and the dynamics of superspace prepotentials*, Adv. Stud. Theor. Phys. (2007, in press), [arXiv:hep-th/0605269].
- [24] C.F. Doran, M.G. Faux, S.J. Gates Jr., T. Hübsch, K.M. Iga and G.D. Landweber, *Super-Zeeman embedding models on N -supersymmetric world-lines*, J. Phys. A **42** (2008), 065402, [arXiv:0803.3434].
- [25] C.F. Doran, M.G. Faux, S.J. Gates Jr., T. Hübsch, K.M. Iga and G.D. Landweber, *Off-shell supersymmetry and filtered Clifford supermodules*, [arXiv:math-ph/0603012].

- [26] R.T. Bilous and G.H.J. van Rees, *An enumeration of binary self-dual codes of length 32*, Des. Codes Cryptogr. **26** (2002), 61–86.
- [27] S.J. Gates Jr., C.M. Hull and M. Roček, *Twisted multiplets and new supersymmetric nonlinear sigma models and new supersymmetric nonlinear sigma models*, Nucl. Phys. B **248** (1984), 157.
- [28] H.B. Lawson and M.-L. Michelsohn Jr., *Spin geometry*, Princeton Mathematical Series, **38**, Princeton University Press, Princeton, NJ, 1989.
- [29] S.J. Gates Jr., *Vector multiplets and the phases of $n=2$ theories in 2d: Through the looking glass*, Phys. Lett. B **352** (1995), 43–49.
- [30] D.R. Morrison and M.R. Plesser, *Towards mirror symmetry as duality for two dimensional abelian gauge theories*, Nucl. Phys. Proc. Suppl. **46** (1996), 177–186, [arXiv:hep-th/9508107].
- [31] S.-T. Yau (Ed.), *Mirror manifolds*, International Press, 1990.
- [32] B. Greene and S.-T. Yau (Eds.), *Mirror manifolds II*, International Press, 1996.
- [33] D.H. Phong, L. Vinet and S.-T. Yau (Eds.), *Mirror manifolds III*, American Mathematical Society, Providence, RI, 1999.
- [34] A. Dimakis, *A new representation of Clifford algebras*, J. Phys. A **22** (1989), 3171–3193.
- [35] J.A. Wood, *Self-orthogonal codes and the topology of spinor groups*, Coding Theory and Design Theory **20** (1990), 219–239.
- [36] S.J. Gates and L. Rana Jr., *Ultramultiplets: a new representation of rigid 2-d, $N=8$ supersymmetry*, Phys. Lett. B **342** (1995), 132–137, [arXiv:hep-th/9410150].
- [37] S.J. Gates Jr., W. Linch, J. Phillips and L. Rana, *The fundamental supersymmetry challenge remains*, Gravit. Cosmol. **8(1–2)** (2002), 96–100, [arXiv:hep-th/0109109].
- [38] P. Gaborit, *Mass formulas for self-dual codes over Z_4 and $F_q + uF_q$ rings*, IEEE Trans. Inf. Theory **42(4)** (1996), 1222–1228.
- [39] W. Siegel, *On-shell $O(N)$ supergravity in superspace*, Nucl. Phys. B **177** (1981), 325.
- [40] E. Bergshoeff, M. de Roo, and B. de Wit, *Extended conformal supergravity*, Nucl. Phys. B **182** (1981), 173.
- [41] J.H. Conway and V. Pless, *On the enumeration of self-dual codes*, J. Comb. Theory, Series A **28** (1980), 26.
- [42] R.T. Bilous and G.H.J. van Rees, Self-dual code lists, <http://www.cs.umanitoba.ca/~umbilou1/SelfDualCodes/toc.html>.
- [43] B. McKay, *Practical graph isomorphism*, Congr. Numer. 30 **30** (1981), 45–87.

- [44] B. McKay, *Isomorph-free exhaustive generation*, J. Algorithms **26** (1998), 306324.
- [45] J.S. Leon, *Permutation group algorithms based on partition, I: theory and algorithm*, J. Symbolic Comput. **12** (1991), 533583.
- [46] B. McKay, *The nauty page*, <http://cs.anu.edu.au/~bdm/nauty/> (2008).
- [47] R. Baart, T. Boothby, J. Cramwinckel, W. Joyner, R. Miller, E. Minkes, E. Roijackers, L. Ruscio, and C. Tjhai, Gap package guava, <http://opensourcemath.org/guava/> (2008).
- [48] W. Stein, *Sage: open source mathematical software* (version 2.10.3), the Sage group, <http://www.sagemath.org> (2008).
- [49] P. Darga, M. Liffiton, K. Sakallah, and I. Markov, *Exploiting structure in symmetry detection for CNF*, Proceedings of 41st Design Automation Conference, 2004, <http://vlsicad.eecs.umich.edu/BK/SAUCY/>