Study of lifting of 1D adinkras to 2D

Kevin Iga and Yan X Zhang

October 8, 2014

1 Definitions

1.1 1-d Adinkras

Let $V(A)$ be the vertices of A.

Recall that a 1-d adinkra A has a rank function $g : V(A) \to \mathbf{Z}$ (making it into a graded poset). Let $V(A, r)$ be the vertices of A with rank r. We will assume 0 to be the lowest rank in the support of g.

Recall that a 1-d adinkra has a code $C(A)$, which is necessarily doubly-even. The code has parameters (n, k), where n is the number of coordinates and k is the dimensions of the code. In this situation, the vertices of A are labeled by cosets $\{0, 1\}^n + C(A)$, so $|V(A)| = 2^{n-k}$.

1.2 2-d Adinkras

If I read Tristan's stuff right, we can completely translate the combinatorial rules to: a *2-d adinkra* (of dimension n) is a finite simple connected graph A such that:

- It is an 1-d adinkra (with the associated ranking, dashing, etc.).
- It has $p + q = n$ colors, where the first p-colors are called "left-moving" and the second q-colors are called "right-moving."
- A coherence condition: for any cycle, we imagine the following sum: going up (here "up" comes from the grading we have from the engineering dimension in our ranking for the 1-d adinkra) a left-handed edge adds -1, and going up a right-handed edge adds 1; going down the edges give contributions with opposite signs. The sum of this around any cycle must be 0. (in particular, this rules out things like ambidextrous bow-ties)

Assuming I interpreted these rules correctly, now I can do combinatorics without needing any physics.

2 Structure Theorems

2.1 Bigrading

The first structural fact we can impose is a bi-grading that is compatible with the grading we already have from the 1-d adinkra structure, in the sense that the 1-d grading is simply one of the coordinates of our bi-grading.

Proposition 2.1. *A* 1-*d adinkra can be extended to a* 2-*d adinkra if and only if the* 1-*d adinkra has a* bigrading *to* $\mathbf{Z}^2$*. This is a map* $g : V \to \mathbf{Z}^2$*, such that all left-moving edges correspond to displacements of* $(0, 1)$ *and right-moving edges correspond to displacements of* $(1, 0)$*.*

Proof. Proof delayed until talking more with Kevin and Tristan about the easiest way to write things up to avoid reinventing wheels. □

This grading allows us to define the notation $V(A, (x, y))$, which is the set of vertices of A of grading $(x, y) \in \mathbf{Z}^2$. We will also use the notation $V(A, r)$ to similar refer to vertices of A of grading $r \in \mathbf{Z}$ when referring to 1-d adinkras, which will come up in our study. Notice that when we consider a 2-d adinkra as a 1-d adinkra, we have the relation

$$V(A, r) = \cup_{x+y=r} V(A, (x, y)).$$

A really important consequence of having a bigrading is that we get to "complete the square" with the following Corollary:

Corollary 2.2. *In a* 2-*d adinkra, suppose we have a path* $(x, y \pm_1 1) \to (x, y) \to (x \pm_2 1, y)$*, where each* $\pm_i$ *corresponds to a choice of sign, the first and the last vertices are connected to* $(x \pm_2 1, y \pm_1 1)$ *via the corresponding colors in a square.*

Proof. Because we have an (1-d) adinkra, the two edges in this path correspond to two different colors (WLOG 1 and 2 in order) respectively, and if we use the colors 2 and 1 in order we must also reach $(x \pm_2 1, y)$ from $(x, y \pm_1 1)$. Because left-moving colors only correspond to y-axis moves in the $\mathbf{Z}^2$ bigrading, and right-moving colors only correspond to x-axis moves, the first move must have displacement $(\pm_2 1, 0)$ and the second move must have displacement $(0, \pm_1)$. This is exactly equivalent to the statement. □

2.2 Rectangle

Let the *support* of a 2-d adinkra (and/or its bigrading function g) be defined as the range of g, its bigrading function. Now, we show that the support of 2-d adinkra must form a rectangle in $\mathbf{Z}^2$. In this and the following sections, it helps to have some standard assumptions:

- Recall that any 1-d adinkra has vertices labeled by equivalence classes of $\{0,1\}$ by some (n,k) doubly-even code L. We will refer to vertices by these equivalence classes (or particular elements of the equivalence classes).

- Recall that we have $n = p + q$ colors. WLOG, let the first p colors be left-moving and the last q colors be right-moving. We will also refer to these colors as elements of $\{1, 2, \ldots, n\}$, thinking of them as the indices of the vertices labeled as elements of $\{0,1\}^n$. So an edge of color i moves a vertex $(v_1 v_2 \cdots v_n)$ by changing the i-th bit to $(1 - v_i)$ and leaving the other vertices intact.

- We will always define $\overline{0}$ to be the vertex corresponding to the equivalence class of $(00 \cdots 0)$. We will also assume that $\overline{0}$ has the coordinate $(0,0)$. (to be precise, this means $g(\overline{0}) = (0,0)$).

For every vertex pair (v, w) in our 2-d adinkra, there exist (many) paths from v to w. Ignoring the dashings for now, let the sequence of colors on any path be called a *color sequence* for the path. So, for example, in a chromotopology corresponding to the unique trivial $(4,0)$ code, the path with color sequence $(3,2,1,1)$ carries $\overline{0} = 0000$ to 0010, 0110, 1110, and finally 0110.

Now, define a map s that takes a color sequence and returns an element of $V = \{0,1\}^n$ where the i-th element is the number of times (modulo 2) that color i appears in the sequence. For example, $s(3,2,1,1) = 0110$. Note that $s(d)$ will return (a member of the equivalence class of) the bitstring obtained by applying the sequence d to $\overline{0}$. In particular that we may permute a sequence from $d = (d_n)$ to $d' = (d'_n)$ without changing the result of s.

The previous work on 1-d adinkras states that if we start at any vertex, following two paths with color sequences d and d' must end up at the same resulting vertex if and only if $s(d) = s(d') \pmod{L}$. Let $l(d)$ be a rearrangement of d that moves all the left-moving colors to the beginning, and let $r(d)$ be a rearrangement of d that moves all the right-moving colors to the beginning. Thus, suppose $p = q = 2$, we have $l((3,2,1,1)) = (2,1,1,3)$ and $r((3,2,1,1)) = (3,2,1,1)$. We always have $s(l(d)) = s(r(d))$ in general, since $l(d)$ and $r(d)$ are just permutations of each other.

Proposition 2.3. *Suppose we have any path from (x,y) to (v,w) in a 2-d adinkra A; then the vertices (x,w) and (v,y) are in A's support.*

Proof. Let this path take color sequence d. $l(d)$ and $r(d)$ must both end up at (v,w), but $l(d)$ only changes along the y-axis in the first part of moves that only use left-moving colors, so it must end up at coordinate (x,w) to be able to end up at (v,w) (since it can only use right-moving / x-axis colors afterwards). Using the same argument for $r(d)$ shows we must end up in (v,y) at some point in the path. □

Corollary 2.4. *The support of a 2-d adinkra is exactly some $k \times l$ rectangle. In other words, the set of points in $\mathbf{Z}^2$ in the range of the bigrading map can be taken to the set of points (x,y), $0 \le x < l$, $0 \le y < k$.*

Proof. If the support is not a rectangle, then there must be some coordinates (x, y) and (v, w) in the support such that one of the other two diagonal coordinates are missing. This violates Proposition 2.3. □

2.3 Factored by the boundary

Now, we show the main classification theorem, which is that a 2-d adinkra is completely controlled by its boundary. Let the *left boundary* A_L and *right boundary* A_R of the rectangle be the subgraphs of A induced by vertices of A that occur at the sets $\{(i, 0)|0 \leq i < l\}$ and $\{(0, j)|0 \leq j < k\}$ respectively. It helps to picture the normal Cartesian plane rotated 45 degrees ounterclockwise, because then adjectives refer to the direction of "movement": this way, the left boundary literally corresponds to the left-most vertices in the support rectangle and vice-versa for the right.

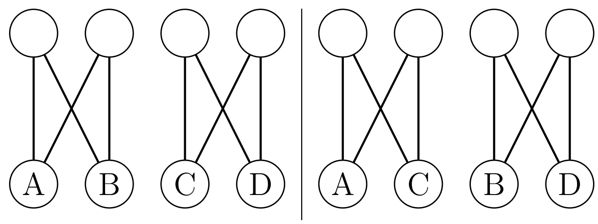

Figure 1: Tensoring the two adinkras here with the following identification gives a non-disconnected adinkra with 16 vertices.

Note that we can have both A_L and A_R be disconnected graphs. An example that we will often return to is displayed in Figure 1. However, we show that the disconnectedness is not really an issue:

Lemma 2.5. *All disconnected components in A_L (and respectively A_R) are isomorphic as graded posets.*

[⋆⋆⋆ TODO ⋆⋆⋆]

Theorem 2.6. *A 2-d adinkra A of type (n, k) is uniquely determined by A_L, A_R, and an identification of $V(A_L, 0)$ and $V(A_R, 0)$.*

Here is another way to factor the adinkra:

Theorem 2.7. *Pick any vertex $v \in V(A, (0, 0))$. Let the connected component of the right boundary that v belongs to be labeled A_R^v; similarly, let the connected component in the left boundary that v belongs to be labeled A_L^v. The adinkra A is uniquely determined by the vertices and edges of A_R^v and A_L^v.*

Proof. Consider the color sequence d of any path from v to a vertex u in A_R^v. Pick any left-moving color c. Examine the sequence cd. Corollary 2.2 will force the cd_1 to end with the same x-axis displacement as d_1, cd_1d_2 to end with the same x-axis displacement as d_1d_2, etc. so inductively any color sequence cd will end with the same x-axis displacement as d. Using induction, we get that for every one of the elements u of A_L^0, the right-moving (x-axis) colors starting at u forms a copy A_R^v whose x-axis positions are forced by the right boundary.

The key observation is that every element u of the adinkra can be reached starting at v via moving to an element of the left boundary and then using only right-moving moves (i.e. by taking $l(d)$ for any color sequence d that goes from v to u). Because of the discussion in our previous paragraph, A_R^v and A_L^v uniquely tell us how to obtain the coordinates of u. □

We will worry less about exact rankings right now with the following construction: for an 1-d adinkra A let $\mathrm{Val}(A)$ be the valise form of A. For any 2-d adinkra A with boundary pair (A_L, A_R), the adinkra defined by the boundary pair $(\mathrm{Val}(A_L), \mathrm{Val}(A_R))$ is again a 2-d adinkra. We will abuse notation and call this new adinkra $\mathrm{Val}(A)$, and call such an adinkra (i.e. where the rectangle is a 2×2 square) a *valise* (2-d) adinkra. We seek the following theorem, which is very similar to Theorem 2.6

Corollary 2.8. *A valise 2-d adinkra A of type (n, k) is uniquely determined by A_L, A_R, and an identification of $V(A_L, 0)$ and $V(A_R, 0)$. Furthermore, $|A_L| = |A_R| = 2^{n-k-1}$.*

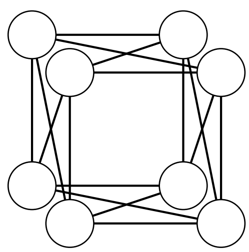

Figure 2: This is a tight valise which cannot be lifted.

Problem 1. What are all the 2-d adinkras A with the same valise adinkra $\mathrm{Val}(A)$? Not all lifts are possible. For example, the adinkra in Figure 2 cannot be lifted to any non-valise form!

3 Quotienting

Define a *even-split doubly-even (ESDE) code* to be a doubly-even code isomorphic to a direct sum $C_L \oplus C_R$ of even codes. It was proved [?] that 1-d adinkras are ismorphic to quotienting the hamming cube H^n by a doubly-even code, so any adinkra A has a well-defined associated code $C(A)$ that is uniquely determined by just the graph structure of the adinkra.

Our goal is to show

Theorem 3.1. *The image of C of the set of 2-d adinkras are exactly the ESDE codes.*

This answers a conjecture of Hübsch [].

To do this we need to show two things:

Proposition 3.2. *Any valise 2-d adinkra A has $C(A)$ equal to a ESDE code.*

Proof. Note that the automorphism group of the adinkra is defined by the basepoint.

TODO □

Proposition 3.3. *Any ESDE code defines a valise 2-d adinkra A.*

Proof. TODO □

Here are some other problems:

Problem 2. Given two valise 1-d adinkras A_L and A_R of equal size, what identifications of $V(A_L, 0)$ and $V(A_R, 0)$ are possible?

Proof. Data: if we have $\{0,1\} \cup \{2,3\}$ on one side, the other side must be $\{0,2\} \cup \{1,3\}$. □

[$\star\star\star$ There are two kinds of quotienting that we can think of: one quotient is directly quotienting the mega hypercube adinkra by a ESDE code; one quotient is given the valise adinkra with associated A_L, A_R each with 2^{d+1} vertices, the necessary d-dimensional quotienting that occurs when we naively tensor the two parts (which gives 2^{2d} vertices in each corner, for 2^{2d+2} total vertices, when in the end we just want 2^{d+2} vertices. $\star\star\star$]

4 ESDE Codes

In this section, we classify all ESDE codes.

[$\star\star\star$ TODO! $\star\star\star$]