CUBOIDS, A CLASS OF CLUTTERS

AHMAD ABDI, GÉRARD CORNUÉJOLS, NATÁLIA GURIČANOVÁ, AND DABEEN LEE

ABSTRACT. The $\tau = 2$ Conjecture, the Replication Conjecture and the f-Flowing Conjecture, and the classification of binary matroids with the sums of circuits property are foundational to Clutter Theory and have far-reaching consequences in Combinatorial Optimization, Matroid Theory and Graph Theory. We prove that these conjectures and result can equivalently be formulated in terms of *cuboids*, which form a special class of clutters. Cuboids are used as means to (a) manifest the geometry behind primal integrality and dual integrality of set covering linear programs, and (b) reveal a geometric rift between these two properties, in turn explaining why primal integrality does not imply dual integrality for set covering linear programs. Along the way, we see that the geometry supports the $\tau = 2$ Conjecture. Studying the geometry also leads to over 700 new ideal minimally non-packing clutters over at most 14 elements, a surprising revelation as there was once thought to be only one such clutter.

CONTENTS

Date: October 16, 2019.

1. Introduction

Let E be a finite set of *elements*, and let $\mathcal{C}$ be a family of subsets of E, called *members*. We say that $\mathcal{C}$ is a *clutter* over *ground set* E if no member is contained in another one [12]. Two clutters are *isomorphic* if one is obtained from the other after relabeling its ground set. A *cover of* $\mathcal{C}$ is a subset of E that intersects every member, and a cover is *minimal* if it does not properly contain another cover. The *set covering polyhedron of* $\mathcal{C}$ is defined as

$$Q(\mathcal{C}) := \left\{x \in \mathbb{R}^E_+ : x(C) \geq 1 \;\forall\, C \in \mathcal{C}\right\}$$

while the *set covering polytope of* $\mathcal{C}$ refers to

$$P(\mathcal{C}) := \left\{x \in [0,1]^E : x(C) \geq 1 \;\forall\, C \in \mathcal{C}\right\}.$$

Here, $x(C)$ is shorthand notation for $\sum_{e\in C} x_e$.

Proposition 1.1 (folklore)**.** *Let $\mathcal{C}$ be a clutter. Then the integral extreme points of $Q(\mathcal{C})$ are precisely the incidence vectors of the minimal covers of $\mathcal{C}$ and the integral extreme points of $P(\mathcal{C})$ are precisely the incidence vectors of the covers of $\mathcal{C}$. Moreover, $Q(\mathcal{C})$ is an integral polyhedron if, and only if, $P(\mathcal{C})$ is an integral polytope.*

We say that $\mathcal{C}$ is *ideal* if the corresponding set covering polyhedron (or polytope) is integral [11]. Consider the primal-dual pair of linear programs

$$(P)\quad \begin{array}{ll} \min & w^\top x \\ \text{s.t.} & x(C) \geq 1 \quad C \in \mathcal{C} \\ & x \geq \mathbf{0} \end{array} \qquad\qquad (D)\quad \begin{array}{ll} \max & \mathbf{1}^\top y \\ \text{s.t.} & \sum (y_C : e \in C \in \mathcal{C}) \leq w_e \quad e \in E \\ & y \geq \mathbf{0}. \end{array}$$

It is well-known that $\mathcal{C}$ is an ideal clutter if, and only if, the primal linear program (P) has an integral optimal solution for all $w \in \mathbb{Z}^E_+$ (see [8], Theorem 4.1). We say that (P) is *totally dual integral* if for all $w \in \mathbb{Z}^E_+$, the dual linear program (D) has an integral optimal solution. It is also well-known that if (P) is totally dual integral, then $\mathcal{C}$ is an ideal clutter ([20, 13], see also [8], Theorem 4.26). The converse however does not hold, as we will explain shortly.

Define the *covering number* $\tau(\mathcal{C})$ as the minimum cardinality of a cover, and the *packing number* $\nu(\mathcal{C})$ as the maximum number of pairwise disjoint members. As every member of a packing picks a distinct element of a cover, it follows that $\tau(\mathcal{C}) \geq \nu(\mathcal{C})$. If equality holds here, then $\mathcal{C}$ *packs*, otherwise it is *non-packing*. Observe that $\tau(\mathcal{C})$ and $\nu(\mathcal{C})$ are the integral optimal values of (P) and (D), respectively, for $w = \mathbf{1}$. Thus, if (P) is totally dual integral, then $\mathcal{C}$ must pack.

Consider the clutter over ground set $\{1, \dots, 6\}$ whose members are

$$Q_6 := \big\{\{2,4,6\}, \{1,3,6\}, \{1,4,5\}, \{2,3,5\}\big\}.$$

Notice that Q_6 is isomorphic to the clutter of triangles (or claws) of the complete graph on four vertices. This clutter does not pack as $\tau(Q_6) = 2 > 1 = \nu(Q_6)$. This clutter was found by Lovász [26], but Seymour [36] was the person who established the significant role of Q_6 among non-packing clutters in his seminal paper on the matroids with the max-flow min-cut property. Even though Q_6 does not pack, it is an ideal clutter [36]. In fact, as we will see in §4,

Proposition 1.2. *Q_6 is the only ideal non-packing clutter over at most 6 elements, up to isomorphism.*

Given disjoint sets $I, J \subseteq E$, the *minor* of $\mathcal{C}$ obtained after *deleting* I and *contracting* J is the clutter

$$\mathcal{C} \setminus I/J := \text{ the minimal sets of } \{C - J : C \in \mathcal{C}, C \cap I = \emptyset\}.$$

We say that the minor is *proper* if $I \cup J \neq \emptyset$. In terms of the set covering polyhedron, contractions correspond to restricting the corresponding coordinates to 0, while deletions correspond to projecting away the corresponding coordinates; in terms of the set covering polytope, deletions can also be thought of as restricting the corresponding coordinates to 1, which is sometimes convenient. Due to these geometric interpretations, if a clutter is ideal then so is every minor of it [36]. A clutter is *minimally non-ideal* if it is not ideal but every proper minor is ideal. In the same vein, a clutter is *minimally non-packing* if it does not pack but every proper minor packs. A minimally non-packing clutter is either ideal or minimally non-ideal – this is a fascinating consequence of Lehman's seminal theorem on minimally non-ideal clutters [25] and was first noticed in [10].

Proposition 1.2 implies that Q_6 is in fact an ideal minimally non-packing clutter. Despite what Seymour [36] conjectured, Q_6 is not the only ideal minimally non-packing clutter. Schrijver [31] found an ideal minimally non-packing clutter over 9 elements, which was a minor of the clutter of dijoins of a directed graph, as a counterexample to a conjecture of Edmonds and Giles [13]. Two decades later, Cornuéjols, Guenin and Margot grew the known list to a dozen sporadic instances as well as an infinite class $\{Q_{r,t} : r \geq 1, t \geq 1\}$ of ideal minimally non-packing clutters [10]. All their examples of ideal minimally non-packing clutters, however, have covering number two, so they conjecture the following:

The $\tau = 2$ Conjecture ([10])**.** *Every ideal minimally non-packing clutter has covering number two.*

We will prove this conjecture for clutters over at most 8 elements (§4). For the most part, however, we take a different perspective towards the $\tau = 2$ Conjecture. Take an integer $n \geq 1$. We will be working over $\{0,1\}^n$, the vertices of the unit n-dimensional hypercube, represented for convenience as $0,1$ strings of length n. Take a set $S \subseteq \{0,1\}^n$. The *cuboid of* S, denoted $\mathrm{cuboid}(S)$, is the clutter over ground set $[2n]$ whose members have incidence vectors

$$(x_1, 1-x_1, \ldots, x_n, 1-x_n) \quad x \in S.^1$$

Observe that every member of $\mathrm{cuboid}(S)$ has cardinality n, and that for each $i \in [n]$, $\{2i-1, 2i\}$ is a cover. For example, the cuboid of $R_{1,1} := \{000, 110, 101, 011\} \subseteq \{0,1\}^3$ is $\big\{\{2,4,6\}, \{1,3,6\}, \{1,4,5\}, \{2,3,5\}\big\}$ which is Q_6. Thus, the smallest ideal minimally non-packing clutter is a cuboid. Abdi, Cornuéjols and Pashkovich showed that cuboids play a central role among all ideal minimally non-packing clutters [4]. They found two new ideal minimally non-packing cuboids, and observed that each clutter of $\{Q_{r,t}, r \geq 1, t \geq 1\}$ – the only known infinite class of ideal minimally non-packing clutters – is a cuboid. This was also observed by Flores, Gitler and Reyes, who referred to cuboids as 2-*partitionable* clutters [16]. However, to emphasize the fact that these clutters come from subsets of a hypercube, we refrain from this terminology. The following theorem further stresses the importance of cuboids among ideal minimally non-packing clutters:

Theorem 1.3. *Every minimally non-packing cuboid is ideal.*

This theorem is proved in §2. In this paper, we will see that the $\tau = 2$ Conjecture is equivalent to a conjecture on cuboids (§4), and furthermore, we will show how Seymour's classification of binary matroids with the sums of circuits property [33], his characterization of binary matroids with the max-flow min-cut property [36], as well as his f-Flowing Conjecture [36, 33] translate into the world of cuboids (§2 and §3). We will also reduce the Replication Conjecture of Conforti and Cornuéjols [7] to cuboids (§4). After reading this paper, we hope to have convinced the reader that cuboids are an important class of clutters.

1.1. **Cube-idealness.** Let $n \geq 1$ be an integer and $S \subseteq \{0,1\}^n$ an arbitrary set of vertices of the unit n-dimensional hypercube. Take a coordinate $i \in [n]$. To *twist coordinate* i is to replace S by

$$S \triangle e_i := \{x \triangle e_i : x \in S\};$$

this terminology is due to Bouchet [6]. (The symmetric difference operator $\triangle$ performs coordinatewise addition modulo 2. Novick and Sebő [29] refer to twisting as *switching*.) Observe that the cuboid of S encodes all of its twistings. If S' is obtained from S after twisting and relabeling some coordinates, then we say that S' is *isomorphic* to S and write it as $S' \cong S$. Notice that if S', S are isomorphic, then so are their cuboids.

The set obtained from $S \cap \{x : x_i = 0\}$ after dropping coordinate i is called the 0-*restriction of* S *over coordinate* i, and the set obtained from $S \cap \{x : x_i = 1\}$ after dropping coordinate i is called the 1-*restriction of* S *over coordinate* i. If S' is obtained from S after 0- and 1-restricting some coordinates, then we say that S' is a *restriction* of S. The set obtained from S after dropping coordinate i is called the *projection of* S *over coordinate* i. If S' is obtained from S after projecting away some coordinates, then we say that S' is a *projection* of S. If S' is obtained from S after a series of restrictions and projections, then we say that S' is a *minor* of S; we say that S' is a *proper* minor if at least one minor operation is applied. These minor operations can be defined directly on cuboids:

[1] For an integer $m \geq 1$, $[m] := \{1, \ldots, m\}$.

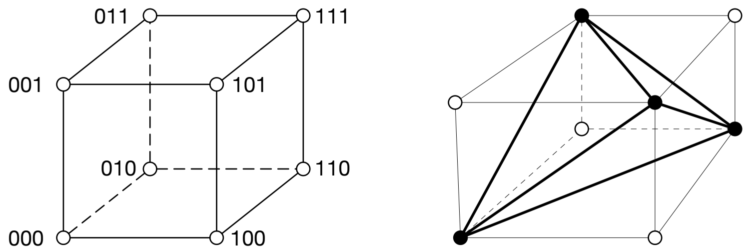


Figure 1. An illustration of the coordinate system, and the convex hull of $R_{1,1}$.

Remark 1.4 ([4]). *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then, for each $i \in [n]$, the following statements hold:*

- *If S' is the 0-restriction of S over i, then* $\mathrm{cuboid}(S') = \mathrm{cuboid}(S) \setminus (2i-1)/2i$.
- *If S' is the 1-restriction of S over i, then* $\mathrm{cuboid}(S') = \mathrm{cuboid}(S)/(2i-1) \setminus 2i$.
- *If S' is the projection of S over i, then* $\mathrm{cuboid}(S') = \mathrm{cuboid}(S)/\{2i-1, 2i\}$.

If S' is a minor of S, we will say that $\mathrm{cuboid}(S')$ is a *cuboid minor* of $\mathrm{cuboid}(S)$.

Inequalities of the form $1 \geq x_i \geq 0, i \in [n]$ are called *hypercube inequalities*, and the ones of the form

$$\sum_{i \in I} x_i + \sum_{j \in J}(1 - x_j) \geq 1 \qquad I, J \subseteq [n], I \cap J = \emptyset$$

are called *generalized set covering inequalities*. Observe that these two classes of inequalities are closed under twistings, i.e. the change of variables $x_i \mapsto 1 - x_i, i \in [n]$.

We say that S is *cube-ideal* if its convex hull $\mathrm{conv}(S)$ can be described using hypercube and generalized set covering inequalities. For instance, the set $R_{1,1} = \{000, 110, 101, 011\}$ is cube-ideal as its convex hull is

$$\mathrm{conv}(R_{1,1}) = \left\{ x \in [0,1]^3 : \begin{array}{ll} (1-x_1) + x_2 + x_3 & \geq 1 \\ x_1 + (1-x_2) + x_3 & \geq 1 \\ x_1 + x_2 + (1-x_3) & \geq 1 \\ (1-x_1) + (1-x_2) + (1-x_3) & \geq 1 \end{array} \right\},$$

as illustrated in Figure 1.

Remark 1.5. *Take an integer $n \geq 1$ and a cube-ideal set $S \subseteq \{0,1\}^n$. If S' is isomorphic to a minor of S, then S' is cube-ideal.*

Proof. Since the hypercube and generalized set covering inequalities are closed under relabelings and the transformation $x_i \mapsto 1 - x_i, i \in [n]$, being cube-ideal is closed under relabelings and twistings. It therefore suffices to prove the remark for the case when S' is obtained from S after a single minor operation. Suppose that $\mathrm{conv}(S) = \{x \in [0,1]^n : \sum_{i \in I} x_i + \sum_{j \in J}(1 - x_j) \geq 1, (I, J) \in \mathcal{V}\}$ for an appropriate $\mathcal{V}$. If S' is obtained from S after 0-restricting coordinate 1, then $\mathrm{conv}(S') = \{x \in [0,1]^{n-1} : \sum_{i \in I - \{1\}} x_i + \sum_{j \in J}(1 - x_j) \geq 1, (I, J) \in \mathcal{V}, 1 \notin J\}$. If S' is obtained from S after 1-restricting coordinate 1, then $\mathrm{conv}(S') = \{x \in [0,1]^{n-1} : \sum_{i \in I} x_i + \sum_{j \in J - \{1\}}(1 - x_j) \geq 1, (I, J) \in \mathcal{V}, 1 \notin I\}$. If S' is obtained from S after projecting away coordinate

1, then $\mathrm{conv}(S') = \{x \in [0,1]^{n-1} : \sum_{i\in I} x_i + \sum_{j\in J}(1-x_j) \geq 1, (I,J) \in \mathcal{V}, 1 \notin I \cup J\}$. In each case, we see that S' is still cube-ideal, thereby finishing the proof. $\square$

Cube-idealness of subsets of a hypercube can be defined solely in terms of cuboids:

Theorem 1.6. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then S is cube-ideal if, and only if,* $\mathrm{cuboid}(S)$ *is an ideal clutter.*

Using this theorem, which is proved in §2, we can use cube-idealness to link idealness to another deep property. We say that S is a *vector space over $GF(2)$*, or simply a *binary space*, if $a \triangle b \in S$ for all (possibly equal) points $a, b \in S$. A binary space is by definition the cycle space of a binary matroid (see [30]). For instance, $R_{1,1}$ is a binary space, and it corresponds to the cycle space of the graph on two vertices and three parallel edges. We will see in §2 that a binary space is cube-ideal if, and only if, the associated binary matroid has the *sums of circuits property*. Paul Seymour introduced this rich property in [35], and after developing his splitter theorems and decomposition of regular matroids [32], he classified the binary matroids with the sums of circuits property. (In that paper, he also posed the *cycle double cover conjecture* [33, 37].)

Theorem 1.6 reduces cube-idealness of subsets of a hypercube to clutter idealness; Theorem 4.3 gives a converse reduction (though with an exponential blow-up). As such, cube-idealness provides a framework to interpret clutter idealness geometrically, rather than combinatorially, as foreseen by Jon Lee [22]. To this end, take a point $x \in \{0,1\}^n$. The *induced clutter of S with respect to x*, denoted $\mathrm{ind}(S \triangle x)$, is the clutter over ground set $[n]$ whose members are

$$\mathrm{ind}(S \triangle x) = \text{ the minimal sets of } \big\{C \subseteq [n] : \chi_C \in S \triangle x\big\}.$$

In words, $\mathrm{ind}(S \triangle x)$ is the clutter corresponding to the points of $S \triangle x$ of minimal support. Notice that if $S = \emptyset$ then every induced clutter is $\{\}$, and in general, if $x \in S$ then $\mathrm{ind}(S \triangle x) = \{\emptyset\}$. Observe that

$$\mathrm{ind}(S \triangle x) = \mathrm{cuboid}(S)/\{2i : i \in [n], x_i = 0\}/\{2i-1 : i \in [n], x_i = 1\}.$$

Hence,

Remark 1.7. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then the 2^n induced clutters $\mathrm{ind}(S \triangle x), x \in \{0,1\}^n$ are in correspondence with the 2^n minors of $\mathrm{cuboid}(S)$ obtained after contracting, for each $i \in [n]$, exactly one of $2i-1, 2i$.*

It therefore follows from Theorem 1.6 that if S is cube-ideal, then all of its induced clutters are ideal. The converse of this statement, proved in §2, is also true:

Theorem 1.8. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then S is cube-ideal if, and only if, every induced clutter of S is ideal.*

Let $\mathcal{P}$ be a minor-closed property defined on clutters. Motivated by Theorem 1.8, we say that $\mathcal{P}$ is a *local property* if for all integers $n \geq 1$ and sets $S \subseteq \{0,1\}^n$, the following statements are equivalent:

- $\mathrm{cuboid}(S)$ has property $\mathcal{P}$,
- the induced clutters of S have property $\mathcal{P}$.

Otherwise, we say that $\mathcal{P}$ is a *non-local property*. Notice that Theorems 1.6 and 1.8 imply that idealness is a local property. Using the locality of idealness, we will be able to use the famous result of Edmonds and Johnson on T-join polytopes [14] to prove Seymour's result that graphic matroids have the sums of circuits property [35], as well as find a new link between the binary matroids with the sums of circuits property and the f-flowing binary matroids, and formulate the famous f-Flowing Conjecture in terms of cube-idealness of binary spaces (§2).

1.2. **Strict polarity.** We say that a clutter has the *packing property* if every minor, including the clutter itself, packs. Notice that a clutter has the packing property if, and only if, it has no minimally non-packing minor. Let us consider $R_{1,1}$ again. The induced clutters of this set are isomorphic to either $\{\emptyset\}$ or $\{\{1\},\{2\},\{3\}\}$, so they all have the packing property, whereas $\mathrm{cuboid}(R_{1,1}) = Q_6$ does not pack. Therefore, in contrast to idealness, the packing property is non-local. We will now see what causes the packing property to become local.

Let $n \geq 1$ be an integer. A pair of points $a, b \in \{0,1\}^n$ are *antipodal* if $a + b = \mathbf{1}$. Take a set $S \subseteq \{0,1\}^n$. We will refer to the points in S as *feasible* points and to the points in $\{0,1\}^n - S$ as *infeasible* points. We say that S is *polar* if either there are antipodal feasible points or all the feasible points agree on a coordinate:

$$\{x, \mathbf{1} - x\} \subseteq S \text{ for some } x \in \{0,1\}^n \qquad \text{or} \qquad S \subseteq \{x : x_i = a\} \text{ for some } i \in [n] \text{ and } a \in \{0,1\}.$$

For instance, the set $R_{1,1}$ is non-polar. Notice that if a set is polar, then so is every twisting of it. Moreover,

Remark 1.9. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then S is polar if, and only if,* $\mathrm{cuboid}(S)$ *packs.*

We say that S is *strictly polar* if every restriction, including S itself, is polar.

Remark 1.10. *Take an integer $n \geq 1$ and a strictly polar set $S \subseteq \{0,1\}^n$. If S' is isomorphic to a minor of S, then S' is polar.*

Proof. Being polar is closed under twistings and relabelings, so it suffices to prove that every minor of S is polar. To this end, let S' be a minor of S. Then there are disjoint sets $I, J, K \subseteq [n]$ such that S' is obtained after 0-restricting I, 1-restricting J and projecting away K; among all possible I, J, K we may assume that K is minimal, so that no single projection can be replaced by a single restriction. Let R be the restriction of S obtained after 0-restricting I and 1-restricting J; notice that S' is obtained from R after projecting away K. Since S is strictly polar, it follows from the definition that R is polar. If R contains antipodal points, then the same points give antipodal points in the projection S'. Otherwise, the points in R agree on a coordinate, so by the minimality of K, the points in the projection S' also agree on the same coordinate. In either cases, we see that S' is polar, as required. □

As a result, a set is strictly polar if, and only if, every cuboid minor of the corresponding cuboid packs. In particular, if $\mathrm{cuboid}(S)$ has the packing property, then S is strictly polar. We will see that once strict polarity is extracted, the non-local packing property becomes a local property:

Theorem 1.11. *Let S be a strictly polar set. Then* $\mathrm{cuboid}(S)$ *has the packing property if, and only if, all of the induced clutters of S have the packing property.*

This theorem is proved in §3. In that section, we will also see that strict polarity is a tractable property:

Theorem 1.12. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then the following statements are equivalent:*

(i) S is not strictly polar,

(ii) there are distinct points $a, b, c \in S$ such that the restriction of S containing them of smallest dimension is not polar.

As a result, in time $O(n|S|^4)$ one can certify whether or not S is strictly polar.

A set is *strictly non-polar* if it is not polar and every proper restriction is polar. Theorem 1.12 equivalently states that every strictly non-polar set has three distinct feasible points that do not all agree on a coordinate. A set is *minimally non-polar* if it is not polar and every proper minor is polar. A minimally non-polar set is strictly non-polar, and as we will see in §3, there are strictly non-polar sets that are not minimally non-polar. Observe that a set is strictly polar if, and only if, it has no strictly non-polar restriction if, and only if, it has no minimally non-polar minor.

1.3. **The Polarity Conjecture.** A fascinating consequence of Lehman's theorem on minimally non-ideal clutters [25] is the following:

Theorem 1.13 ([10])**.** *If a clutter has the packing property, then it is ideal.*

The converse however is not true, as there are ideal non-packing cuboids such as Q_6. And after all, we should not expect the two properties to be the same, because idealness is a local property but the packing property is not. However, as Theorem 1.11 shows, strict polarity makes the packing property local. We conjecture that strict polarity does far more than that:

The Polarity Conjecture. *Let S be a strictly polar set. Then* $\mathrm{cuboid}(S)$ *is ideal if, and only if,* $\mathrm{cuboid}(S)$ *has the packing property.*

Justified by Theorems 1.6 and 1.13, we may rephrase this conjecture as follows:

The Polarity Conjecture (rephrased)**.** *If a set is cube-ideal and strictly polar, then its cuboid has the packing property.*

As we will see in §4,

Theorem 1.14. *The Polarity Conjecture is equivalent to the $\tau = 2$ Conjecture.*

Take an integer $n \geq 3$ and a set $S \subseteq \{0,1\}^n$. We say that S is *critically non-polar* if it is strictly non-polar and, for each $i \in [n]$, both the 0- and 1-restrictions of S over coordinate i have antipodal points. We will see in §3 that critical non-polarity implies minimal non-polarity. In §4, we will see that if the Polarity Conjecture is true, then so is the following conjecture:

Conjecture 1.15. *If a set is cube-ideal and critically non-polar, then its cuboid is minimally non-packing.*

We will show in §4 that the Polarity Conjecture and Conjecture 1.15 are true for sets of *degree* at most 8 – this notion is defined later in the introduction.

In §5 we study three basic binary operations on pairs of sets. Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Define the *product*

$$S_1 \times S_2 := \big\{(x,y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} : x \in S_1 \text{ and } y \in S_2\big\}$$

and the *coproduct*

$$S_1 \oplus S_2 := \big\{(x,y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} : x \in S_1 \text{ or } y \in S_2\big\}.$$

Observe that $S_1 \oplus S_2 = \overline{\overline{S_1} \times \overline{S_2}}$, thereby justifying our terminology. We will observe that if the cuboids of two sets are ideal (resp. have the packing property), then so is (resp. does) the cuboid of their product. Moreover, by exploiting the locality of idealness, and the locality of the packing property once strict polarity is enforced, we show that if the cuboids of two sets are ideal (resp. have the packing property), then so is (resp. does) the cuboid of their coproduct. Define the *reflective product*

$$S_1 * S_2 := (S_1 \times S_2) \cup (\overline{S_1} \times \overline{S_2}).$$

In words, the reflective product $S_1 * S_2$ is obtained from S_1 after replacing each feasible point by a copy of S_2 and each infeasible point by a copy of $\overline{S_2}$. Observe that $S_1 * S_2 = \overline{S_1} * \overline{S_2}$ and $\overline{S_1 * S_2} = \overline{S_1} * S_2 = S_1 * \overline{S_2}$. We will see in §5 that,

Theorem 1.16. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. If $S_1, \overline{S_1}, S_2, \overline{S_2}$ are cube-ideal, then so are $S_1 * S_2, \overline{S_1 * S_2}$.*

That is, by Theorem 1.6, if $\text{cuboid}(S_1), \text{cuboid}(\overline{S_1}), \text{cuboid}(S_2), \text{cuboid}(\overline{S_2})$ are ideal, then so are $\text{cuboid}(S_1 * S_2), \text{cuboid}(\overline{S_1 * S_2})$. In contrast, the analogue of this for the packing property does not hold. For instance, let $S_1 := \{00, 11\}$ and $S_2 := \{0\}$. Then $\text{cuboid}(S_1), \text{cuboid}(\overline{S_1}), \text{cuboid}(S_2), \text{cuboid}(\overline{S_2})$ all have the packing property, while $\text{cuboid}(S_1 * S_2), \text{cuboid}(\overline{S_1 * S_2})$ are isomorphic to Q_6 and therefore do not pack. This phenomenon raises an intriguing question: can we build a counterexample to the Polarity Conjecture by taking the reflective product of two sets that are not counterexamples? As we will prove in §5, the answer is no:

Theorem 1.17. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$, where $\text{cuboid}(S_1)$, $\text{cuboid}(\overline{S_1}), \text{cuboid}(S_2), \text{cuboid}(\overline{S_2})$ have the packing property. If $S_1 * S_2$ is strictly polar, then $\text{cuboid}(S_1 * S_2)$ has the packing property.*

1.4. **Strictly non-polar sets.** A set is *antipodally symmetric* if a point is feasible if and only if its antipodal point is feasible. We will prove the following in §5:

Theorem 1.18. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$, where $S_1 * S_2$ is strictly non-polar. Then the following statements hold:*

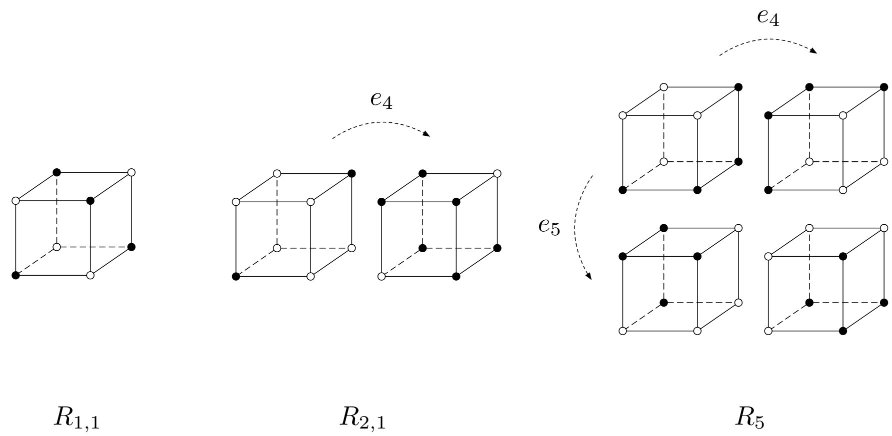


Figure 2. The strictly non-polar sets of degree at most 2. The filled-in points are feasible.

(1) $S_1, \overline{S_1}, S_2, \overline{S_2}$ *are nonempty.*

(2) Either $n_1 = 1$ *and* S_2 *is antipodally symmetric, or* $n_2 = 1$ *and* S_1 *is antipodally symmetric. In particular,* $S_1 * S_2 \cong \overline{S_1 * S_2}$.

(3) $S_1 * S_2$ *is critically non-polar.*

(4) If $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1}), \mathrm{cuboid}(S_2), \mathrm{cuboid}(\overline{S_2})$ *have the packing property, then* $\mathrm{cuboid}(S_1 * S_2)$ *is an ideal minimally non-packing clutter.*

For an integer $k \geq 1$, let

$$R_{k,1} := \{\mathbf{0}^{k+1}, \mathbf{1}^{k+1}\} * \{0\} \subseteq \{0,1\}^{k+2}.$$

(Hereinafter, $\mathbf{0}^m, \mathbf{1}^m$ are the m-dimensional vectors all of whose entries are $0, 1$, respectively.) See Figure 2 for an illustration of $R_{1,1}$ and $R_{2,1}$. The reader can readily check that $\{R_{k,1} : k \geq 1\}$ are strictly non-polar sets, and that their cuboids are isomorphic to the ideal minimally non-packing clutters $\{Q_{k,1} : k \geq 1\}$.

Take an integer $n \geq 1$. Denote by G_n the *skeleton graph of* $\{0,1\}^n$ whose vertices are the points in $\{0,1\}^n$ and two points u, v are adjacent if they differ in exactly one coordinate. A set $R \subseteq \{0,1\}^n$ is *connected* if $G_n[R]$ is connected. We say that $R \subseteq \{0,1\}^n$ is *strictly connected* if every restriction of R is connected.

The following result, proved in §5, is the second half of Theorem 1.18:

Theorem 1.19. *Take an integer* $n \geq 1$ *and an antipodally symmetric set* $S \subseteq \{0,1\}^n$ *such that* $S * \{0\}$ *is strictly non-polar. If* $S * \{0\}$ *is not isomorphic to any of* $\{R_{k,1} : k \geq 1\}$*, then both* S *and* $\overline{S}$ *are strictly connected.*

For an integer $k \geq 5$, let

$$C_{k-1} := \left\{ \sum_{i=1}^{d} e_i, \mathbf{1}^{k-1} - \sum_{i=1}^{d} e_i : d \in [k-1] \right\} \subseteq \{0,1\}^{k-1}$$
$$R_k := C_{k-1} * \{0\} \subseteq \{0,1\}^k.$$

See Figure 2 for an illustration of R_5. Notice that $G_{k-1}[C_{k-1}]$ is a cycle of length $2(k-1)$. The reader can readily check that $\{R_k : k \geq 5\}$ are strictly non-polar sets, and that $C_{k-1}, \overline{C_{k-1}}$ are strictly connected, verifying

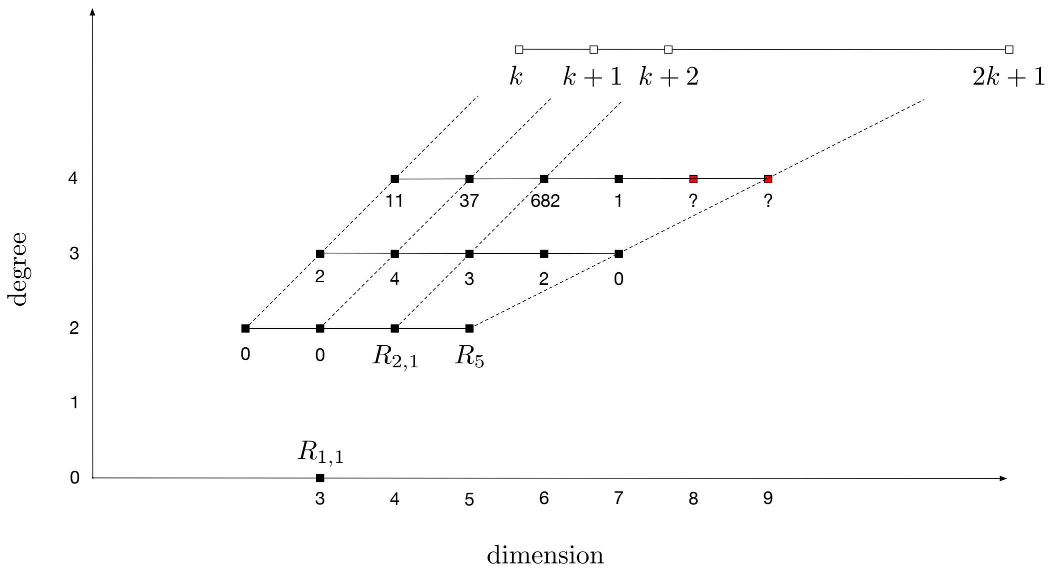


Figure 3. The spectrum of strictly non-polar sets. The number next to the filled-in square at coordinates (x, y) indicates the number of non-isomorphic strictly non-polar sets of dimension x and degree y.

Theorem 1.19. The cuboid of R_5 is the ideal minimally non-packing clutter Q_{10} found in [4], and as we will see in §3, the cuboids of $\{R_k : k \geq 6\}$ are not ideal and not minimally non-packing.

Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. For an integer $k \in \{0, 1, \ldots, n\}$, we say that S has *degree at most* k if every infeasible point has at most k infeasible neighbors in G_n. We say that S has *degree* k if it has degree at most k and not $k - 1$. As a result, given a set of degree k, every infeasible point has at most k infeasible neighbors, and there is an infeasible point achieving this bound. For each $k \geq 1$, it is known that every strictly non-polar set of degree at most k must have dimension at most $4k+1$ ([4], Theorem 1.10 (i)). It was also shown that, up to isomorphism, the strictly non-polar sets of degree at most 2 are $R_{1,1}, R_{2,1}, R_5$, as displayed in Figure 2 ([4], Theorem 1.9). We will improve in §6 the upper bound of $4k + 1$ as follows:

Theorem 1.20. *Take an integer $k \geq 2$ and a strictly non-polar set S of degree k, whose dimension is n. Then the following statements hold:*

(1) $n \in \{k, k + 1, \ldots, 2k + 1\}$.

(2) If $n = k + 1$, then either S is minimally non-polar, or after a possible relabeling,

$$S \subseteq \left\{x \in \{0,1\}^{k+1} : x_k = x_{k+1}\right\}$$

and the projection of S over coordinate $k+1$ is a critically non-polar set that is the reflective product of two other sets.

(3) If $n \geq k + 2$, then S is critically non-polar.

(4) If $n = 2k + 1$, then $|S| = 2^{n-1}$, every infeasible point has exactly k infeasible neighbors, and $\operatorname{cuboid}(S)$ *is an ideal minimally non-packing clutter.*

Part (4) is done by using Mantel's Theorem [27] as well as the local structure of delta free clutters. Notice that R_5, which is of degree 2 and dimension 5, has 16 points, every infeasible point has exactly 2 infeasible neighbors, and $\mathrm{cuboid}(R_5) = Q_{10}$ is an ideal minimally non-packing clutter. In §6 we will describe a computer code, whose correctness relies on Theorem 1.20 (1), that generates all the non-isomorphic strictly non-polar sets of degree at most 3, as well as all the non-isomorphic strictly non-polar sets of degree 4 and dimension at most 7. As we will see, there are exactly 745 non-isomorphic strictly non-polar sets of degree at most 4 and dimension at most 7, explicitly described in the appendix and summarized in Figure 3, 716 sets of which have ideal minimally non-packing cuboids.

2. Cube-ideal sets

In this section we prove Theorems 1.3, 1.6, and 1.8. We will also characterize the cube-ideal binary spaces, discuss the sums of circuits property, the theorem of Edmonds and Johnson on T-join polytopes [14] and the f-Flowing Conjecture.

2.1. **Idealness is a local property.** Given a clutter $\mathcal{C}$, the *blocker* $b(\mathcal{C})$ is another clutter over the same ground set whose members are the minimal covers of $\mathcal{C}$. It is well-known that $b(b(\mathcal{C})) = \mathcal{C}$ [21, 12]. We will need the following lemma; recall that $Q(\mathcal{C})$ denotes the set covering polyhedron of $\mathcal{C}$.

Lemma 2.1 ([4], Lemma 3.1, also see [19, 28])**.** *Take a clutter $\mathcal{C}$ over ground set $E = \{e_1, f_1, \ldots, e_n, f_n\}$, where for each $i \in [n]$, $\{e_i, f_i\}$ intersects every member exactly once. Then the following statements are equivalent:*

(i) $b(\mathcal{C})$ *is ideal,*
(ii) $\mathrm{conv}\{\chi_C : C \in \mathcal{C}\} = Q\big(b(\mathcal{C})\big) \cap \{x : x_{e_i} + x_{f_i} = 1 \;\forall i \in [n]\}$.

Lehman's Width-Length Inequality implies that a clutter is ideal if, and only if, its blocker is ideal ([24, 17], also see [9], Theorem 1.21). Using this fact and the preceding lemma, we are ready to prove Theorem 1.6, stating that a subset of a hypercube is cube-ideal if and only if the corresponding cuboid is ideal.

Proof of Theorem 1.6. Let $\mathcal{C} := \mathrm{cuboid}(S)$. Recall that $\mathcal{C}$ is over ground set $E := \{1, 2, \ldots, 2n-1, 2n\}$, where for each $i \in [n]$, $\{2i-1, 2i\}$ intersects every member exactly once. We may therefore apply Lemma 2.1. ($\Leftarrow$) Assume that $\mathcal{C}$ is ideal. Then $b(\mathcal{C})$ is ideal also. Thus by Lemma 2.1, we have that

$$\mathrm{conv}\{\chi_C : C \in \mathcal{C}\} = Q\big(b(\mathcal{C})\big) \cap \{x : x_{2i-1} + x_{2i} = 1 \;\forall i \in [n]\}.$$

By projecting away the even coordinates, we get that

$$\mathrm{conv}(S) = \left\{ y \in [0,1]^n : \sum \big(y_i : 2i-1 \in B\big) + \sum \big(1 - y_j : 2j \in B\big) \geq 1 \quad \forall B \in b(\mathcal{C}) \right\}.$$

As a result, S is cube-ideal. ($\Rightarrow$) Assume conversely that S is cube-ideal. Then

$$\mathrm{conv}(S) = \left\{ y \in [0,1]^n : \sum \big(y_i : i \in I\big) + \sum \big(1 - y_j : j \in J\big) \geq 1 \quad \forall\, (I, J) \in \mathcal{V} \right\},$$

for some appropriate set $\mathcal{V}$. We may assume that for each $(I,J)\in\mathcal{V}$, $I\cap J=\emptyset$. After the change of variables $y_i\mapsto x_{2i-1}$ and $1-y_i\mapsto x_{2i}$ to the equation above, we get that

$$\operatorname{conv}\{\chi_C : C\in\mathcal{C}\}=\left\{x\in\mathbb{R}^{2n}_+ : \begin{array}{l}\sum\left(x_{2i-1}: i\in I\right)+\sum\left(x_{2j}: j\in J\right)\geq 1 \quad \forall\,(I,J)\in\mathcal{V}\\ x_{2i-1}+x_{2i}=1 \quad \forall\, i\in[n]\end{array}\right\}.$$

Together with Lemma 2.1, this equation implies that $b(\mathcal{C})$ is an ideal clutter, so $\mathcal{C}$ is ideal, as required. □

A pair of columns of a $0-1$ matrix are *complementary* if they add up to the all ones vector. Moving forward, we need the following consequence of Lehman's theorem:

Lemma 2.2 ([25, 34], also see [4], Theorem 4.6)**.** *Let $\mathcal{C}$ be a minimally non-ideal clutter. Then $\tau(\mathcal{C})\geq 2$ and $M(\mathcal{C})$ does not have complementary columns.*

An immediate consequence of Lemma 2.2 is Theorem 1.3, stating that every minimally non-packing cuboid is ideal.

Proof of Theorem 1.3. Let $\mathcal{C}$ be a cuboid that is minimally non-packing. Then $\mathcal{C}$ is either minimally non-ideal or ideal. From the definition of cuboids, $M(\mathcal{C})$ has complementary columns, so Lemma 2.2 implies that $\mathcal{C}$ is not minimally non-ideal. Thus, $\mathcal{C}$ is an ideal clutter. □

Another consequence of Lemma 2.2 is Theorem 1.8, stating that a subset of a hypercube is cube-ideal if and only if all of its induced clutters are ideal.

Proof of Theorem 1.8. Let $\mathcal{C}:=\mathrm{cuboid}(S)$. By Theorem 1.6, it suffices to show that $\mathcal{C}$ is ideal if, and only if, all of the induced clutters of S are ideal. ($\Rightarrow$) Assume that $\mathcal{C}$ is ideal. Then all of its minors are ideal, so by Remark 1.7, all of the induced clutters of S are ideal. ($\Leftarrow$) Assume that $\mathcal{C}$ is non-ideal. Pick disjoint $I,J\subseteq[2n]$ such that the minor $\mathcal{C}':=\mathcal{C}\setminus I/J$ is minimally non-ideal. By Lemma 2.2, $\tau(\mathcal{C}')\geq 2$ and $M(\mathcal{C}')$ does not have complementary columns; these facts imply that for each $i\in[n]$,

- if $I\cap\{2i-1,2i\}\neq\emptyset$ then $J\cap\{2i-1,2i\}\neq\emptyset$, and so
- $J\cap\{2i-1,2i\}\neq\emptyset$.

The latter implies by Remark 1.7 that $\mathcal{C}'$ is a minor of an induced clutter of S, implying in turn that an induced clutter of S is non-ideal, as required. □

2.2. **The sums of circuits property.** Take an integer $n\geq 1$ and a set $S\subseteq\{0,1\}^n$. We say that S is an *affine vector space over* $GF(2)$, or simply an *affine binary space*, if the symmetric difference of any odd number of feasible points is also feasible. Notice that affine binary spaces are nothing but twists of binary spaces. Basic Linear Algebra implies that S is an affine binary space if and only if

$$S=\left\{x\in\{0,1\}^n : Ax\equiv b \pmod 2\right\}$$

for a $0-1$ matrix A and a $0-1$ vector b of appropriate dimensions, and that S is a binary space if and only if $b=\mathbf{0}$. We will need the following routine application of the Gaussian elimination method:

Lemma 2.3 (folklore, see [23], (44))**.** *Take integers $m, n \geq 1$, an $m \times n$ matrix A and an m-dimensional vector b with $0-1$ entries. If the system $Ax \equiv b \pmod 2$ does not have a solution in $\{0,1\}^n$, then there exists a vector $c \in \{0,1\}^m$ such that $c^\top A \equiv \mathbf{0}$ and $c^\top b \equiv 1 \pmod 2$.*

This result can be viewed as the Farkas lemma for binary spaces. Moving forward, take a binary space S represented as

$$S = \big\{x \in \{0,1\}^n : Ax \equiv \mathbf{0} \pmod 2\big\}.$$

By definition, S is the cycle space of a binary matroid M (see [30]). We refer to M as the *associated* binary matroid. The cocycle space of M is precisely the binary space generated by the rows of A ([30], Proposition 9.2.2).

Theorem 2.4. *Take an integer $n \geq 1$ and a binary space $S \subseteq \{0,1\}^n$, and let M be the associated binary matroid. Then S is cube-ideal if, and only if,*

$$\operatorname{conv}(S) = \big\{x \in [0,1]^n : x(F) - x(D-F) \leq |F| - 1 \quad \forall \textit{ cocycles } D \textit{ and odd subsets } F \subseteq D\big\}.$$

Proof. ($\Leftarrow$) Notice that each inequality $x(F) - x(D-F) \leq |F| - 1$ can be rewritten as

$$\sum_{i \in D-F} x_i + \sum_{j \in F} (1 - x_j) \geq 1,$$

which is a generalized set covering inequality. Thus, S is cube-ideal. ($\Rightarrow$) Suppose conversely that S is cube-ideal. We first prove that

$$\operatorname{conv}(S) \subseteq \big\{x \in [0,1]^n : x(F) - x(D-F) \leq |F| - 1 \quad \forall \text{ cocycles } D \text{ and odd subsets } F \subseteq D\big\}.$$

Denote by P the polytope on the right-hand side. To prove this inclusion, it suffices to show that for every cycle C, χ_C belongs to P. Well, for every cocycle D and odd subset $F \subseteq D$, we have $C \cap D \neq F$ because $|C \cap D|$ is even, so if $F \subseteq C$ then $C \cap (D - F) \neq \emptyset$, implying in turn that

$$\chi_C(F) - \chi_C(D-F) \leq |F| - 1.$$

Thus, $\chi_C \in P$. To prove the reverse inclusion, it suffices to prove that every inequality defining $\operatorname{conv}(S)$ is valid for P. Since S is cube-ideal, $\operatorname{conv}(S)$ is described by hypercube inequalities – which are valid for P – and by generalized set covering inequalities. Take disjoint subsets $I, J \subseteq [n]$ such that $\sum_{i \in I} x_i + \sum_{j \in J}(1 - x_j) \geq 1$ is a defining inequality of $\operatorname{conv}(S)$.

Claim. *There is a cocycle D such that $D \subseteq I \cup J$ and $|D \cap J|$ is odd.*

Proof of Claim. To see this, write

$$S = \big\{x \in \{0,1\}^n : Ax \equiv \mathbf{0} \pmod 2\big\}$$

for some $0-1$ matrix A. Let d be the sum of the columns in J of A, and let B be the submatrix of A obtained after dropping columns $I \cup J$. Since $\sum_{i \in I} x_i + \sum_{j \in J}(1 - x_j) \geq 1$ is valid for every point of S, the system

$$By \equiv d \pmod 2$$

has no $0-1$ solution. (For if y is a solution, then by setting the coordinates in I to 0 and the coordinates in J to 1, we can extend y to a feasible point x of S for which $\sum_{i\in I} x_i + \sum_{j\in J}(1-x_j) = 0$, which is not the case.) By Lemma 2.3, there is a $0-1$ vector c such that

$$c^\top B \equiv \mathbf{0} \quad \text{and} \quad c^\top d \equiv 1 \pmod 2.$$

Consider the cocycle $D \subseteq [n]$ for which $\chi_D = c^\top A$. Then the first equation implies that $D \subseteq I \cup J$, while the second equation implies that $|D \cap J|$ is odd, as required. ◊

Let $F := D \cap J$. Then F is an odd subset of the cocycle D. Observe that the inequality

$$\sum_{i\in I} x_i + \sum_{j\in J}(1-x_j) \geq 1$$

is dominated by the inequality

$$\sum_{i\in D-F} x_i + \sum_{j\in F}(1-x_j) \geq 1 \qquad \text{which is equivalent to} \qquad x(F) - x(D-F) \leq |F|-1,$$

because $D - F \subseteq I$ and $F \subseteq J$. As a result, every inequality defining $\mathrm{conv}(S)$ is valid for P, so $\mathrm{conv}(S) \supseteq P$. Hence, $\mathrm{conv}(S) = P$, thereby finishing the proof. □

If S is a binary space, then $S \triangle x = S$ for every feasible point x. Taking advantage of this transitive property of binary spaces, Barahona and Grötschel proved the following striking result:

Theorem 2.5 ([5], (3.2))**.** *Take an integer $n \geq 1$ and a binary space $S \subseteq \{0,1\}^n$, and let M be the associated binary matroid. Then*

$$\mathrm{conv}(S) = \left\{x \in [0,1]^n : x(F) - x(D-F) \leq |F|-1 \quad \forall \textit{ cocycles } D \textit{ and odd subsets } F \subseteq D\right\}$$

if, and only if, M has the sums of circuits property.

A binary matroid M over ground set $[n]$ has the *sums of circuits property* if for all $w \in \mathbb{R}^n_+$ such that

$$w(D - \{f\}) \geq w_f \quad \forall \text{ cocycles } D \text{ and } f \in D,$$

there exists an assignment $y_C \in \mathbb{R}_+$ to every circuit C such that

$$w = \sum \left(y_C \cdot \chi_C : C \text{ is a circuit}\right).$$

As an immediate consequence of Theorems 2.4 and 2.5, we get that

Corollary 2.6. *A binary space is cube-ideal if, and only if, the associated binary matroid has the sums of circuits property.*

Paul Seymour introduced the sums of circuits property in 1979 and then he proved that graphic matroids have this property. Let us prove this result by using the locality of idealness as well as the seminal result of Edmonds and Johnson. Let $G = (V, E)$ be a graph and take a subset $T \subseteq V$ of even cardinality. A *T-join* is an edge subset whose odd-degree vertices in G are precisely T.

Theorem 2.7 ([14], see [9], Theorems 1.21 and 2.1)**.** *Let $G=(V,E)$ be a graph and take a subset $T\subseteq V$ of even cardinality. Then the clutter of minimal T-joins is ideal.*

Using this theorem we can now prove the following:[2]

Theorem 2.8 ([35], (1.4))**.** *Graphic matroids have the sums of circuits property.*

Proof. Let $G=(V,E)$ be a graph. Consider the binary space associated with G:

$$S:=\{\chi_C : C\subseteq E \text{ is a cycle of } G\}\subseteq\{0,1\}^E.$$

By Corollary 2.6, it suffices to show that S is cube-ideal, and to do this, it suffices by Theorem 1.8 to show that the induced clutters of S are ideal. Take a point $\chi_A\in\{0,1\}^E$. Then

$$S\triangle\chi_A=\{\chi_C\triangle\chi_A : C \text{ is a cycle}\}=\{\chi_{C\triangle A} : C \text{ is a cycle}\}=\{\chi_J : J\triangle A \text{ is a cycle}\}=\{\chi_J : J \text{ is a } T\text{-join}\}$$

where T is the set of odd-degree vertices of $A\subseteq E$. As a result, $\mathrm{ind}(S\triangle\chi_A)$ is the clutter of minimal T-joins of G, which by Theorem 2.7 is an ideal clutter, as required. □

Let $G=(V,E)$ be a bridgeless graph. We just proved that there is an assignment $y_C\in\mathbb{R}_+$ to every circuit C such that

$$\mathbf{1}=\sum\left(y_C\cdot\chi_C : C \text{ is a circuit}\right).$$

The famous *cycle double cover conjecture* predicts that y may be chosen to be half-integral [37, 35, 33].

2.3. **The f-Flowing Conjecture.** In addition to graphic matroids, the Fano matroid and the cut matroid of the Wagner graph happen to have the sums of circuits property as well. After developing his so-called splitter theorems and a decomposition theorem for regular matroids, Seymour proved that these are *the* building blocks of binary matroids with the sums of circuits property, and obtained the following as a consequence:[3]

Theorem 2.9 ([33], (16.4))**.** *A binary matroid has the sums of circuits property if, and only if, it has none of $F_7^\star, R_{10}, M(K_5)^\star$ as an isomorphic minor.*[4]

$F_7^\star$ is the dual of the Fano matroid, R_{10} is the binary matroid whose graft representation is displayed in Figure 4, and $M(K_5)^\star$ is the cut matroid of K_5. What does this result say in terms of cube-ideal affine binary spaces? We will need the following remark:

Remark 2.10. *Let S be a binary space, and let M be the associated binary matroid. Then for an element e,*

- *the 0-restriction of S over e is a binary space whose associated binary matroid is $M\setminus e$,*
- *the 1-restriction of S over e is either empty or isomorphic to the 0-restriction of S over e,*

[2]In the same vein, using his strengthening of Theorem 2.7 that the clutter of T-cuts of a bipartite graph packs, Seymour proved that cographic matroids are 2-cycling [33].

[3]To be accurate, he proved all of these results on the dual matroid, where the sums of circuits property corresponds to the ∞-*flowing property*.

[4]The prefix "isomorphic" from "isomorphic minor" will be omitted hereinafter.

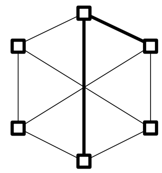


Figure 4. A graft representation of R_{10}.

- *the projection of S over e is a binary space whose associated binary matroid is M/e.*

For a binary matroid M over ground set E, denote by $\mathcal{C}(M)$ the binary space $\{\chi_C : C \text{ is a cycle}\} \subseteq \{0,1\}^E$. Then by the preceding remark, Theorem 2.9 equivalently states that an affine binary space is cube-ideal if, and only if, it has none of $\mathcal{C}(F_7^\star), \mathcal{C}(R_{10}), \mathcal{C}(M(K_5)^\star)$ as a minor. We know by Theorem 1.8 that a set is cube-ideal if and only if all of its induced clutters are ideal. So what does Theorem 2.9 say in terms of the induced clutters of a cube-ideal affine binary space? We will need the following remark:

Remark 2.11. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then the following statements hold: (i) an induced clutter of a minor of S is a minor of an induced clutter of S, and (ii) a minor of an induced clutter of S is an induced clutter of a minor of S.*

Since $\mathcal{C}(F_7^\star), \mathcal{C}(R_{10}), \mathcal{C}(M(K_5)^\star)$ are not cube-ideal, it follows from Theorem 1.8 that each one of these has a non-ideal induced clutter. Consider the induced clutters

$$\mathbb{L}_7 := \mathrm{ind}\big(\mathcal{C}(F_7^\star)\triangle \mathbf{1}\big) \quad \text{and} \quad \mathbb{O}_5 := \mathrm{ind}\big(\mathcal{C}(R_{10})\triangle\{a,b,c\}\big) \quad \text{and} \quad b(\mathbb{O}_5) := \mathrm{ind}\big(\mathcal{C}(M(K_5)^\star)\triangle \mathbf{1}\big),$$

where, in the graft representation of R_{10} in Figure 4, a and b are the elements corresponding to the two bold edges, and c corresponds to the T set. In words, $\mathbb{L}_7$ is the clutter of lines of the Fano plane, $\mathbb{O}_5$ is the clutter of odd circuits of K_5, while $b(\mathbb{O}_5)$ is the clutter of cut complements of K_5 and is the blocker of $\mathbb{O}_5$. It can be readily checked that $\mathbb{L}_7$ (resp. $\mathbb{O}_5, b(\mathbb{O}_5)$) is non-ideal, and up to isomorphism, it is the unique non-ideal induced clutter of $\mathcal{C}(F_7^\star)$ (resp. $\mathcal{C}(R_{10}), \mathcal{C}(M(K_5)^\star)$). Thus, together with Theorem 2.9 and Remark 2.11, we get the following:

Corollary 2.12. *An affine binary space is cube-ideal if, and only if, its induced clutters have none of $\mathbb{L}_7, \mathbb{O}_5, b(\mathbb{O}_5)$ as a minor.*

This corollary turns out to be a weakening of a well-known conjecture in the field – let us elaborate. We say that a clutter is *binary* if the symmetric difference of any odd number of (not necessarily distinct) members contains a member [23]. For instance, the clutters $\mathbb{L}_7, \mathbb{O}_5, b(\mathbb{O}_5)$ are binary.

The f-Flowing Conjecture ([36, 33])**.** *A binary clutter is ideal if, and only if, it has none of $\mathbb{L}_7, \mathbb{O}_5, b(\mathbb{O}_5)$ as a minor.*

We can rephrase this conjecture in terms of induced clutters of affine binary spaces. We will need the following:

Remark 2.13. *A clutter is binary if, and only if, it is an induced clutter of an affine binary space.*

Proof. ($\Rightarrow$) Let $\mathcal{C}$ be a binary clutter over ground set E, and let

$$S := \big\{\chi_C : C \subseteq E \text{ is the symmetric difference of an odd number of members}\big\}.$$

By definition, S is an affine binary space. As $\mathcal{C}$ is a binary clutter, it follows that $\mathrm{ind}(S) = \mathcal{C}$, as required. ($\Leftarrow$) Let $\mathcal{C}$ be an induced clutter of an affine binary space S. After a possible twisting, we may assume that $\mathcal{C} = \mathrm{ind}(S)$. In S, the sum of any odd number of feasible points modulo 2 is another feasible point, implying in turn that the symmetric difference of any odd number of members of $\mathcal{C}$ contains a member, so $\mathcal{C}$ is a binary clutter, as required. $\square$

As a result, we have the following rephrasing:

The f-Flowing Conjecture (rephrased)**.** *Let $\mathcal{C}$ be an induced clutter of an affine binary space. Then $\mathcal{C}$ is ideal if, and only if, it has none of $\mathbb{L}_7, \mathbb{O}_5, b(\mathbb{O}_5)$ as a minor.*

The crux of this conjecture is the ($\Leftarrow$) direction. To this end, assume that $\mathcal{C}$ is a non-ideal induced clutter of an affine binary space S. Then by Theorem 1.8, S is not cube-ideal, so by Corollary 2.12, we get that an induced clutter of S has one of $\mathbb{L}_7, \mathbb{O}_5, b(\mathbb{O}_5)$ as minor. Now, if this induced clutter happens to be $\mathcal{C}$, then we are done. In this sense, Corollary 2.12 is a weakening of the f-Flowing Conjecture.

3. Strictly polar sets

In this section we prove Theorems 1.11 and 1.12, discuss strict, minimal and critical non-polarity, characterize when the cuboid of a strictly non-polar set is minimally non-packing, characterize the strictly polar binary spaces, and discuss Seymour's characterization of the binary matroids with the max-flow min-cut property.

3.1. **Strict polarity makes the packing property local.** We will need the following characterization of strict polarity in terms of cuboids:

Proposition 3.1. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. The following statements are equivalent: (i) S is strictly polar, (ii) every cuboid minor of $\mathrm{cuboid}(S)$ packs, (iii) every minor of $\mathrm{cuboid}(S)$ has a cover of cardinality one, or two disjoint members.*

Proof. **(i)** $\Rightarrow$ **(ii)**: Since S is strictly polar, Remark 1.10 implies that every minor of S is polar, so every cuboid minor of $\mathrm{cuboid}(S)$ packs. **(ii)** $\Rightarrow$ **(iii)**: Let $\mathcal{C}$ be a minor of $\mathrm{cuboid}(S)$ such that $\tau(\mathcal{C}) \geq 2$ and every element of $\mathcal{C}$ is contained in a member. It suffices to show that $\mathcal{C}$ has two disjoint members. To this end, pick disjoint $I, J \subseteq [2n]$ such that $\mathrm{cuboid}(S) \setminus I/J = \mathcal{C}$. As $\tau(\mathcal{C}) \geq 2$, for each $i \in [n]$ such that $I \cap \{2i-1, 2i\} \neq \emptyset$, we must have that $J \cap \{2i-1, 2i\} \neq \emptyset$. Let $\mathcal{C}'$ be the cuboid minor of $\mathrm{cuboid}(S)$ obtained after deleting I and contracting $\{2j-1 : j \in [n], 2j \in I\} \cup \{2j : j \in [n], 2j-1 \in I\} \subseteq J$. By (ii), the cuboid $\mathcal{C}'$ packs. Since $\tau(\mathcal{C}) \geq 2$ and every element of $\mathcal{C}$ is contained in a member, we see that $\tau(\mathcal{C}') = 2$, implying in turn that $\mathcal{C}'$ has two disjoint members. Since $\mathcal{C}$ is a contraction minor of $\mathcal{C}'$, we get that $\mathcal{C}$ too has two disjoint members. **(iii)** $\Rightarrow$ **(i)**: In particular, every cuboid minor of $\mathrm{cuboid}(S)$ packs, so every minor of S is polar, implying in turn that S is strictly polar. $\square$

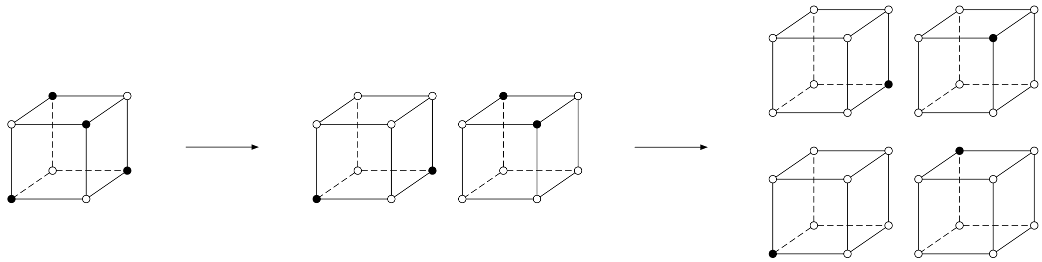


Figure 5. An illustration of Remark 3.2: constructing new strictly non-polar sets starting from $R_{1,1}$.

We are now ready to prove Theorem 1.11, stating that for a strictly polar set, the cuboid has the packing property if and only if all of the induced clutters have the packing property:

Proof of Theorem 1.11. Let S be a strictly polar set. ($\Rightarrow$) Suppose that $\mathrm{cuboid}(S)$ has the packing property. Then all of the minors of $\mathrm{cuboid}(S)$, including all of the induced clutters of S by Remark 1.7, have the packing property. ($\Leftarrow$) Suppose that $\mathrm{cuboid}(S)$ does not have the packing property. Let $\mathcal{C}$ be a non-packing minor of $\mathrm{cuboid}(S)$. As S is strictly polar, it follows from Proposition 3.1 (iii) that either $\tau(\mathcal{C}) = 1$ or $\mathcal{C}$ has two disjoint members. However, $\mathcal{C}$ does not pack, so $\tau(\mathcal{C}) \neq 1$, implying in turn that $\mathcal{C}$ has two disjoint members, so $\tau(\mathcal{C}) \geq 3$. Pick disjoint subsets $I, J \subseteq [2n]$ such that $\mathcal{C} = \mathrm{cuboid}(S) \setminus I/J$. Since $\tau(\mathcal{C}) \geq 3$, it follows that for each $i \in [n]$, $J \cap \{2i-1, 2i\} \neq \emptyset$. Hence, $\mathcal{C}$ is a minor of an induced clutter of S by Remark 1.7, implying in turn that an induced clutter of S does not have the packing property, as required. $\square$

3.2. **Strict, minimal and critical non-polarity.** As we noted, $\{R_{k,1} : k \geq 1\} \cup \{R_k : k \geq 5\}$ are strictly non-polar sets. We may in fact construct infinitely many strictly non-polar sets by starting from an existing one:

Remark 3.2. *Take an integer $n \geq 3$, a set $S \subseteq \{0,1\}^n$, a coordinate $i \in [n]$ and $a \in \{0,1\}$. Then S is strictly non-polar if, and only if, the set*

$$\big\{(x,a) : x \in S, x_i = 0\big\} \cup \big\{(x,1-a) : x \in S, x_i = 1\big\} \subseteq \{0,1\}^{n+1}$$

is strictly non-polar.

We leave the proof of this remark as an exercise for the reader, but we will demonstrate it with the example illustrated in Figure 5. Starting with the strictly non-polar set $R_{1,1}$, and setting $i := 3, a := 0$, we obtain the strictly non-polar set $\{0000, 1100, 0111, 1011\}$; starting with this new strictly non-polar set, and setting $i = 1, a = 1$, we obtain the strictly non-polar set $\{00001, 11000, 01111, 10110\}$. These new strictly non-polar sets, however, are not very interesting as they have $R_{1,1}$ as a projection.

Recall that a set is minimally non-polar if it is not polar and every proper minor is polar. Notice that every minimally non-polar set is strictly non-polar, but the converse is obviously not true given Remark 3.2. The following proposition tells us when a strictly non-polar set is minimally non-polar:

Proposition 3.3. *Take an integer $n \geq 3$ and a strictly non-polar set $S \subseteq \{0,1\}^n$. Then the following statements are equivalent:*

(i) S is minimally non-polar,

(ii) for each $i \in [n]$, either the 0-restriction or the 1-restriction of S over coordinate i has antipodal points.

Proof. **(i)** $\Rightarrow$ **(ii)**: Take a coordinate $i \in [n]$, and let $S' \subseteq \{0,1\}^{[n]-\{i\}}$ be the projection of S over coordinate i. As S is minimally non-polar, it follows that S' is polar. Since the points in S do not all agree on a coordinate, neither do the points in S', so S' has antipodal points; and as S does not contain antipodal points, these points yield antipodal points in either the 0- or 1-restriction of S over coordinate i. **(ii)** $\Rightarrow$ **(i)**: Take a nonempty and proper subset I of $[n]$. It suffices to show that the projection of S over the coordinates I has antipodal points. Pick a coordinate $i \in I$. Then either the 0- or 1-restriction of S over i has antipodal points, so the projection of S over i has antipodal points, implying in particular that the projection of S over I has antipodal points, as required. $\square$

It may now be checked that $\{R_{k,1} : k \geq 1\} \cup \{R_k : k \geq 5\}$ are minimally non-polar sets. Notice that S is minimally non-polar if, and only if, $\mathrm{cuboid}(S)$ does not pack and every proper cuboid minor packs. In particular, if $\mathrm{cuboid}(S)$ is minimally non-packing, then S is minimally non-polar. Even though it is the case for $\{R_{k,1} : k \geq 1\} \cup \{R_5\}$, the converse does not hold in general.

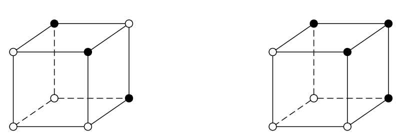
Remark 3.4. *Take an integer $k \geq 6$. Then $\mathrm{cuboid}(R_k)$ has $\big\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,1\}\big\}$ as a minor. In particular, $\mathrm{cuboid}(R_k)$ is not ideal and not minimally non-packing.*

Proof. Recall that $R_k = C_{k-1} * \{0\}$ where

$$C_{k-1} = \left\{\sum_{i=1}^{d} e_i, \mathbf{1}^{k-1} - \sum_{i=1}^{d} e_i : d \in [k-1]\right\} \subseteq \{0,1\}^{k-1}.$$

Notice that the projection of C_k over the last coordinate is C_{k-1}. As a result, $R_k = C_{k-1} * \{0\}$ has a C_5 minor. Since $\mathrm{ind}(C_5 \triangle 01010)$ is isomorphic to $\mathcal{C}_5^2 := \big\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,1\}\big\}$, it follows from Remark 1.7 that $\mathrm{cuboid}(R_k)$ has a $\mathcal{C}_5^2$ minor, thereby proving the first part of the remark. For the second part, notice that $\mathcal{C}_5^2$ is non-ideal as $\left(\frac{1}{2}, \ldots, \frac{1}{2}\right) \in \mathbb{R}_+^5$ is an extreme point of the corresponding set covering polyhedron, and that $\mathcal{C}_5^2$ does not pack as $\tau(\mathcal{C}_5^2) = 3 > 2 = \nu(\mathcal{C}_5^2)$.[5] $\square$

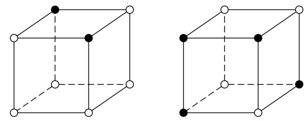
There are smaller examples of minimally non-polar sets whose cuboids are not minimally non-packing clutters. For example, consider the two minimally non-polar sets $P_3 := \{110, 011, 101\}$ and $S_3 := \{110, 011, 101, 111\}$ displayed below. Notice that $\mathrm{ind}(P_3) = \mathrm{ind}(S_3) = \Delta_3$, so $\mathrm{cuboid}(P_3)$ and $\mathrm{cuboid}(S_3)$ are *not* minimally



P_3 S_3

non-packing. As another example, consider the minimally non-polar set $\{1010, 0110, 0001, 0011, 1011, 1101\}$ displayed below. Even though all of the induced clutters of this set have the packing property, its cuboid has a

[5]The proof shows that $\{C_k : k \geq 5\}$ are not cube-ideal. In particular, strict polarity does not imply cube-idealness.



proper Q_6 minor, so it is *not* minimally non-packing. So given a minimally non-polar set, when is the cuboid minimally non-packing? Recall that S is critically non-polar if it is strictly non-polar and, for each $i \in [n]$, both the 0- and 1-restrictions of S over coordinate i have antipodal points. By Proposition 3.3,

Remark 3.5. *Critical non-polarity implies minimal non-polarity.*

Observe that $\{R_{k,1} : k \geq 1\} \cup \{R_k : k \geq 5\}$ are in fact critically non-polar sets.

Theorem 3.6. *Take an integer $n \geq 3$ and a strictly non-polar set $S \subseteq \{0,1\}^n$. Then the following statements are equivalent:*

(i) $\mathrm{cuboid}(S)$ *is minimally non-packing,*
(ii) S is critically non-polar, and the induced clutters of S have the packing property.

Proof. **(i)** $\Rightarrow$ **(ii)**: Since $\mathrm{cuboid}(S)$ is minimally non-packing, its proper minors – including all of the induced clutters by Remark 1.7 – have the packing property. To prove that S is critically non-polar, take a coordinate $i \in [n]$. As $\mathrm{cuboid}(S)/(2i-1)$ has covering number two, it has two disjoint members, which correspond to antipodal points in the 1-restriction of S over coordinate i. Similarly, as $\mathrm{cuboid}(S)/2i$ has covering number two, it has two disjoint members, which correspond to antipodal points in the 0-restriction of S over coordinate i. Thus, S is critically non-polar. **(ii)** $\Rightarrow$ **(i)**: By Remark 2.11, the induced clutters of the minors of S also have the packing property. Hence, since proper restrictions of S are strictly polar, it follows from Theorem 1.11 that for each $i \in [n]$, $\mathrm{cuboid}(S) \setminus (2i-1)/2i$ and $\mathrm{cuboid}(S) \setminus 2i/(2i-1)$ have the packing property, implying in turn that all proper deletion minors of $\mathrm{cuboid}(S)$ have the packing property. It remains to show that for each nonempty $J \subseteq [2n]$, $\mathrm{cuboid}(S)/J$ packs. If $J \cap \{2i-1, 2i\} \neq \emptyset$ for each $i \in [n]$, then $\mathrm{cuboid}(S)/J$ is a minor of an induced clutter of S by Remark 1.7, so $\mathrm{cuboid}(S)$ packs. Otherwise, $\mathrm{cuboid}(S)/J$ has covering number two. Take a coordinate $j \in [n]$ such that $J \cap \{2j-1, 2j\} \neq \emptyset$. Since both the 0- and 1-restrictions of S over coordinate j have antipodal points, it follows that both $\mathrm{cuboid}(S)/(2j-1)$ and $\mathrm{cuboid}(S)/2j$ have disjoint members; one of these two pairs of disjoint members corresponds to a pair of disjoint members in $\mathrm{cuboid}(S)/J$, so $\mathrm{cuboid}(S)/J$ packs, as required. □

Therefore, together with Theorems 1.3 and 1.6, Theorem 3.6 implies that if $\mathrm{cuboid}(S)$ is minimally non-packing, then S is cube-ideal and critically non-polar. And Conjecture 1.15 predicts that the converse also holds. Moving forward, we will need the following result:

Proposition 3.7. *Take an integer $n \geq 3$ and a critically non-polar set $S \subseteq \{0,1\}^n$. Then every proper minor of* $\mathrm{cuboid}(S)$ *has a cover of cardinality one, or two disjoint members. In particular, every proper minimally non-packing minor of* $\mathrm{cuboid}(S)$*, if any, has covering number at least three.*

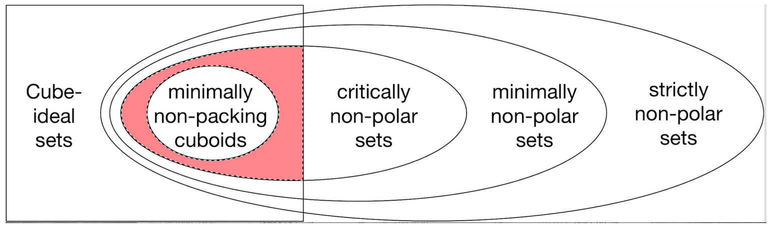


Figure 6. Conjecture 1.15 predicts that the shaded region is empty.

Proof. Let $\mathcal{C}$ be a proper minor of $\text{cuboid}(S)$ such that $\tau(\mathcal{C}) \geq 2$ and every element of $\mathcal{C}$ is contained in a member. It suffices to show that $\mathcal{C}$ has two disjoint members. To this end, pick disjoint $I, J \subseteq [2n]$ such that $\text{cuboid}(S) \setminus I/J = \mathcal{C}$. As $\tau(\mathcal{C}) \geq 2$, for each $i \in [n]$ such that $I \cap \{2i-1, 2i\} \neq \emptyset$, we must have that $J \cap \{2i-1, 2i\} \neq \emptyset$. Assume in the first case that $I \neq \emptyset$. Then by Remark 1.4, there is a proper restriction S' of S such that $\mathcal{C}$ is a minor of $\text{cuboid}(S')$. As S is critically non-polar, it is also strictly non-polar, so S' is strictly polar. Thus, $\mathcal{C}$ has disjoint members by Proposition 3.1. Assume in the remaining case that $I = \emptyset$. As $\mathcal{C}$ is a proper minor of $\text{cuboid}(S)$, $J \neq \emptyset$. Take a coordinate $j \in [n]$ such that $J \cap \{2j-1, 2j\} \neq \emptyset$. Since both the 0- and 1-restrictions of S over coordinate j have antipodal points, it follows that both $\text{cuboid}(S)/(2j-1)$ and $\text{cuboid}(S)/2j$ have disjoint members; one of these two pairs of disjoint members corresponds to a pair of disjoint members in $\text{cuboid}(S)/J = \mathcal{C}$, as required. $\square$

Figure 6 is a Venn diagram of cube-ideal sets, the various non-polar sets studied in this section, as well as minimally non-packing cuboids.

3.3. **Testing strict polarity in polynomial time.** We will need the following tool:

Lemma 3.8. *Take an integer $n \geq 3$ and a strictly non-polar set $S \subseteq \{0,1\}^n$. Then there exist distinct points $a, b \in S$ such that for $I := \{i \in [n] : a_i = b_i\}$ the following statement holds: for every $x \in S$, either $x_i = a_i$ for all $i \in I$, or $x_i = 1 - a_i$ for all $i \in I$. In particular, S has three points that do not all agree on a coordinate.*

Proof. Consider the incidence matrix $M(\text{cuboid}(S))$, whose column labels are $[2n]$. After possibly relabeling and twisting the elements of S, we may assume that

(1) among all the columns in $M(\text{cuboid}(S))$, column 1 has the maximum number of zeros, and
(2) for each $j \in \{2, \ldots, n\}$, there is a point $x \in S$ such that $x_1 = 0$ and $x_j = 0$.

Let $I \subseteq [n]$ be the set of coordinates i such that $S \subseteq \{x \in \{0,1\}^n : x_i = x_1\}$. Notice that $1 \in I$, and since S is not polar, $I \neq [n]$. Let $S' \subseteq \{0,1\}^{[n]-I}$ be obtained from S after 0-restricting the coordinates in I. As S is strictly non-polar, and $I \neq \emptyset$, it follows that S' is polar.

Claim. *S' has antipodal points.*

Proof of Claim. Suppose otherwise. Since S' is polar, there exist an $a \in \{0,1\}$ and a coordinate $j \in [n] - I$ such that

$$S' \subseteq \left\{y \in \{0,1\}^{[n]-I} : y_j = a\right\}.$$

Together with our choice of I, this implies that

for each $x \in S$: if $x_1 = 0$ then $x_j = a$.

Thus by (2) we must have that $a = 0$. Hence, in the incidence matrix $M(\text{cuboid}(S))$, column $2j - 1$ has just as many zeros as column 1, so by (1),

for each $x \in S$: if $x_1 = 1$ then $x_j = 1$.

But then j must have belonged to I, a contradiction. $\Diamond$

Let a', b' be antipodal points of S', and let a, b be the corresponding points in S – these are the desired points. Moreover, since the points in S do not all agree on a coordinate, there exists a point $c \in S - \{a, b\}$ such that $c_i = 1 - a_i$ for all $i \in I$. In particular, the points a, b, c do not all agree on a coordinate. $\square$

The first consequence of Lemma 3.8 is Theorem 1.12, which provides a polynomial time characterization of strictly polar sets:

Proof of Theorem 1.12. Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. We need to show that the following statements are equivalent:

(i) S is not strictly polar,

(ii) there are distinct points $a, b, c \in S$ such that the smallest restriction of S containing them is not polar.

(ii) ⇒ (i) holds trivially. **(i) ⇒ (ii)**: Let $S' \subseteq \{0,1\}^J$ be a strictly non-polar restriction of S. By Lemma 3.8, there exist points a', b', c' of S' that do not all agree on a coordinate of J. The points a', b', c' correspond naturally to some points a, b, c of S, respectively, and as a', b', c' do not all agree on a coordinate in J, it follows that S' is the smallest restriction of S containing a, b, c. Thus, since S' is not polar, (ii) holds.

We will next show that in time $O(n|S|^4)$, one can certify whether or not S is strictly polar. Since (i) and (ii) are equivalent, it suffices to test (ii). For any three points a, b, c in S, it takes time $O(n|S|)$ to determine whether or not the smallest restriction of S containing a, b, c is polar. As a result, it takes time $O(n|S|^4)$ to test (ii), as required. $\square$

For points $a, b \in \{0,1\}^n$, denote by $\text{dist}(a, b)$ the number of coordinates a and b differ on, i.e. $\text{dist}(a, b)$ is the Hamming distance between a and b. Another consequence of Lemma 3.8 is the following:

Theorem 3.9. *Take an integer $n \geq 3$ and a strictly non-polar set $S \subseteq \{0,1\}^n$. Then either there are feasible points at distance $n - 1$, or $M(\text{cuboid}(S))$ has two identical columns.*

Proof. By Theorem 3.8, there are distinct points a, b such that for $I := \{i \in [n] : a_i = b_i\}$, the following statement holds: for each $x \in S$, either $x_i = a_i$ for all $i \in I$, or $x_i = 1 - a_i$ for all $i \in I$. Since S is not polar, it follows that $I \neq \emptyset$. If $|I| = 1$, then $\text{dist}(a, b) = n - 1$. Otherwise, $|I| \geq 2$. Pick distinct coordinates $i, j \in I$. Then for each $x \in S$, either

- $x_i = a_i$ and $x_j = a_j$, or
- $x_i = 1 - a_i$ and $x_j = 1 - a_j$.

If $a_i = a_j$, then $x_i = x_j$ for all $x \in S$, so columns $2i-1, 2j-1$ of $M(\mathrm{cuboid}(S))$ are identical. Otherwise, $a_i + a_j = 1$, so $x_i + x_j = 1$ for all $x \in S$, so columns $2i-1, 2j$ of $M(\mathrm{cuboid}(S))$ are identical, as required. □

As a result, if S is a strictly non-polar set such that $M(\mathrm{cuboid}(S))$ does not have identical columns, then there is a coordinate $i \in [n]$ such that either the 0- or 1-restriction of S over i has antipodal points.

Question 3.10. *If S is a strictly non-polar set such that $M(\mathrm{cuboid}(S))$ does not have identical columns, is S necessarily minimally non-polar?*

3.4. **Seymour's max-flow min-cut theorem.** Here we characterize when an affine binary space is strictly polar. But first, let us prove that if an affine binary space is strictly polar, then its cuboid has the packing property. We will need the following observation:

Remark 3.11. *The cuboid of an affine binary space is a binary clutter.*

Proof. Take an integer $n \geq 1$ and an affine binary space $S \subseteq \{0,1\}^n$. Take an odd number of points $a^1, \ldots, a^k \in S$. Since k is odd, it follows that $a := a^1 \triangle a^2 \triangle \cdots \triangle a^k \in S$ and

$$\begin{aligned}\triangle_{i=1}^k (a_1^i, 1-a_1^i, \ldots, a_n^i, 1-a_n^i) &= \triangle_{i=1}^k (a_1^i, 1\triangle a_1^i, \ldots, a_n^i, 1\triangle a_n^i)\\ &= (a_1, 1\triangle a_1, \ldots, a_n, 1\triangle a_n)\\ &= (a_1, 1-a_1, \ldots, a_n, 1-a_n).\end{aligned}$$

As a result, the symmetric difference of any odd number of members of $\mathrm{cuboid}(S)$ *is* also a member. In particular, $\mathrm{cuboid}(S)$ is a binary clutter. □

We also need the following seminal result providing an exact co-NP characterization of the binary clutters with the packing property:

Theorem 3.12 (equivalent to [36], Theorem on page 209, also see [18])**.** *Let $\mathcal{C}$ be a binary clutter. Then $\mathcal{C}$ has the packing property if, and only if, it has no Q_6 minor.*

In particular, Q_6 is the only minimally non-packing clutter that is binary, thereby verifying the $\tau = 2$ Conjecture for binary clutters. This theorem also proves the Polarity Conjecture for affine binary spaces:

Corollary 3.13. *Take an affine binary space S. If S is strictly polar, then $\mathrm{cuboid}(S)$ has the packing property.*

Proof. Since S is strictly polar, it follows from Proposition 3.1 (iii) that every minor of $\mathrm{cuboid}(S)$ has a cover of cardinality one, or two disjoint members. In particular, $\mathrm{cuboid}(S)$ does not have a Q_6 minor. By Remark 3.11, $\mathrm{cuboid}(S)$ is a binary clutter, so by Theorem 3.12, $\mathrm{cuboid}(S)$ has the packing property, as required. □

Let us now characterize when an affine binary space is strictly polar. Denote by H_3 the graph on two vertices a, b and three parallel edges whose ends are a, b. Observe that the cycle matroid $M(H_3)$ of this graph is the binary matroid associated with $R_{1,1}$. Using this observation we prove the following, which for convenience is stated only for binary spaces:

Theorem 3.14. *For a binary space S, the following statements are equivalent:*

(i) *S is strictly polar,*
(ii) *S does not have an $R_{1,1}$ minor,*
(iii) *$S = \langle a_1, \ldots, a_k \rangle \pmod 2$ for some points $a_1, \ldots, a_k$ of pairwise disjoint supports.*

Proof. Let M be the binary matroid associated with S. **(i)** $\Rightarrow$ **(ii)** follows from Remark 1.10 and the fact that $R_{1,1}$ is not polar. **(ii)** $\Rightarrow$ **(iii)**: If $S = \{\mathbf{0}\}$ then we are done. We may therefore assume that $S \neq \{\mathbf{0}\}$.

Claim. *If M has two circuits that intersect, then M has an $M(H_3)$ minor.*

Proof of Claim. Among all intersecting pairs of circuits of M, pick intersecting circuits C, C' such that $C \cup C'$ is minimal. We claim the following:

> ($\star$) Take a subset $C'' \subseteq C \cup C'$ such that $C'' \notin \{\emptyset, C \triangle C'\}$. If C'' is a circuit, then $C'' \in \{C, C'\}$.

Since $C \triangle C' \neq C''$, $C \triangle C'$ is a cycle and C'' is a circuit, it follows that $C \triangle C' \not\subseteq C''$. As a result, either $C'' \cup C \subsetneq C \cup C'$ or $C'' \cup C' \subsetneq C \cup C'$. However, $C'' \cap C \neq \emptyset$ and $C'' \cap C' \neq \emptyset$, so the minimality of $C \cup C'$ implies that $C'' \in \{C, C'\}$. This proves ($\star$). Let $P_1 := C \cap C'$, $P_2 := C - C'$ and $P_3 := C' - C$. Then ($\star$) implies that the only cycles contained in $C \cup C'$ are $P_1 \cup P_2, P_2 \cup P_3, P_3 \cup P_1$. As a result, $M|(C \cup C')$ is the cycle matroid of a graph G whose edges can be partitioned into three internally vertex-disjoint paths P_1, P_2, P_3 with the same ends. Clearly, G has H_3 as a minor, implying in turn that $M|(C \cup C')$, and therefore M, has an $M(H_3)$ minor, as required. $\Diamond$

Since the circuits of M generate its cycle space, it follows that

$$S = \langle \chi_C : C \text{ is a circuit of } M \rangle \pmod 2.$$

Moreover, S has no $R_{1,1}$ minor, so by Remark 2.10, we get that M has no $M(H_3)$ minor. Thus the claim above implies that every pair of circuits of M are pairwise disjoint, so the generators above have pairwise disjoint supports, so (iii) follows. **(iii)** $\Rightarrow$ **(i)**: By Remark 2.10, it suffices to show that the binary space associated with every minor of M is polar. Let N be a minor of M. Observe that by (iii), M is the cycle matroid of a graph that is the vertex-disjoint union of loops, circuits and paths. As a result, N is also the cycle matroid of a graph that is the vertex-disjoint union of loops, circuits and paths. Hence, the binary space R associated with N can be written as

$$R = \langle b_1, \ldots, b_\ell \rangle \pmod 2$$

for some points $b_1, \ldots, b_\ell$ of pairwise disjoint supports. If $b_1 + \cdots + b_\ell = \mathbf{1}$, then b_1 and $b_2 + \cdots + b_\ell$ are antipodal points in R. Otherwise, all the points in R agree on a coordinate (which is set to 0). Either way, we see that R is polar, as required. $\square$

As a result, $R_{1,1}$ is the only binary space that is minimally non-polar.

4. The Polarity Conjecture

In this section, we prove Proposition 1.2 and Theorem 1.14, and show that the Polarity Conjecture implies Conjecture 1.15. We will also prove the $\tau = 2$ Conjecture for clutters over at most 8 elements, and the Polarity Conjecture and Conjecture 1.15 for sets of degree at most 8. Moreover, we will see how the Replication Conjecture of Conforti and Cornuéjols [7] can be reduced to cuboids.

4.1. **Up-monotone sets and the Replication Conjecture.** Here we introduce a concept needed for the proof of Theorem 1.14. Take an integer $n \geq 1$ and a subset $S \subseteq \{0,1\}^n$. We say that S is *up-monotone* if for all points $x, y \in \{0,1\}^n$ such that $x \geq y$, if y is feasible then so is x.

Remark 4.1. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. Then S is up-monotone if, and only if, there is a clutter $\mathcal{C}$ over ground set $[n]$ such that*

$$S = \left\{\chi_C : C \subseteq [n] \text{ contains a member of } \mathcal{C}\right\}.$$

Proof. ($\Rightarrow$) Take an up-monotone set S. Then $\mathcal{C} := \operatorname{ind}(S)$ is the desired clutter. ($\Leftarrow$) follows immediately from the construction of S. $\square$

Thus, there is a bijection between up-monotone subsets of a hypercube and clutters. Notice that the clutter associated with an up-monotone set is simply its induced clutter with respect to the origin. We say that S is *down-monotone* if for all points $x, y \in \{0,1\}^n$ such that $x \geq y$, if x is feasible then so is y.

Theorem 4.2. *Take an integer $n \geq 1$. Let $S \subseteq \{0,1\}^n$ be an up-monotone set and let $\mathcal{C} := \operatorname{ind}(S)$. Then $\overline{S}\triangle\mathbf{1}$ is up-monotone and $\operatorname{ind}(\overline{S}\triangle\mathbf{1}) = b(\mathcal{C})$.*

Proof. Since S is up-monotone, $\overline{S}$ is down-monotone, so $R := \overline{S}\triangle\mathbf{1}$ is up-monotone. To see that $\operatorname{ind}(R)$ is the blocker of $\operatorname{ind}(S)$, take a point $x = \chi_B \in \{0,1\}^n$. Then $x \in R \Leftrightarrow \mathbf{1} - x \notin S \Leftrightarrow \overline{B}$ does not contain a member of $\operatorname{ind}(S)$ (as S is up-monotone) $\Leftrightarrow B$ intersects every member of $\operatorname{ind}(S) \Leftrightarrow B$ contains a member of $b(\operatorname{ind}(S))$. This chain of equivalent statements implies that $\operatorname{ind}(R)$ is the blocker of $\operatorname{ind}(S)$. $\square$

As a result, two monotone sets that are complements of each other can equivalently be thought of as two clutters that are blockers of each other.

Theorem 4.3. *Let $\mathcal{C}$ be a clutter and let S be the associated up-monotone set. Then $\mathcal{C}$ is an ideal clutter if, and only if, S is a cube-ideal set.*

Proof. ($\Rightarrow$) Assume first that $\mathcal{C}$ is ideal. Then $b(\mathcal{C})$ is also ideal, so the set covering polytope $P(b(\mathcal{C}))$ is integral. By Proposition 1.1, the vertices of $P(b(\mathcal{C}))$ are the incidence vectors of the covers of $b(\mathcal{C})$, i.e. the points in S. Hence, $P(b(\mathcal{C})) = \operatorname{conv}(S)$, implying in turn that S is cube-ideal, as required. ($\Leftarrow$) Assume conversely that S is cube-ideal. Then by Theorem 1.8, the induced clutter $\operatorname{ind}(S) = \mathcal{C}$ is an ideal clutter, as required. $\square$

As a consequence,

Corollary 4.4. *Let S be an up-monotone set. If S is cube-ideal, then so is $\overline{S}$.*

Proof. Assume that S is cube-ideal. By Theorem 4.3, $\mathrm{ind}(S)$ is an ideal clutter. Then the blocker of $\mathrm{ind}(S)$, which is $\mathrm{ind}(\overline{S}\triangle\mathbf{1})$ by Theorem 4.2, is also ideal. Another application of Theorem 4.3 tells us that the up-monotone set $\overline{S}\triangle\mathbf{1}$ is cube-ideal, so its twisting $\overline{S}$ is cube-ideal, as required. □

The analogue of Theorem 4.3 for the packing property also holds. To see this, we will need the following tool, which is also needed in the next section:

Proposition 4.5. *Let $\mathcal{C}$ be a clutter, where every minor has a cover of cardinality one, or two disjoint members. Let S be the corresponding up-monotone set. Then S is strictly polar.*

Proof. Let E be the ground set of $\mathcal{C}$. Take disjoint $I, J \subseteq E$. Let $S' \subseteq \{0,1\}^{E-(I\cup J)}$ be obtained from S after 0-restricting the coordinates I and 1-restricting the coordinates J. It suffices to show that S' is polar. Notice that S' is an up-monotone set whose corresponding clutter is $\mathcal{C} \setminus I/J$. By assumption, either $\mathcal{C} \setminus I/J$ has a cover of cardinality one, or two disjoint members. Since S' is up-monotone, this implies that either the points in S' all agree on a coordinate, or S' contains antipodal points. Hence, S' is polar, as required. □

An element of a clutter is *free* if it is not contained in any member. We leave the following as an exercise for the reader:

Remark 4.6. *Take an integer $n \geq 1$, an up-monotone set $S \subseteq \{0,1\}^n$, and a point $x \in \{0,1\}^n$. Then $\mathrm{ind}(S\triangle x)$ is, after deleting free elements, equal to $\mathrm{ind}(S)/\{i \in [n] : x_i = 1\}$.*

Using the preceding two results, we prove the following analogue of Theorem 4.3:

Theorem 4.7. *Let $\mathcal{C}$ be a clutter and let S be the associated up-monotone set. Then $\mathcal{C}$ has the packing property if, and only if, $\mathrm{cuboid}(S)$ has the packing property.*

Proof. ($\Leftarrow$) is immediate as $\mathcal{C}$ is a minor of $\mathrm{cuboid}(S)$. ($\Rightarrow$) Conversely, assume that $\mathcal{C}$ has the packing property. It follows from Proposition 4.5 that S is strictly polar. Thus, to prove that $\mathrm{cuboid}(S)$ has the packing property, it suffices by Theorem 1.11 to prove that the induced clutters of S have the packing property. After deleting free elements, the induced clutters of S are simply contraction minors of $\mathcal{C}$ by Remark 4.6, so they all have the packing property, as required. Hence, $\mathrm{cuboid}(S)$ has the packing property. □

An immediate, but important, consequence of this result is that the following conjecture

(?) A cuboid with the packing property has the max-flow min-cut property.[6] (?)

is equivalent to the Replication Conjecture of Conforti and Cornuéjols [7]:

(?) A clutter with the packing property has the max-flow min-cut property. (?)

(So Conjecture 3.4 of [16] is just as strong as the Replication Conjecture.)

[6] A clutter has the *max-flow min-cut property* if the corresponding set covering program (P) is totally dual integral.

4.2. **The $\tau = 2$ Conjecture is equivalent to the Polarity Conjecture.** We will need the following remark:

Remark 4.8. *Take an integer $k \geq 0$. If a set has degree at most k, then so does every minor of it.*

We also need the following remark:

Remark 4.9. *A minimally non-packing clutter has no member of cardinality one.*

Proof. If a non-packing clutter $\mathcal{C}$ has a member of the form $\{e\}$, then $\mathcal{C} \setminus e$ is also a non-packing clutter. This proves the remark. □

Using these two remarks, we prove the following:

Theorem 4.10. *Take an integer k such that every ideal minimally non-packing clutter over at most k elements has covering number two. Then the following statements hold:*

(1) If S is cube-ideal and has degree at most k, then every minimally non-packing minor of $\mathrm{cuboid}(S)$ has covering number two.
(2) If S is cube-ideal, strictly polar and has degree at most k, then $\mathrm{cuboid}(S)$ has the packing property.
(3) If S is cube-ideal, critically non-polar and has degree at most k, then $\mathrm{cuboid}(S)$ is minimally non-packing.

Proof. **(1)** Suppose for a contradiction that for some disjoint $I, J \subseteq [2n]$, the minor $\mathcal{C} := \mathrm{cuboid}(S) \setminus I/J$ is a minimally non-packing clutter such that $\tau(\mathcal{C}) \geq 3$. Then for each $i \in [n]$, $J \cap \{2i-1, 2i\} \neq \emptyset$, implying in turn that $\mathcal{C}$ is a minor of an induced clutter of S by Remark 1.7. By Remark 2.11, $\mathcal{C}$ is an induced clutter of a minor $S' \subseteq \{0,1\}^m$ of S, where m is the number of elements of $\mathcal{C}$. After possibly twisting S, and S' accordingly, we may assume that $\mathcal{C} = \mathrm{ind}(S')$. Since S is cube-ideal, it follows from Remark 1.5 that S' is cube-ideal, so by Theorem 1.8, $\mathcal{C}$ is ideal. By Remark 4.9, $\mathcal{C}$ has no member of cardinality (at most) one, so $\mathbf{0}, e_1, \ldots, e_m \notin S'$, so S' has degree m. As S has degree at most k, S' has degree at most k by Remark 4.8, so $m \leq k$. As a consequence, $\mathcal{C}$ is an ideal minimally non-packing clutter over at most k elements, so our hypothesis implies that $\tau(\mathcal{C}) = 2$, a contradiction.

(2) As S is cube-ideal and has degree at most k, (1) implies that every minimally non-packing minor of $\mathrm{cuboid}(S)$, if any, has covering number two. As S is strictly polar, Proposition 3.1 implies that every minor of $\mathrm{cuboid}(S)$ has covering number one, or two disjoint members. Put together, we see that $\mathrm{cuboid}(S)$ has no minimally non-packing minor, so it has the packing property.

(3) By (1), every minimally non-packing minor of $\mathrm{cuboid}(S)$ has covering number two, and by Proposition 3.7, every proper minimally non-packing minor of $\mathrm{cuboid}(S)$ has covering number at least three. Put together, these facts imply that $\mathrm{cuboid}(S)$ does not have a proper minimally non-packing minor. Thus, as $\mathrm{cuboid}(S)$ does not pack, it must be minimally non-packing. □

We will see in the next section that every ideal minimally non-packing clutter over at most $k = 8$ elements has covering number two. For now, we are ready to prove Theorem 1.14, stating that the $\tau = 2$ Conjecture and the Polarity Conjecture are equivalent:

Proof of Theorem 1.14. Assume first that the $\tau = 2$ Conjecture is true, that is, every ideal minimally non-packing clutter has covering number two. It then follows from Theorem 4.10 (2) that whenever S is cube-ideal and strictly polar, then $\mathrm{cuboid}(S)$ has the packing property, so the Polarity Conjecture is true. Assume conversely that the $\tau = 2$ Conjecture is false, that is, there is an ideal minimally non-packing clutter $\mathcal{C}$ such that $\tau(\mathcal{C}) \geq 3$. Then every proper minor of $\mathcal{C}$ packs. Moreover, for an arbitrary element e, $\tau(\mathcal{C} \setminus e) \geq 2$, so $\mathcal{C} \setminus e$ has two disjoint members, implying in turn that $\mathcal{C}$ has two disjoint members. Thus,

every minor of $\mathcal{C}$ has a cover of cardinality one, or two disjoint members.

Let S be the up-monotone set associated with $\mathcal{C}$. It then follows from Theorem 4.3 and Proposition 4.5 that S is cube-ideal and strictly polar. Since $\mathcal{C} = \mathrm{ind}(S)$, $\mathcal{C}$ is a minor of $\mathrm{cuboid}(S)$, so $\mathrm{cuboid}(S)$ does not have the packing property. Hence, the Polarity Conjecture is false. Thus, the $\tau = 2$ Conjecture and the Polarity Conjecture are equivalent. □

Moreover, as an immediate application of Theorem 4.10 (3),

Corollary 4.11. *If the $\tau = 2$ Conjecture is true, then so is Conjecture 1.15. That is, if every ideal minimally non-packing clutter has covering number two, then the cuboid of every cube-ideal and critically non-polar set is minimally non-packing.*

4.3. **Q_6 is the only ideal non-packing clutter over at most 6 elements.** Here we prove Proposition 1.2, for which we need a few tools. Take an integer $n \geq 3$. A *delta of dimension* n is the clutter over ground set $[n]$ whose members are

$$\Delta_n := \big\{\{1,2\},\{1,3\},\ldots,\{1,n\},\{2,3,\ldots,n\}\big\}.$$

Δ_n does not pack as $\tau(\Delta_n) = 2 > 1 = \nu(\Delta_n)$, and more importantly, it is non-ideal as $\begin{pmatrix}\frac{n-2}{n-1} & \frac{1}{n-1} & \cdots & \frac{1}{n-1}\end{pmatrix}$ is an extreme point of the corresponding set covering polyhedron.

Theorem 4.12 ([2], Corollary 2.6)**.** *Let $\mathcal{C}$ be a clutter that has members of the form $\{e,f\}, C_e, C_f$ such that $C_e \cap \{e,f\} = \{e\}, C_f \cap \{e,f\} = \{f\}$. Then at least one of the following statements holds: (i) $C_e \cap C_f = \emptyset$, (ii) $(C_e \cup C_f) - \{e,f\}$ contains a member, or (iii) $\mathcal{C}$ has a delta minor through e and f.*

Using this tool we prove the following:

Theorem 4.13. *Let $\mathcal{C}$ be an ideal minimally non-packing clutter over ground set E such that $|E| \leq 8$. Then $\tau(\mathcal{C}) = 2$.*

Proof. Let us write the primal-dual pair

$$(P)\quad \begin{array}{ll} \min & \mathbf{1}^\top x \\ \text{s.t.} & x(C) \geq 1 \quad C \in \mathcal{C} \\ & x \geq \mathbf{0} \end{array} \qquad\qquad (D)\quad \begin{array}{ll} \max & \mathbf{1}^\top y \\ \text{s.t.} & \sum (y_C : e \in C \in \mathcal{C}) \leq 1 \quad e \in E \\ & y \geq \mathbf{0}. \end{array}$$

Suppose for a contradiction that $\tau := \tau(\mathcal{C}) \geq 3$. Then every integer feasible solution of (P) has objective value at least 3, so as $\mathcal{C}$ is ideal, every feasible solution of (P) has value at least 3.

Claim 1. *There is a member of cardinality two.*

Proof of Claim. By Remark 4.9, every member has cardinality at least two. Consider the point $\bar{x} := (\frac{1}{3}, \ldots, \frac{1}{3}) \in \mathbb{R}^E$. Then $\mathbf{1}^\top \bar{x} < 3$ as $|E| \leq 8$, so $\bar{x}$ cannot be a feasible solution of (P). So $\mathcal{C}$ has a member of cardinality two. $\Diamond$

Claim 2. *For every member C of cardinality two, there is a minimum cover B such that $C \subseteq B$.*

Proof of Claim. Since $\mathcal{C}$ does not have τ disjoint members, $\mathcal{C} \setminus C$ does not have $\tau - 1$ disjoint members. Thus, $\tau(\mathcal{C} \setminus C) \leq \tau - 2$ as $\mathcal{C} \setminus C$ packs, so there is a cover B of $\mathcal{C}$ such that $|B - C| \leq \tau - 2$. Since $|C| = 2$ and $|B| \geq \tau$, it follows that B is a minimum cover and $C \subseteq B$. $\Diamond$

Let $y^\star \in \mathbb{R}^{\mathcal{C}}_+$ be an optimal solution for (D). As $\mathcal{C}$ is ideal, it follows from LP Strong Duality that $y^\star$ has objective value τ, i.e. $\sum(y^\star_C : C \in \mathcal{C}) = \tau$.

Claim 3. *For every $C \in \mathcal{C}$, $y^\star_C < 1$.*

Proof of Claim. Suppose for a contradiction that $y^\star_C = 1$. Then $\sum(y^\star_{C'} : C' \in \mathcal{C} - \{C\}) = \tau - 1$, and as $y^\star$ is feasible for (D), $C' \cap C = \emptyset$ for all $C' \in \mathcal{C} - \{C\}$ such that $y^\star_{C'} > 0$. As a result, $y^\star$ certifies the inequality $\tau(\mathcal{C} \setminus C) \geq \tau - 1$. As $\mathcal{C} \setminus C$ packs, it has $\tau - 1$ disjoint members; together with C, we get τ disjoint members in $\mathcal{C}$, a contradiction. $\Diamond$

By Claim 1, there are distinct elements $e, f \in E$ such that $\{e, f\}$ is a member. By Claim 2, there is a minimum cover B such that $\{e, f\} \subseteq B$. Notice that B yields an optimal solution to (P). As e, f belong to a minimum cover, the Complementary Slackness conditions imply that

$$\sum(y^\star_C : C \ni e) = \sum(y^\star_C : C \ni f) = 1.$$

By Claim 3, there are distinct members C_1, C_2 such that $e \in C_1 \cap C_2$ and $y^\star_{C_1}, y^\star_{C_2}$ are non-zero, and there are distinct members C_3, C_4 such that $f \in C_3 \cap C_4$ and $y^\star_{C_3}, y^\star_{C_4}$ are non-zero. Applying the Complementary Slackness conditions again, we see that each one of C_1, C_2, C_3, C_4 intersect every minimum cover exactly once. As a result, $C_1 \cap B = C_2 \cap B = \{e\}$ and $C_3 \cap B = C_4 \cap B = \{f\}$, and by Claim 2, $|C_i| \geq 3$ for $i \in [4]$. In particular, as $|E - B| \leq 5$, we have that $C_i \cap C_j \neq \emptyset$ for some $i \in \{1, 2\}$ and $j \in \{3, 4\}$. Moreover, as $(C_i \cup C_j) - \{e, f\}$ is disjoint from the cover B, it does not contain a member. Thus, by Theorem 4.12, $\mathcal{C}$ has a delta minor, a contradiction as $\mathcal{C}$ is an ideal clutter and the deltas are non-ideal. $\square$

Thus, the $\tau = 2$ Conjecture is true for clutters over at most 8 elements. Hence, by Theorem 4.10,

Corollary 4.14. *The following statements hold:*

1. *If S is cube-ideal, strictly polar and has degree at most 8, then $\mathrm{cuboid}(S)$ has the packing property. That is, the Polarity Conjecture is true for sets of degree at most 8.*
2. *If S is cube-ideal, critically non-polar and has degree at most 8, then $\mathrm{cuboid}(S)$ is an ideal minimally non-packing clutter. That is, Conjecture 1.15 is true for sets of degree at most 8.*

Moving forward, we need the following tool:

Theorem 4.15 ([10], Theorem 3)**.** *Let $\mathcal{C}$ be an ideal minimally non-packing clutter such that $\tau(\mathcal{C}) = 2$. Then there are members of the form*

$$\begin{aligned} C_1 &= I_1 \cup I_3 \cup I_6 & \qquad C_3 &= I_2 \cup I_3 \cup I_5 \\ C_2 &= I_1 \cup I_4 \cup I_5 & \qquad C_4 &= I_2 \cup I_4 \cup I_6 \end{aligned}$$

for some partition of its ground set into nonempty parts $I_1, I_2, I_3, I_4, I_5, I_6$.

We are now ready to prove Proposition 1.2, stating that Q_6 is the only ideal non-packing clutter over at most 6 elements:

Proof of Proposition 1.2. Let $\mathcal{C}$ be an ideal minimally non-packing clutter over ground set E such that $|E| \le 6$. It suffices to show that $\mathcal{C}$ is isomorphic to Q_6. By Theorem 4.13, $\tau(\mathcal{C}) = 2$. Theorem 4.15 now tells us that $|E| = 6$, and after a possible relabeling of E, we may assume that $\{1,3,6\}, \{1,4,5\}, \{2,3,5\}, \{2,4,6\}$ are members. Since $\mathcal{C}$ is minimally non-packing, every element appears in a minimum cover (otherwise deleting the element keeps the clutter non-packing). The four distinguished members now tell us that $\{1,2\}, \{3,4\}, \{5,6\}$ are minimum covers. Now by using the fact that $\mathcal{C}$ does not have disjoint members, it can be readily checked that $\mathcal{C} = \big\{\{1,3,6\}, \{1,4,5\}, \{2,3,5\}, \{2,4,6\}\big\} = Q_6$, as required. □

5. Basic binary operations

In this section, we prove Theorems 1.16, 1.17, 1.18 and 1.19. We need a few basic facts on the products and coproducts of clutters and sets.

5.1. **Products and coproducts of clutters.** Let $\mathcal{C}_1, \mathcal{C}_2$ be clutters over disjoint ground sets E_1, E_2, respectively. Define the *product* of $\mathcal{C}_1$ and $\mathcal{C}_2$ as the clutter over ground set $E_1 \cup E_2$ whose members are

$$\mathcal{C}_1 \times \mathcal{C}_2 := \big\{C_1 \cup C_2 : C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2\big\}$$

and the *coproduct* of $\mathcal{C}_1$ and $\mathcal{C}_2$ as the clutter over ground set $E_1 \cup E_2$ whose members are

$$\mathcal{C}_1 \oplus \mathcal{C}_2 := \text{ the minimal sets of } \mathcal{C}_1 \cup \mathcal{C}_2.$$

Remark 5.1. *For clutters $\mathcal{C}_1, \mathcal{C}_2$ over disjoint ground sets, the following statements hold:*

(1) $b(\mathcal{C}_1 \times \mathcal{C}_2) = b(\mathcal{C}_1) \oplus b(\mathcal{C}_2)$ and $b(\mathcal{C}_1 \oplus \mathcal{C}_2) = b(\mathcal{C}_1) \times b(\mathcal{C}_2)$,

(2) for an element e of $\mathcal{C}_1$, $(\mathcal{C}_1 \times \mathcal{C}_2) \setminus e = (\mathcal{C}_1 \setminus e) \times \mathcal{C}_2$ and $(\mathcal{C}_1 \times \mathcal{C}_2)/e = (\mathcal{C}_1/e) \times \mathcal{C}_2$,

(3) for an element e of $\mathcal{C}_1$, $(\mathcal{C}_1 \oplus \mathcal{C}_2) \setminus e = (\mathcal{C}_1 \setminus e) \oplus \mathcal{C}_2$ and $(\mathcal{C}_1 \oplus \mathcal{C}_2)/e = (\mathcal{C}_1/e) \oplus \mathcal{C}_2$.

Proof. **(1)** It suffices to show that $b(\mathcal{C}_1 \times \mathcal{C}_2) = b(\mathcal{C}_1) \oplus b(\mathcal{C}_2)$. Since every cover of $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) is clearly a cover of $\mathcal{C}_1 \times \mathcal{C}_2$, it follows that every member of $b(\mathcal{C}_1) \oplus b(\mathcal{C}_2)$ contains a member of $b(\mathcal{C}_1 \times \mathcal{C}_2)$. Conversely, take a cover B of $\mathcal{C}_1 \times \mathcal{C}_2$. We need to show that B is a cover of $\mathcal{C}_1$ or of $\mathcal{C}_2$. If B is a cover of $\mathcal{C}_1$, then we are done. Otherwise, there is a member $C_1 \in \mathcal{C}_1$ such that $B \cap C_1 = \emptyset$. Since B is a cover of $\mathcal{C}_1 \times \mathcal{C}_2$, it intersects all sets of the form $\{C_1 \cup C_2 : C_2 \in \mathcal{C}_2\}$, implying in turn that B is a cover of $\mathcal{C}_2$, as required. Thus, every member of $b(\mathcal{C}_1 \times \mathcal{C}_2)$ contains a member of $b(\mathcal{C}_1) \oplus b(\mathcal{C}_2)$. Hence, $b(\mathcal{C}_1 \times \mathcal{C}_2) = b(\mathcal{C}_1) \oplus b(\mathcal{C}_2)$. **(2)** and **(3)** are immediate. $\square$

Using this remark, the reader can easily prove the following:

Remark 5.2. *Let $\mathcal{C}_1, \mathcal{C}_2$ be clutters over disjoint ground sets. Then the following statements hold:*

(1) If $\mathcal{C}_1, \mathcal{C}_2$ are ideal, then so are $\mathcal{C}_1 \times \mathcal{C}_2$ and $\mathcal{C}_1 \oplus \mathcal{C}_2$.
(2) If $\mathcal{C}_1, \mathcal{C}_2$ pack, then so do $\mathcal{C}_1 \times \mathcal{C}_2$ and $\mathcal{C}_1 \oplus \mathcal{C}_2$.
(3) If $\mathcal{C}_1, \mathcal{C}_2$ have the packing property, then so do $\mathcal{C}_1 \times \mathcal{C}_2$ and $\mathcal{C}_1 \oplus \mathcal{C}_2$.

In particular, we have the following tool which is used in §6:

Corollary 5.3. *Let E_1, E_2, E_3 be disjoint, nonempty, finite sets, and let $\mathcal{C}$ be the clutter over ground $E_1 \cup E_2 \cup E_3$ whose members are*

$$\big\{\{e\} : e \in E_1\big\} \cup \big\{\{f, g\} : f \in E_2, g \in E_3\big\}.$$

Then $\mathcal{C}$ has the packing property.

Proof. For $i \in [3]$, let $\mathcal{C}_i$ be the clutter over ground set E_i whose members are $\big\{\{e\} : e \in E_i\big\}$. Clearly, $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ have the packing property. As a result, a repeated application of Remark 5.2 (3) implies that $\mathcal{C} = \mathcal{C}_1 \oplus \big(\mathcal{C}_2 \times \mathcal{C}_3\big)$ has the packing property, as required. $\square$

5.2. **Products and coproducts of sets.** Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Recall that

$$S_1 \times S_2 = \big\{(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} : x \in S_1 \text{ and } y \in S_2\big\}$$
$$S_1 \oplus S_2 = \big\{(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} : x \in S_1 \text{ or } y \in S_2\big\} = \overline{\overline{S_1} \times \overline{S_2}}.$$

In words, the product $S_1 \times S_2$ is obtained from S_1 after replacing each feasible point by a copy of S_2 and each infeasible point by an infeasible hypercube, and the coproduct $S_1 \oplus S_2$ is obtained from S_1 after replacing each feasible point by a feasible hypercube and each infeasible point by a copy of S_2. Notice that for $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$,

$$(S_1 \times S_2) \triangle (x, y) = (S_1 \triangle x) \times (S_2 \triangle y)$$
$$(S_1 \oplus S_2) \triangle (x, y) = (S_1 \triangle x) \oplus (S_2 \triangle y).$$

Remark 5.4. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Then, viewing $\operatorname{ind}(S_1)$ and $\operatorname{ind}(S_2)$ as clutters over disjoint ground sets, we have that*

$$\operatorname{ind}(S_1 \times S_2) = \operatorname{ind}(S_1) \times \operatorname{ind}(S_2)$$
$$\operatorname{ind}(S_1 \oplus S_2) = \operatorname{ind}(S_1) \oplus \operatorname{ind}(S_2).$$

Observe further that $\mathrm{cuboid}(S_1 \times S_2) = \mathrm{cuboid}(S_1) \times \mathrm{cuboid}(S_2)$, but $\mathrm{cuboid}(S_1 \oplus S_2)$ is not necessarily $\mathrm{cuboid}(S_1) \oplus \mathrm{cuboid}(S_2)$. However, locality ensures that the set coproduct still preserves the properties considered so far:

Proposition 5.5. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Then the following statements hold:*

(1) If S_1, S_2 are cube-ideal, then so are $S_1 \times S_2$ and $S_1 \oplus S_2$.
(2) If S_1, S_2 are strictly polar, then so are $S_1 \times S_2$ and $S_1 \oplus S_2$.
(3) If $\mathrm{cuboid}(S_1), \mathrm{cuboid}(S_2)$ have the packing property, then so do $\mathrm{cuboid}(S_1 \times S_2)$ and $\mathrm{cuboid}(S_1 \oplus S_2)$.

Proof. **(1)** Assume that S_1, S_2 are cube-ideal sets. By Theorem 1.6, $\mathrm{cuboid}(S_1), \mathrm{cuboid}(S_2)$ are ideal clutters, so by Remark 5.2 (1), $\mathrm{cuboid}(S_1) \times \mathrm{cuboid}(S_2) = \mathrm{cuboid}(S_1 \times S_2)$ is ideal, so Theorem 1.6 implies that $S_1 \times S_2$ is cube-ideal. To prove that $S_1 \oplus S_2$ are cube-ideal, it suffices by Theorem 1.8 to show that the induced clutters of $S_1 \oplus S_2$ are ideal. To this end, take $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$. Then

$$\mathrm{ind}((S_1 \oplus S_2)\triangle(x,y)) = \mathrm{ind}(S_1 \triangle x) \oplus \mathrm{ind}(S_2 \triangle y).$$

Since S_1, S_2 is cube-ideal, it follows from Theorem 1.8 that $\mathrm{ind}(S_1\triangle x), \mathrm{ind}(S_2 \triangle y)$ are ideal clutters, so by Remark 5.2 (1), $\mathrm{ind}((S_1 \oplus S_2)\triangle(x,y))$ is an ideal clutter, as required.

(2) Assume that S_1, S_2 are strictly polar. Since a restriction of $S_1 \times S_2$ (resp. $S_1 \oplus S_2$) is the product (resp. coproduct) of a restriction of S_1 and a restriction of S_2, it suffices to prove that $S_1 \times S_2$ and $S_1 \oplus S_2$ are polar. It is easy to see that $S_1 \oplus S_2$ is polar. To show that $S_1 \times S_2$ is polar, there are two cases to consider: either the points in one of S_1, S_2 all agree on a coordinate; or each one of S_1, S_2 contains antipodal points. If the points in S_1 all agree on a coordinate, it is clear that the points in $S_1 \times S_2$ also agree on a coordinate. Thus, we may assume that each one of S_1, S_2 contains antipodal points. Let $\{p_1, q_1\}$ and $\{p_2, q_2\}$ denote pairs of antipodal points in S_1 and S_2, respectively. Then the two points $p_1 \times p_2$ and $q_1 \times q_2$ are antipodal points of $S_1 \times S_2$. In either cases, we see that $S_1 \times S_2$ is polar, so we are done.

(3) Assume that $\mathrm{cuboid}(S_1), \mathrm{cuboid}(S_2)$ have the packing property. By Remark 5.2 (3), $\mathrm{cuboid}(S_1) \times \mathrm{cuboid}(S_2) = \mathrm{cuboid}(S_1 \times S_2)$ has the packing property too. To prove that $\mathrm{cuboid}(S_1 \oplus S_2)$ has the packing property, we appeal to the locality of the packing property once strict polarity is enforced. By Proposition 3.1, S_1, S_2 are strictly polar because $\mathrm{cuboid}(S_1), \mathrm{cuboid}(S_2)$ have the packing property, so by (2), $S_1 \oplus S_2$ is strictly polar. Hence, by Theorem 1.11, it suffices to show that the induced clutters of $S_1 \oplus S_2$ have the packing property; this follows from Remark 5.2 (3) and Remark 5.4. □

5.3. **Reflective products of sets.** Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Recall that the reflective product of S_1 and S_2 is

$$S_1 * S_2 = (S_1 \times S_2) \cup (\overline{S_1} \times \overline{S_2}).$$

Notice that for $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$,

$$(S_1 * S_2)\triangle(x,y) = (S_1 \triangle x) * (S_2 \triangle y).$$

Proposition 5.6. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$. Then, viewing $\mathrm{ind}(S_1)$ and $\mathrm{ind}(S_2)$ as clutters over disjoint ground sets, we have*

$$\mathrm{ind}(S_1 * S_2) = \begin{cases} \{\emptyset\} & \text{if } \mathbf{0} \in S_1 \text{ and } \mathbf{0} \in S_2 \\ \{\emptyset\} & \text{if } \mathbf{0} \in \overline{S_1} \text{ and } \mathbf{0} \in \overline{S_2} \\ \mathrm{ind}(S_1) \oplus \mathrm{ind}(\overline{S_2}) & \text{if } \mathbf{0} \in \overline{S_1} \text{ and } \mathbf{0} \in S_2 \\ \mathrm{ind}(\overline{S_1}) \oplus \mathrm{ind}(S_2) & \text{if } \mathbf{0} \in S_1 \text{ and } \mathbf{0} \in \overline{S_2}. \end{cases}$$

Proof. By the symmetry between S_1 and S_2, it suffices to prove the first and third cases. If $\mathbf{0} \in S_1$ and $\mathbf{0} \in S_2$, then $\mathbf{0} \in S_1 \times S_2 \subseteq S_1 * S_2$, so $\mathrm{ind}(S_1 * S_2) = \{\emptyset\}$. Suppose next that $\mathbf{0} \in \overline{S_1}$ and $\mathbf{0} \in S_2$. Then by Remark 5.4, $\mathrm{ind}(S_1 \times S_2) = \mathrm{ind}(S_1)$ and $\mathrm{ind}(\overline{S_1} \times \overline{S_2}) = \mathrm{ind}(\overline{S_2})$, implying in turn that $\mathrm{ind}(S_1 * S_2) = \mathrm{ind}(S_1) \oplus \mathrm{ind}(\overline{S_2})$, as required. $\square$

We are now ready to prove Theorem 1.16, stating that if $S_1, \overline{S_1}, S_2, \overline{S_2}$ are cube-ideal, then so are $S_1 * S_2, \overline{S_1 * S_2}$:

Proof of Theorem 1.16. Assume that $S_1, \overline{S_1}, S_2, \overline{S_2}$ are cube-ideal. Since $\overline{S_1 * S_2} = \overline{S_1} * S_2$, it suffices by symmetry to show that $S_1 * S_2$ is cube-ideal. Take an arbitrary $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$. By Theorem 1.8, it suffices to show that $\mathrm{ind}((S_1 * S_2) \triangle (x, y)) = \mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y))$ is an ideal clutter. By Proposition 5.6,

$$\mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y)) = \{\emptyset\} \quad \text{or} \quad \mathrm{ind}(S_1 \triangle x) \oplus \mathrm{ind}(\overline{S_2 \triangle y}) \quad \text{or} \quad \mathrm{ind}(\overline{S_1 \triangle x}) \oplus \mathrm{ind}(S_2 \triangle y).$$

Since $S_1, \overline{S_1}, S_2, \overline{S_2}$ are cube-ideal, it follows from Theorem 1.8 that $\mathrm{ind}(S_1 \triangle x), \mathrm{ind}(\overline{S_1} \triangle x), \mathrm{ind}(S_2 \triangle y)$ and $\mathrm{ind}(\overline{S_2} \triangle y)$ are ideal. Since $\overline{S_1 \triangle x} = \overline{S_1} \triangle x$ and $\overline{S_2 \triangle y} = \overline{S_2} \triangle y$, we get from Remark 5.2 (1) that $\mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y))$ is an ideal clutter, as required. $\square$

We are also ready to prove Theorem 1.17, stating that if $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1}), \mathrm{cuboid}(S_2), \mathrm{cuboid}(\overline{S_2})$ have the packing property and $S_1 * S_2$ is strictly polar, then $\mathrm{cuboid}(S_1 * S_2)$ has the packing property:

Proof of Theorem 1.17. Assume that $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1}), \mathrm{cuboid}(S_2), \mathrm{cuboid}(\overline{S_2})$ have the packing property and $S_1 * S_2$ is strictly polar. Take an arbitrary $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$. To prove that $\mathrm{cuboid}(S_1 * S_2)$ has the packing property, it suffices by Theorem 1.11 to show that $\mathrm{ind}((S_1 * S_2) \triangle (x, y)) = \mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y))$ has the packing property. By Proposition 5.6, $\mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y))$ is either

$$\{\emptyset\} \quad \text{or} \quad \mathrm{ind}(S_1 \triangle x) \oplus \mathrm{ind}(\overline{S_2 \triangle y}) \quad \text{or} \quad \mathrm{ind}(\overline{S_1 \triangle x}) \oplus \mathrm{ind}(S_2 \triangle y).$$

Since $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1}), \mathrm{cuboid}(S_2), \mathrm{cuboid}(\overline{S_2})$ have the packing property, it follows that $\mathrm{ind}(S_1 \triangle x)$, $\mathrm{ind}(\overline{S_1} \triangle x), \mathrm{ind}(S_2 \triangle y)$ and $\mathrm{ind}(\overline{S_2} \triangle y)$ have the packing property also. Since $\overline{S_1 \triangle x} = \overline{S_1} \triangle x$ and $\overline{S_2 \triangle y} = \overline{S_2} \triangle y$, we get from Remark 5.2 (3) that $\mathrm{ind}((S_1 \triangle x) * (S_2 \triangle y))$ has the packing property, as required. $\square$

We will need the following two remarks for Theorem 1.18:

Remark 5.7. *Take an integer $n \geq 3$ and a strictly non-polar set $S \subseteq \{0,1\}^n$. Then there is no set $S' \subseteq \{0,1\}^{n-1}$ such that $S \cong S' \times \{0,1\}$.*

Proof. Suppose otherwise. If the points in S' all agreed on a coordinate, then so would the points in $S'\times\{0,1\}\cong S$, and if S' contained antipodal points, then so would $S'\times\{0,1\}\cong S$. Thus, as S is not polar, S' is not polar either, a contradiction as S is strictly non-polar and S' is a proper restriction of S. $\square$

Recall that for an integer $n\geq 1$ and $S\subseteq\{0,1\}^n$, S is antipodally symmetric if for each $x\in\{0,1\}^n$, $x\in S$ if and only if $\mathbf{1}-x\in S$. Observe that if S is antipodally symmetric, then so is $\overline{S}$.

Remark 5.8. *Take an integer $n\geq 2$, an antipodally symmetric set $S\subseteq\{0,1\}^n$, and let $S'\subseteq\{0,1\}^{n-1}$ be the 0-restriction of S over coordinate n. If S' is also antipodally symmetric, then $S=S'\times\{0,1\}$.*

We are now ready to prove Theorem 1.18:

Proof of Theorem 1.18. Take integers $n_1,n_2\geq 1$ and sets $S_1\subseteq\{0,1\}^{n_1}$ and $S_2\subseteq\{0,1\}^{n_2}$, where S_1*S_2 is strictly non-polar. We need to prove four statements:

(1) $S_1,\overline{S_1},S_2,\overline{S_2}$ are nonempty.

Suppose otherwise. Then $S_1*S_2\cong S'\times\{0,1\}^k$ for some $S'\in\left\{S_1,\overline{S_1},S_2,\overline{S_2}\right\}$ and $k\in\{n_1,n_2\}$, thereby contradicting Remark 5.7.

(2) Either $n_1=1$ and S_2 is antipodally symmetric, or $n_2=1$ and S_1 is antipodally symmetric. In particular, $S_1*S_2=\overline{S_1*S_2}$.

We first prove that one of S_1,S_2 is antipodally symmetric. Suppose otherwise. Then for some $a\in S_1$ and $b\in S_2$, we have $\mathbf{1}^{n_1}-a\in\overline{S_1}$ and $\mathbf{1}^{n_2}-b\in\overline{S_2}$. But then (a,b) and $(\mathbf{1}^{n_1}-a,\mathbf{1}^{n_2}-b)$ are antipodal points in $(S_1\times S_2)\cup(\overline{S_1}\times\overline{S_2})=S_1*S_2$, a contradiction as S_1*S_2 is not polar. We may therefore assume that S_2 is antipodally symmetric. Thus, as $S_2,\overline{S_2}$ are nonempty and S_1*S_2 does not have antipodal points, neither of $S_1,\overline{S_1}$ has antipodal points. As proper restrictions of S_1*S_2, both $S_1,\overline{S_1}$ are polar, so the points in S_1 all agree on a coordinate and the points in $\overline{S_1}$ all agree on a coordinate. This implies that $S_1\cong\{0,1\}^{n_1-1}\times\{0\}$. Consequently, $S_1*S_2\cong S'\times\{0,1\}^{n_1-1}$ for some set $S'\subseteq\{0,1\}^{n_2+1}$. It now follows from Remark 5.7 that $n_1-1=0$, so $n_1=1$. In particular, $S_1\cong\overline{S_1}$, so $\overline{S_1*S_2}=\overline{S_1}*S_2\cong S_1*S_2$, thereby proving (2).

(3) S_1*S_2 is critically non-polar.

By (1) and (2), we may assume that $n_2=1$, $S_2=\{0\}\subseteq\{0,1\}^{n_2}$, and $S_1,\overline{S_1}$ are nonempty and antipodally symmetric. Let $S:=S_1*\{0\}=(S_1\times\{0\})\cup(\overline{S_1}\times\{1\})$. Since $S_1,\overline{S_1}$ are nonempty and antipodally symmetric, both the 0- and 1-restriction of S over coordinate n_1+1 have antipodal points. After a possible twisting and relabeling, it suffices to prove that the 0-restriction of S over coordinate n_1 has antipodal points. Let S_1' be the 0-restriction of S_1 over coordinate n_1. Notice that S_1' is not antipodally symmetric. For if it were, then by Remark 5.8, $S_1=S_1'\times\{0,1\}$, implying in turn that

$$S=S_1*\{0\}=(S_1'\times\{0,1\})*\{0\}\cong(S_1'*\{0\})\times\{0,1\},$$

thereby contradicting Remark 5.7. Thus, S_1' is not antipodally symmetric, implying in turn that $(S_1'\times\{0\})\cup(\overline{S_1'}\times\{1\})=S_1'*\{0\}$ has antipodal points. Since $S_1'*\{0\}$ is the 0-restriction of S over coordinate n_1, (3) follows.

(4) If $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1}), \mathrm{cuboid}(S_2), \mathrm{cuboid}(\overline{S_2})$ have the packing property, then the clutter $\mathrm{cuboid}(S_1 * S_2)$ is ideal and minimally non-packing.

Let us first prove that the induced clutters of $S = S_1 * S_2$ have the packing property. By Proposition 5.6, an induced clutter of S is the coproduct of induced clutters of $S_1, \overline{S_2}$ or of $\overline{S_1}, S_2$. Since $\mathrm{cuboid}(S_1), \mathrm{cuboid}(\overline{S_1})$, $\mathrm{cuboid}(S_2)$ and $\mathrm{cuboid}(\overline{S_2})$ have the packing property, and taking clutter coproducts preserves the packing property by Remark 5.2 (3), it follows that the induced clutters of S have the packing property. Since S is critically non-polar by (3), Theorem 3.6 tells us that $\mathrm{cuboid}(S)$ is a minimally non-packing clutter, implying in turn that $\mathrm{cuboid}(S)$ is ideal by Theorem 1.3, thereby proving (4). □

5.4. **Strict connectivity and the $R_{k,1}$'s.** Theorem 1.18 sheds light on strictly non-polar sets that are obtained by taking a reflective product, and given that their cuboids are ideal minimally non-packing clutters under certain conditions, the pressing question is: what are these strictly non-polar sets? As we know, $\{R_{k,1} : k \geq 1\} \cup \{R_k : k \geq 5\}$ are examples of such sets. Even though we are not able to explicitly describe them all, here we extract an attribute of the strictly non-polar sets, different from $\{R_{k,1} : k \geq 1\}$, that are obtained by taking a reflective product.

Take an integer $n \geq 1$. Recall that G_n is the skeleton graph of $\{0,1\}^n$. Let us start with a basic remark:

Remark 5.9. *For an integer $n \geq 1$, the following statements hold:*

(1) For distinct points $a, b, c \in \{0,1\}^n$, $\mathrm{dist}(a,b) + \mathrm{dist}(b,c) \geq \mathrm{dist}(a,c)$.

(2) For distinct points $a, b \in \{0,1\}^n$, every ab-path in G_n has at least $\mathrm{dist}(a,b)$ many edges,

(3) For distinct points $a, b \in \{0,1\}^n$, let P be an ab-path in G_n with exactly $\mathrm{dist}(a,b)$ many edges. Then P is contained in every restriction containing a, b.

Take a set $S \subseteq \{0,1\}^n$. We will refer to a path contained in $G_n[S]$ as a *feasible path*. Recall that S is connected if $G_n[S]$ is connected. For $k \geq 2$, let $A_k := \{\mathbf{0}, \mathbf{1}\} \subseteq \{0,1\}^k$, and notice that A_k is not connected. Observe that if S is connected, then for all feasible points a and b, there is a feasible ab-path, and any such path has at least $\mathrm{dist}(a,b)$ many edges. Recall that S is strictly connected if all of its restrictions are connected. The following proposition characterizes strictly connected sets:

Proposition 5.10. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$. The following statements are equivalent:*

(i) S is strictly connected,

(ii) S does not have any of $\{A_k : k \geq 2\}$ as a restriction,

(iii) for all distinct feasible points a and b, there is a feasible ab-path with $\mathrm{dist}(a,b)$ many edges.

Proof. **(i)** ⇒ **(ii)** follows immediately from the fact that none of $\{A_k : k \geq 2\}$ is connected. **(ii)** ⇒ **(iii)**: We prove this by induction on $\mathrm{dist}(a,b) \geq 1$. For the base case when $\mathrm{dist}(a,b) = 1$, the desired path consists of the edge between a, b. For the induction step, assume that $\mathrm{dist}(a,b) \geq 2$. Let $d := \mathrm{dist}(a,b)$. After possibly twisting and relabeling the coordinates, we may assume that $a = \mathbf{0}$ and $b = \sum_{i=1}^{d} e_i$.

Claim. $S \cap \{e_1, \ldots, e_d\} \neq \emptyset$.

Proof of Claim. Let c be a point in $\{x \in S : x_{d+1} = \cdots = x_n = 0, x \neq \mathbf{0}\}$ that minimizes $\operatorname{dist}(\mathbf{0}, c)$. It suffices to prove that $\operatorname{dist}(\mathbf{0}, c) = 1$. Suppose otherwise. Then for $k := \operatorname{dist}(\mathbf{0}, c) \geq 2$, the smallest restriction of S containing $\mathbf{0}, c$ is isomorphic to A_k, a contradiction. $\Diamond$

Pick $j \in [d]$ such that $e_j \in S$. Then $\operatorname{dist}(e_j, b) = d - 1$, so by the induction hypothesis, there is a feasible $e_j b$-path Q with $d-1$ many edges. Let P be the feasible ab-walk obtained by adding the edge $\mathbf{0}e_j$ to Q. Clearly, P has d many edges, and since any feasible ab-path has at least $d = \operatorname{dist}(a, b)$ many edges by Remark 5.9 (2), it follows that P is in fact a path, thereby completing the induction step. **(iii)** $\Rightarrow$ **(i)**: Take feasible points a, b. Then there is a feasible ab-path P with $\operatorname{dist}(a, b)$ many edges. Then by Remark 5.9 (3), P is contained in every restriction containing a, b, implying in turn that a, b belong to the same component in every restriction where they are present. Since this is true for all pairs of feasible points, it follows that every restriction of S is connected, so S is strictly connected. $\square$

Recall that for each $k \geq 1$, $R_{k,1} = A_{k+1} * \{0\}$ and $\overline{R_{k,1}} \cong R_{k,1}$.

Proposition 5.11. *Take integers $n_1, n_2 \geq 1$ and sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$, where $S_1, \overline{S_1}, S_2, \overline{S_2}$ are nonempty. If one of $S_1, \overline{S_1}, S_2, \overline{S_2}$ is not strictly connected, then $S_1 * S_2$ has one of $\{R_{k,1} : k \geq 1\}$ as a restriction.*

Proof. By the symmetry between S_1, S_2, we may assume that one of $S_1, \overline{S_1}$ is not strictly connected. Since $R_{k,1} \cong \overline{R_{k,1}}$, we may in fact assume that S_1 is not strictly connected. Then by Proposition 5.10 (ii), S_1 has one of $\{A_k : k \geq 2\}$ as a restriction. Since both $S_2, \overline{S_2}$ are nonempty, S_2 has $\{0\} \subseteq \{0,1\}^1$ as a restriction. As a result, $S_1 * S_2$ has one of $\{A_k * \{0\} : k \geq 2\} = \{R_{k,1} : k \geq 1\}$ as a restriction, as required. $\square$

We are now ready to prove Theorem 1.19, stating that if $S \subseteq \{0,1\}^n$ is an antipodally symmetric set such that $S * \{0\}$ is strictly non-polar and different from $\{R_{k,1} : k \geq 1\}$, then both S and $\overline{S}$ are strictly connected:

Proof of Theorem 1.19. It follows from Theorem 1.18 (1) that both $S, \overline{S}$ are nonempty, so by Proposition 5.11, both S and $\overline{S}$ are strictly connected. $\square$

6. The spectrum of strictly non-polar sets of constant degree

Here we prove Theorem 1.20, and describe a code that generates the strictly non-polar sets of degree at most 3.

6.1. **Proof of Theorem 1.20.** A graph is *triangle-free* if it has no circuit with three edges, and it is *simple* if it has no loops or parallel edges. We will need the following classic result known as Mantel's Theorem:

Theorem 6.1 ([27]). *For an integer $n \geq 3$, every triangle-free simple graph on n vertices has at most $\left\lfloor \frac{n^2}{4} \right\rfloor$ edges, and this bound is achieved only by the complete bipartite graph $K_{\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil}$.*

Recall that $P_3 = \{110, 101, 011\}$, $S_3 = \{110, 101, 011, 111\}$ and $R_{1,1} = \{000, 110, 101, 011\}$. We are now equipped to prove the following lemma:

Lemma 6.2. *Take integers $n \geq 3$, $k \in \{0, 1, \dots, n\}$ and a set $S \subseteq \{0,1\}^n$ that is not polar, has degree k, and has none of $P_3, S_3, R_{1,1}$ as a restriction. Take an infeasible point x whose set of feasible neighbors is F and whose set of infeasible neighbors is I, where $|I| = k$. Then the following statements hold:*

(1) We have that

$$\frac{(n-k-1)^2-1}{4} \leq \big|\{x\triangle y\triangle z : y, z \in F, y \neq z\} \cap S\big| \leq \big|\{x\triangle y\triangle z : y, z \in I, y \neq z\} \cap S\big| \leq \frac{k^2}{4}.$$

(2) We have that $n \leq 2k+1$.

(3) If $n = 2k+1$, then $k \geq 2$, every point in F has exactly k feasible neighbors, every point in I has exactly k infeasible neighbors, and there is a partition of I into parts I_1, I_2 such that $\big||I_1| - |I_2|\big| \leq 1$ and for distinct $y, z \in I$,

$$x\triangle y\triangle z \in S \Leftrightarrow |I_1 \cap \{y, z\}| = 1.$$

Proof. Let us start with the following claim:

Claim 1. *Every feasible point has at most k feasible neighbors.*

Proof of Claim. S is not polar, so it does not have antipodal points, implying in turn that $G_n[S]$ is isomorphic to a subgraph of $G_n[\overline{S}]$.[7] Thus, as $G_n[\overline{S}]$ has maximum degree at most k, so does $G_n[S]$, so every feasible point has at most k feasible neighbors. $\Diamond$

In particular, since $n \geq 3$ and S has no $R_{1,1}$ restriction, it follows that $k \geq 1$.

Claim 2. *There exist no $x \in \overline{S}$ and coordinates $1 \leq i < j < k \leq n$ such that*

- *Type I: $x\triangle e_i, x\triangle e_j, x\triangle e_k \in S$ and $x\triangle e_i\triangle e_j, x\triangle e_i\triangle e_k, x\triangle e_j\triangle e_k \in \overline{S}$, or*
- *Type II: $x\triangle e_i, x\triangle e_j, x\triangle e_k \in \overline{S}$ and $x\triangle e_i\triangle e_j, x\triangle e_i\triangle e_k, x\triangle e_j\triangle e_k \in S$.*

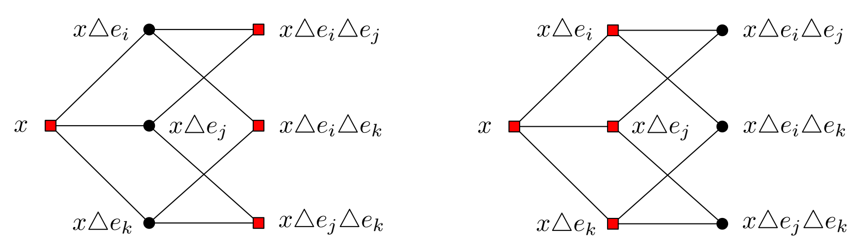


Figure 7. The forbidden configurations of Claim 2. Round points are in S and square points are in $\overline{S}$.

Proof of Claim. Depending on whether or not the point $x\triangle e_i\triangle e_j\triangle e_k$ is feasible, Type I gives an $R_{1,1}$ or a P_3 restriction, while Type II gives an S_3 or a P_3 restriction; as S has none of these restrictions, both configurations are forbidden. $\Diamond$

---

[7] Two graphs are *isomorphic* if one can be obtained from the other after relabeling the vertices.

Since S has degree k, there is an infeasible point with precisely k infeasible points adjacent to it. After a possible twisting and relabeling, if necessary, we may assume that $\mathbf{0}$ is infeasible, its neighbors $e_1,\ldots,e_{n-k}$ are feasible and its other neighbors $e_{n-k+1},\ldots,e_n$ are infeasible. We will partition the points of $\{0,1\}^n$ at distance 2 from $\mathbf{0}$ into three sets as follows:

$$X := \{e_i+e_j\ :\ 1\le i<j\le n-k\}$$
$$Y := \{e_i+e_j\ :\ 1\le i\le n-k<j\le n\}$$
$$Z := \{e_i+e_j\ :\ n-k+1\le i<j\le n\}.$$

Thus the set X consists of all the points e_i+e_j such that $e_i,e_j\in S$, Z of all the points e_i+e_j such that $e_i,e_j\in\overline{S}$, and Y of all the remaining points at distance 2 from $\mathbf{0}$. (If $k=1$ then $Z=\emptyset$.) We now use Claim 2 to deduce some bounds on the number of feasible and infeasible points in X and Z.

Claim 3. *The following statements hold:*

- $|X\cap\overline{S}|\le\left(\frac{n-k}{2}\right)^2$.
- $|Z\cap S|\le\left(\frac{k}{2}\right)^2$. *If* $|Z\cap S|\ge\left(\frac{k}{2}\right)^2-\frac{1}{4}$, *then there is a partition of* $\{e_{n-k+1},\ldots,e_n\}$ *into parts* I_1,I_2 *such that* $\big||I_1|-|I_2|\big|\le 1$ *and for distinct* $e_i,e_j\in I_1\cup I_2$,

$$e_i+e_j\in S\Leftrightarrow|\{e_i,e_j\}\cap I_1|=1.$$

Proof of Claim. Consider the simple graph G on vertices $\{e_1,\ldots,e_{n-k}\}$ and edges $\{e_ie_j\,:\,e_i+e_j\in\overline{S}\}\cong X\cap\overline{S}$. By Claim 2, S has no restriction of Type I, implying in turn that G is triangle-free. Thus, by Theorem 6.1, $|X\cap\overline{S}|\le\left(\frac{n-k}{2}\right)^2$. This proves the first part. For the next part, consider the simple graph G' on vertices $\{e_{n-k+1},\ldots,e_n\}$ and edges $\{e_ie_j\,:\,e_i+e_j\in S\}\cong Z\cap S$. By Claim 2, there is no restriction of Type II, implying in turn that G' is triangle-free. Thus, by Theorem 6.1, $|Z\cap S|\le\left(\frac{k}{2}\right)^2$ and if $|Z\cap S|\ge\left(\frac{k}{2}\right)^2-\frac{1}{4}$, then G' is a complete bipartite graph with bipartition I_1,I_2 such that $\big||I_1|-|I_2|\big|\le 1$, as required. $\Diamond$

Define $A:=\{(i,j):e_i\in S,e_i+e_j\in S\}$ and $B:=\left\{(i,j):e_i\in\overline{S},e_i+e_j\in S\right\}$.

Claim 4. *The following inequalities hold:*

$$2|X\cap S|+|Y\cap S|=|A|\le(n-k)k\le|B|=2|Z\cap S|+|Y\cap S|.$$

In particular, $|Z\cap S|\ge|X\cap S|$ *and if equality holds, then every point in* $\{e_1,\ldots,e_{n-k}\}$ *has precisely* k *feasible neighbors and every point in* $\{e_{n-k+1},\ldots,e_n\}$ *has precisely* k *infeasible neighbors.*

Proof of Claim. Notice that, for all distinct i,j with $e_i+e_j\in X\cap S$ the two pairs $(i,j),(j,i)$ belong to A, for all distinct i,j with $e_i+e_j\in Y\cap S$ exactly one of $(i,j),(j,i)$ belongs to A, while for all distinct i,j with $e_i+e_j\in Z\cap S$ neither of $(i,j),(j,i)$ belongs to A. Hence, $|A|=2|X\cap S|+|Y\cap S|$. Analogously, $|B|=2|Z\cap S|+|Y\cap S|$. On the one hand, each point in $S\cap\{e_i:i\in[n]\}=\{e_1,\ldots,e_{n-k}\}$ has at most k feasible neighbors by Claim 1, so

$$|A|\le|S\cap\{e_i:i\in[n]\}|\cdot k=(n-k)k.$$

On the other hand, each point in $\overline{S}\cap\{e_i : i\in[n]\} = \{e_{n-k+1},\dots,e_n\}$ has at least $n-k$ feasible neighbors by assumption, so

$$|B| \geq (n-k)\cdot|\overline{S}\cap\{e_i : i\in[n]\}| = (n-k)k,$$

(note that $\mathbf{0}\notin S$). All of these (in)equalities put together prove the claim. $\Diamond$

Hence, by Claims 3 and 4,

$$\frac{(n-k-1)^2-1}{4} = \binom{n-k}{2} - \left(\frac{n-k}{2}\right)^2 \leq |X| - |X\cap\overline{S}| = |X\cap S| \leq |Z\cap S| \leq \frac{k^2}{4}.$$

This proves (1). Since $k, n-k-1$ are both integers and $k\geq 1$, we must have that $n-k-1\leq k$, implying in turn that $n\leq 2k+1$, so (2) holds. To prove (3), assume that $n=2k+1$. Since S is not polar and is not one of $P_3, S_3, R_{1,1}$, it follows that $2k+1=n\geq 4$, so $k\geq 2$. Since $n=2k+1$, the inequalities above imply that $|Z\cap S|\geq\frac{k^2-1}{4}$ and $|Z\cap S| = |X\cap S|$, so Claims 3 and 4 prove (3), as required. $\square$

We are now ready to prove parts (1)-(3) of Theorem 1.20:

Proof of Theorem 1.20 (1)-(3). Take an integer $k\geq 2$ and a strictly non-polar set S of degree k, whose dimension is n. We first show that

(1) $n\in\{k,\dots,2k+1\}$.

Clearly, $n\geq k$. If $S\in\{P_3,S_3\}$, then $n=3\leq 7=2k+1$, so we are done. We may therefore assume that S has no P_3, S_3 restriction. Moreover, $S\neq R_{1,1}$ as $k>0$, so S has no $R_{1,1}$ restriction. As a result, we may apply Lemma 6.2. Choosing x to be any infeasible point with exactly k infeasible neighbors, Lemma 6.2 (2) implies that $n\leq 2k+1$, so (1) holds.

We next prove that

(2) if $n=k+1$, then either S is minimally non-polar, or after a possible relabeling,

$$S\subseteq\left\{x\in\{0,1\}^{k+1} : x_k = x_{k+1}\right\}$$

and the projection of S over coordinate $k+1$ is a critically non-polar set that is the reflective product of two other sets.

Assume that S is not minimally non-polar. By Proposition 3.3, and after a possible relabeling, we may assume that neither the 0-restriction nor the 1-restriction of S over coordinate $k+1$ has antipodal points. For $i\in\{0,1\}$, let $S_i\subseteq\{0,1\}^k$ be the i-restriction of S over coordinate $k+1$; as S is strictly non-polar, S_i is polar, so our hypothesis implies that the points in S_i all agree on a coordinate.

Claim 1. *The points in S_0 agree on the same coordinate as the points in S_1.*

Proof of Claim. Suppose otherwise. After a possible twisting and relabeling of S, we may assume that $S_0\subseteq\{x\in\{0,1\}^k : x_k=0\}$ and $S_1\subseteq\{x\in\{0,1\}^k : x_{k-1}=1\}$. Since every point in $\{x\in\{0,1\}^{k+1} : x_{k-1}=0, x_k=1\}$ is infeasible, and has k neighbors in $\{x\in\{0,1\}^{k+1} : x_k=1, x_{k+1}=0\}\cup\{x\in$

$\{0,1\}^{k+1} : x_{k-1} = 0, x_{k+1} = 1\}$, all of which are infeasible, it follows that all the other neighbors of the points in $\{x \in \{0,1\}^{k+1} : x_{k-1} = 0, x_k = 1\}$ are feasible, as S has degree k. Consequently,

$$\{x \in \{0,1\}^{k+1} : x_{k-1} = x_k = x_{k+1} = 0\} \cup \{x \in \{0,1\}^{k+1} : x_{k-1} = x_k = x_{k+1} = 1\} \subseteq S,$$

so S has antipodal points, a contradiction. $\Diamond$

After a possible relabeling, we may assume for $i \in \{0,1\}$ that the points in S_i agree on coordinate k. Since the points in S do not agree on the coordinate k, we may assume after a possible twisting that $S_0 \subseteq \{x \in \{0,1\}^k : x_k = 0\}$ and $S_1 \subseteq \{x \in \{0,1\}^k : x_k = 1\}$. In particular,

$$S \subseteq \left\{x \in \{0,1\}^{k+1} : x_k = x_{k+1}\right\}.$$

Thus by Remark 3.2, the projection of S over coordinate $k+1$ is strictly non-polar. For $i \in \{0,1\}$, let $R_i \subseteq \{0,1\}^{k-1}$ be the i-restriction of S_i over coordinate k. Notice that $(R_0 \times \{0\}) \cup (R_1 \times \{1\})$ is the projection of S over coordinate $k+1$.

Claim 2. $R_0 = \overline{R_1}$.

Proof of Claim. Let us first prove that $R_0 \cap R_1 = \emptyset$. Suppose otherwise. Pick $x^\star \in R_0 \cap R_1$. Then $(x^\star, 0, 0), (x^\star, 1, 1) \in S$, so as S is non-polar, $(\mathbf{1} - x^\star, 0, 0), (\mathbf{1} - x^\star, 1, 1) \in \overline{S}$. Since $\{x \in \{0,1\}^{k+1} : x_k = 1, x_{k+1} = 0\} \subseteq \overline{S}$, it follows that the infeasible point $(\mathbf{1} - x^\star, 1, 0)$ has $k+1$ infeasible neighbors, a contradiction as S has degree k. Thus, $R_0 \cap R_1 = \emptyset$. It remains to prove that $R_0 \cup R_1 = \{0,1\}^{k-1}$. Suppose otherwise. Pick $y^\star \in \{0,1\}^{k-1} - (R_0 \cup R_1)$. Then $(y^\star, 0, 0), (y^\star, 1, 1) \in \overline{S}$, so similarly as above, the infeasible point $(y^\star, 1, 0)$ has $k+1$ infeasible neighbors, a contradiction as S has degree k. Thus, $R_0 \cup R_1 = \{0,1\}^{k-1}$, as required. $\Diamond$

As a result, $(R_0 \times \{0\}) \cup (R_1 \times \{1\}) = (R_0 \times \{0\}) \cup (\overline{R_0} \times \{1\}) = R_0 * \{0\}$. Since $R_0 * \{0\}$ is strictly non-polar, it follows from Theorem 1.18 (3) that the projection of S over coordinate $k+1$ is a critically non-polar set that is the reflective product of two other sets, thereby finishing the proof of (2).

Lastly, we prove that

(3) if $n \geq k+2$, then S is critically non-polar.

Let $S_0 \subseteq \{0,1\}^{n-1}$ be the 0-restriction of S over coordinate n. As S is strictly non-polar, S_0 is polar. By Proposition 3.3, and after a possible twisting and relabeling, it suffices to show that S_0 has antipodal points. Suppose otherwise. Then the points in S_0 all agree on a coordinate. After a possible twisting and relabeling, we may assume that $S_0 \subseteq \{x \in \{0,1\}^{n-1} : x_{n-1} = 1\}$. Thus, the points in the set $\{x \in \{0,1\}^n : x_{n-1} = x_n = 0\}$ are all infeasible, and as each point of the set has $n-2$ neighbors in the set, it follows that $k \geq n-2$ because S has degree k. Hence, since $n \geq k+2$, it follows that $n = k+2$ and the neighbors of each point in $\{x \in \{0,1\}^n : x_{n-1} = x_n = 0\}$ outside the set must be all feasible. So

$$\{x \in \{0,1\}^n : x_{n-1} = 0, x_n = 1\} \cup \{x \in \{0,1\}^n : x_{n-1} = 1, x_n = 0\} \subseteq S,$$

implying in turn that S has antipodal points, a contradiction. This proves (3). $\square$

For the next and last part of Theorem 1.20, we need a second tool for finding delta minors:

Theorem 6.3 ([4], Theorem 2.1)**.** *Let $\mathcal{C}$ be a clutter. If there is an element f and distinct members C_1, C_2, C such that $f \in C_1 \cap C_2$, $f \notin C$ and $C_1 \cup C_2 \subseteq C \cup \{f\}$, then $\mathcal{C}$ has a delta minor through f.*

We are now ready to prove Theorem 1.20 (4):

Proof of Theorem 1.20 (4). Take an integer $k \geq 2$ and a strictly non-polar set $S \subseteq \{0,1\}^{2k+1}$ that has degree k. We need to show that $|S| = 2^{2k}$, every infeasible point has exactly k infeasible neighbors, and $\mathrm{cuboid}(S)$ is an ideal minimally non-packing clutter. As $2k+1 \geq 5$ and S is strictly non-polar, S has no $P_3, S_3, R_{1,1}$ restriction. We may therefore apply Lemma 6.2.

Claim 1. *For each $x \in \{0,1\}^{2k+1}$, we have that $|S \cap \{x, \mathbf{1} - x\}| = 1$. In particular, $|S| = 2^{2k}$.*

Proof of Claim. There are no antipodal feasible points, so $|S \cap \{x, \mathbf{1} - x\}| \leq 1$. Suppose for contradiction that both $x, \mathbf{1} - x$ are infeasible. The infeasible point x has at most k infeasible neighbors, so it has at least $k+1$ feasible neighbors. Similarly, the infeasible point $\mathbf{1} - x$ has at most k infeasible neighbors, so it has at least $k+1$ feasible neighbors. By the Pigeonhole Principle, there are antipodal feasible points, one in the neighborhood of x and the other in the neighborhood of $\mathbf{1} - x$, a contradiction. $\Diamond$

For a point in $\overline{S}$ define its *degree* to be the number of infeasible points adjacent to it, and for a point in S define its *degree* to be the number of feasible points adjacent to it.

Claim 2. *If a point in $\{0,1\}^{2k+1}$ has degree k, then so do all the points of $\{0,1\}^{2k+1}$ adjacent to it.*

Proof of Claim. Lemma 6.2 (3) proves the claim for infeasible points. To conclude that the same holds for all feasible points, notice that $\overline{S} = S \triangle \mathbf{1}$ by Claim 1. Thus, if a feasible point x has degree k, the infeasible point $\mathbf{1} - x$ also has degree k and so do all the points adjacent to it, implying in turn that all the points adjacent to x have degree k as well. This finishes the proof of the claim. $\Diamond$

Since there is at least one point whose degree is k, Claim 2 implies that every point of $\{0,1\}^n$ has degree k. Thus, every infeasible point has exactly k infeasible neighbors. Next we show that $\mathrm{cuboid}(S)$ is an ideal minimally non-packing clutter. By Theorem 1.3, it suffices to show that $\mathrm{cuboid}(S)$ is minimally non-packing. By Theorem 1.20 (3), S is critically non-polar. Thus, by Theorem 3.6, it suffices to show that the induced clutters of S have the packing property. We need the following:

Claim 3. *The induced clutters of proper restrictions of S do not have a delta minor.*

Proof of Claim. Let S' be a proper restriction of S. As S is strictly non-polar, S' is strictly polar. Thus by Proposition 3.1, every minor of $\mathrm{cuboid}(S')$ has a cover of cardinality one, or two disjoint members. In particular, $\mathrm{cuboid}(S')$ does not have a delta minor, implying in turn that the induced clutters of S' do not have a delta minor, as required. $\Diamond$

Take a point $x \in \overline{S}$. By symmetry, it suffices to show that $\mathrm{ind}(S \triangle x)$ has the packing property. After a possible twisting, we may assume that $x = \mathbf{0}$. The infeasible point $\mathbf{0}$ has exactly k infeasible neighbors; after a possible relabeling, we may assume that $\{e_1, \ldots, e_{k+1}\} \subseteq S$ and $\{e_{k+2}, \ldots, e_{2k+1}\} \subseteq \overline{S}$. By Lemma 6.2 (3), there is a partition $I_1 \cup I_2$ of $\{e_{k+2}, \ldots, e_{2k+1}\}$ such that $\big||I_1| - |I_2|\big| \leq 1$ and for all distinct $e_i, e_j \in \{e_{k+2}, \ldots, e_{2k+1}\}$,

$$e_i + e_j \in S \Leftrightarrow \big|I_1 \cap \{e_i, e_j\}\big| = 1.$$

Notice that since $k \geq 2$, $|I_1| + |I_2| \geq 2$.

Claim 4. *Let $S' \subseteq \{0,1\}^{I_1 \cup I_2}$ be obtained from S after 0-restricting coordinates $[k+1]$. Then $\mathrm{ind}(S') = \big\{\{i,j\} : e_i \in I_1, e_j \in I_2\big\}$.*

Proof of Claim. We know that $\big\{\{i,j\} : e_i \in I_1, e_j \in I_2\big\}$ are the only members of $\mathrm{ind}(S')$ of cardinality at most two. Suppose for a contradiction that $\mathrm{ind}(S')$ has another member $C \subseteq I_1 \cup I_2$, so $|C| \geq 3$. After possibly relabeling I_1 and I_2, we may assume that $|C \cap I_2| \geq 2$. Pick distinct coordinates $j, j' \in C \cap I_2$ and pick an arbitrary $i \in I_1$. Notice that $\{i,j\}, \{i,j'\}, C$ are members of $\mathrm{ind}(S')$, implying in particular that $i \notin C$, and so by Theorem 6.3, $\mathrm{ind}(S')$ has a delta minor, thereby contradicting Claim 3. $\Diamond$

As a result,

$$\mathrm{ind}(S) = \big\{\{1\}, \{2\}, \ldots, \{k+1\}\big\} \cup \big\{\{i,j\} : e_i \in I_1, e_j \in I_2\big\},$$

so $\mathrm{ind}(S)$ has the packing property by Corollary 5.3, as required. $\square$

Let us end this subsection with the following question:

Question 6.4. *Is R_5 the only strictly non-polar set of degree k and dimension $2k+1$, for some $k \geq 2$?*

6.2. Generating strictly non-polar sets of degree at most 4.

Using a computer code we have generated all the strictly non-polar sets of degree at most 3, and all the strictly non-polar sets of degree 4 and dimension at most 7. Before describing the code, let us prove that every strictly non-polar set of degree at most 4 that is also critically non-polar has an ideal minimally non-packing cuboid. We need the following result:

Theorem 6.5 ([4], Theorem 1.10 (iii))**.** *Take integers $n, k \geq 1$ and a set $S \subseteq \{0,1\}^n$ of degree at most k. Then every minimally non-ideal minor of $\mathrm{cuboid}(S)$, if any, has at most k elements.*

We leave the following as an exercise for the reader:

Remark 6.6. *Δ_3, Δ_4 are the only minimally non-ideal clutters over at most 4 elements.*

We are now ready to prove the following:

Corollary 6.7. *Take an integer $n \geq 3$ and a critically non-polar set $S \subseteq \{0,1\}^n$ of degree at most 4. Then $\mathrm{cuboid}(S)$ is an ideal minimally non-packing clutter.*

Proof. As Δ_3, Δ_4 are minimally non-packing clutters with covering number two, $\mathrm{cuboid}(S)$ does not have them as a proper minor by Proposition 3.7. Thus, since Δ_3, Δ_4 are not cuboids, $\mathrm{cuboid}(S)$ does not have them as a minor at all. By Theorem 6.5 and Remark 6.6, we therefore get that $\mathrm{cuboid}(S)$ is ideal, so S is cube-ideal by Theorem 1.6. It now follows from Corollary 4.14 (2) that $\mathrm{cuboid}(S)$ is minimally non-packing as well, as required. □

Take an integer $n \geq 1$. A *partial set* is a triple $P = (F, I, U)$ where

- $F \subseteq \{0,1\}^n$ is a set of *feasible points*,
- $I \subseteq \{0,1\}^n \times \mathbb{Z}$ is a set of *infeasible pairs*, where each infeasible pair is of the form (p, d) where p is an *infeasible point* and d is the number of neighbors of p that correspond to an infeasible point, and
- $U \subseteq \{0,1\}^n$ is a set of *undecided points*,

where every point of $\{0,1\}^n$ is either feasible, infeasible or undecided (and not more than one of the three). If $U = \emptyset$, then F is the *corresponding set* of P. Take an integer $k \in \{0, 1, \ldots, n\}$ and a set $S \subseteq \{0,1\}^k$. The *n-dimensional partial set originating from S* is the partial set whose feasible and infeasible points are $S \times \{\mathbf{0}^{n-k}\}$ and $\overline{S} \times \{\mathbf{0}^{n-k}\}$, respectively. We are now ready to describe a pseudocode for finding the strictly non-polar sets of bounded degree.

Input: degree $k \in \{0, 1, 2, \ldots\}$
Output: all non-isomorphic strictly non-polar sets of degree at most k
Algorithm

(1) Enumerate all non-isomorphic subsets of $\{0,1\}^k$ all of whose proper restrictions are polar.

Observe that each set in (1) is either strictly polar or strictly non-polar. For each $n \in \{k, k+1, \ldots, 2k+1\}$, let $\mathcal{P}_n$ be the family of all n-dimensional partial sets originating from a set in (1). Set $n := k$.

(2) While $n \leq 2k+1$:
  - (a) While $\mathcal{P}_n$ has a partial set P with an undecided point:
    - (i) If P has antipodal feasible points, then set $\mathcal{P}_n := \mathcal{P}_n - \{P\}$.
    - (ii) If P has an infeasible point with more than k infeasible neighbors, then set $\mathcal{P}_n := \mathcal{P}_n - \{P\}$.
    - (iii) If P has an undecided point whose antipodal is feasible, update P by making the undecided point infeasible.
    - (iv) If P has an infeasible point with k infeasible neighbors, update P by making the undecided neighbors feasible.
    - (v) Otherwise, take an undecided point q. Let P_1 and P_2 be the partial sets obtained from P after making q feasible and infeasible, respectively. Set $\mathcal{P}_n := \mathcal{P}_n \triangle \{P, P_1, P_2\}$.
  - (b) Set $n := n + 1$.

At this point, the partial sets in $\bigcup_{n=k}^{2k+1} \mathcal{P}_n$ have no undecided point. Let $\mathcal{S}$ be the family of sets corresponding to the partial sets in $\bigcup_{n=k}^{2k+1} \mathcal{P}_n$.

(3) Keep only the sets in $\mathcal{S}$ that are strictly non-polar and of degree at most k.

(4) From every isomorphic class in $\mathcal{S}$, output only one set and ignore the other ones.

End of Algorithm

The justification for the inequality $n \le 2k+1$ in step (2) comes from Theorem 1.20 (1). Our implementation also relies heavily on Lemma 3.8, implying that the following statements are equivalent for a set S:

- every proper restriction of S is polar,
- for all $a, b, c \in S$ that agree on at least one coordinate, the smallest restriction of S containing a, b, c is polar.

This characterization is used for implementing steps (1) and (3) of the code.

Take integers $n, k \ge 1$ and a set $S \subseteq \{0,1\}^n$. We say that S is *half-dense* if $|S| = 2^{n-1}$, and that S is *k-regular* if every infeasible point has exactly k infeasible neighbors. Notice that by Theorem 1.20 (4), every strictly non-polar set of dimension $2k+1$ and degree k is half-dense and k-regular. Therefore, if one is interested in generating strictly non-polar sets of dimension n that are half-dense and k-regular, step (2) of the code can be modified accordingly to speed up the process.

After running the code for $k \in \{0,1,2,3,4\}$, we get the following results:

Theorem 6.8. *The following statements hold, up to isomorphism:*

- $R_{1,1}, R_{2,1}, R_5$ *are the strictly non-polar set of degree at most* 2.
- P_3, S_3 *are the strictly non-polar sets of degree* 3 *and dimension* 3*, both of them are minimally non-polar, and none of them are critically non-polar.*
- *There are* 4 *strictly non-polar sets of degree* 3 *and dimension* 4*, three of them are minimally non-polar, and none of them are critically non-polar.*
- *There are* 3 *strictly non-polar sets of degree* 3 *and dimension* 5*, all of which are critically non-polar by Theorem 1.20 (3).*
- *There are* 2 *strictly non-polar sets of degree* 3 *and dimension* 6*, all of which are critically non-polar by Theorem 1.20 (3). Moreover, each set is half-dense and* 3*-regular.*
- *There is no strictly non-polar set of degree* 3 *and dimension* 7*.*
- *There are* 11 *strictly non-polar sets of degree* 4 *and dimension* 4*,* 6 *of them are minimally non-polar, and none of them are critically non-polar.*
- *There are* 37 *strictly non-polar sets of degree* 4 *and dimension* 5*,* 36 *of them are minimally non-polar, and* 25 *of them are critically non-polar.*
- *There are* 682 *strictly non-polar sets of degree* 4 *and dimension* 6*, all of which are critically non-polar by Theorem 1.20 (3).*
- *There is only* 1 *strictly non-polar sets of degree* 4 *and dimension* 7*, which is critically non-polar by Theorem 1.20 (3). Moreover, this set is half-dense and* 4*-regular.*
- *There is no half-dense* 4*-regular strictly non-polar set of degree* 4 *and dimension* 8*.*

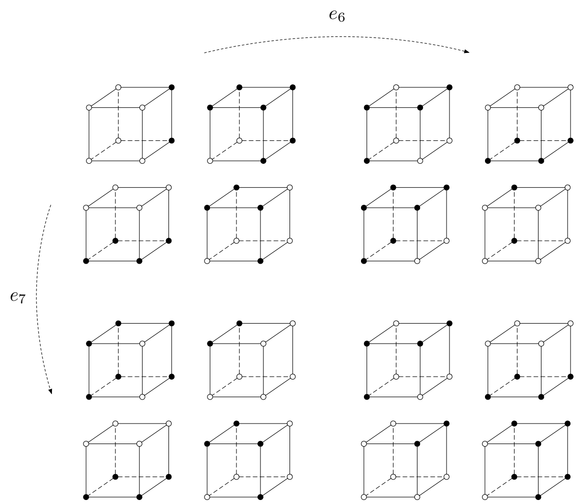


Figure 8. An illustration of the strictly non-polar set of degree 4 and dimension 7.

As a consequence, up to isomorphism, there are exactly 745 *strictly non-polar sets of degree at most* 4 *and dimension at most* 7*,* 738 *of which are minimally non-polar,* 716 *of which are critically non-polar and have ideal minimally non-packing cuboids by Corollary 6.7.*

The computer-assisted proof of this theorem can be found on GitHub [1], where the code is also available. The appendix has an explicit description of the 745 strictly non-polar sets from above. See Figure 3 for a summary, Figure 9 for an illustration of the strictly non-polar sets of degree 3, and Figure 8 for an illustration of the strictly non-polar set of degree 4 and dimension 7.

Out of all the critically non-polar sets of degree at most 4 and dimension at most 7, 71 of them are half-dense and most of the other ones are *nearly* half-dense. For instance, every critically non-polar set of degree 4 and dimension 5 has size at least 11, and among the critically non-polar sets of degree 4 and dimension 6, 10 have size 27, 73 have size 28, 168 have size 29, 234 have size 30, 136 have size 31, and the remaining 61 have size 32.

Question 6.9. *Take an integer $k \geq 2$. Let S be a strictly non-polar set of degree k and of maximum possible dimension $n(k)$. What is $\lim_{k\to\infty} \frac{n(k)}{k}$? Is S necessarily (nearly) half-dense? Is S necessarily k-regular? Is S necessarily cube-ideal? Is $\mathrm{cuboid}(S)$ necessarily minimally non-packing?*

7. Concluding remarks and open questions

Cuboids, a natural home to ideal minimally non-packing clutters with covering number two, were comprehesively studied in this paper. Ideal minimally non-packing cuboids of bounded degree were studied, and more than seven hundred non-isomorphic ones over at most 14 elements were generated. Cuboids were also used as a tool to manifest the geometry behind idealness and the packing property. We saw that idealness is a local property while the packing property is not, resulting in a geometric rift between these two properties. We

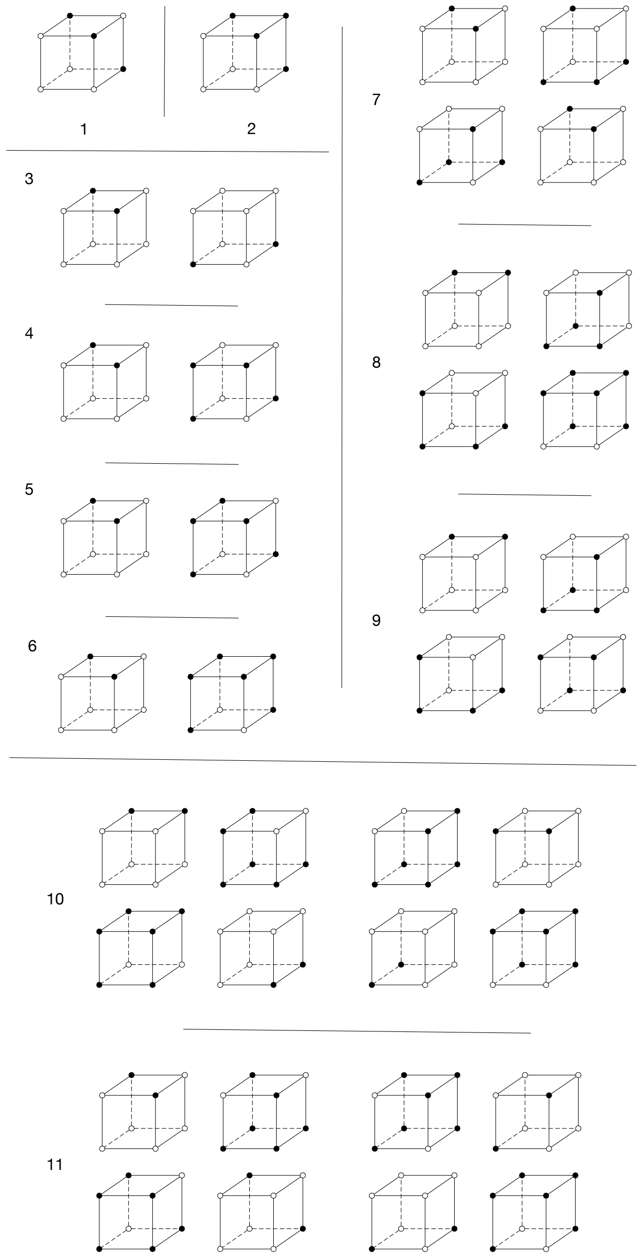


Figure 9. An illustration of the strictly non-polar sets of degree 3.

showed that strict polarity, a tractable property, makes the packing property local. Even though cuboids form a special class of clutters, we saw that some of the main conjectures and theorems about clutters – such as the $\tau = 2$ Conjecture, the Replication Conjecture, the f-Flowing Conjecture, and the classification of the binary matroids with the sums of circuits property – can be formulated equivalently in terms of cuboids. We studied three basic binary operations on cuboids, namely the Cartesian product, the coproduct and the reflective product, and their interplay with idealness and the packing property. This interplay revealed the starring role of the sets

$\{R_{k,1} : k \geq 1\} \cup \{R_5\}$, whose cuboids are ideal minimally non-packing, and it also brought out the importance of strict connectivity and antipodal symmetry when studying such clutters.

Let us wrap up with a few remarks and open questions. Not only is idealness a local property, but

Proposition 7.1. *The minor-closed properties "the blocker has the packing property" and "the blocker has the max-flow min-cut property" are local.*

We leave this as an exercise for the reader. Once strict polarity is enforced, the packing property becomes local too. Well, the Replication Conjecture predicts that the packing property is equivalent to the max-flow min-cut property, so

Conjecture 7.2. *Take an integer $n \geq 1$ and a strictly polar set $S \subseteq \{0,1\}^n$. Then $\mathrm{cuboid}(S)$ has the max-flow min-cut property if, and only if, all of its induced clutters have the max-flow min-cut property.*

We should point out that if the $\tau = 2$ Conjecture is true, then so is the Replication Conjecture ([10], Proposition 2).

Theorem 1.18 (3) and Conjecture 1.15, if true, would imply that if $S_1 * S_2$ is cube-ideal and strictly non-polar, then its cuboid must be minimally non-packing. By Theorems 1.16, 1.18 parts (1), (2), (4) and 1.19, this problem is equivalent to the following:

Conjecture 7.3. *Take an integer $n \geq 1$ and a set $S \subseteq \{0,1\}^n$ such that $S, \overline{S}$ are nonempty, strictly connected, antipodally symmetric, cube-ideal and strictly polar. Then $\mathrm{cuboid}(S), \mathrm{cuboid}(\overline{S})$ have the packing property.*

Perhaps a more pressing question is the following:

Question 7.4. *Are $\{R_{k,1} : k \geq 1\} \cup \{R_5\}$ the only sets with an ideal minimally non-packing cuboid that can be written as the reflective product of two other sets?*

This question is answered affirmatively for a class of ideal minimally non-packing clutters that are cuboids of so-called 1-*resistant* sets ([3], follows from Theorem 1.17).

ACKNOWLEDGEMENTS

We would like to thank Kanstantsin Pashkovich for helpful discussions about parts of this paper. We would also like to thank two anonymous referees for their diligent review of our work; it has led to us to publish our code on GitHub [1]. This work was supported in parts by ONR grant 00014-18-12129, NSF grant CMMI-1560828, and an NSERC PDF grant 516584-2018. Our code is written in DrRacket, Version 7.3 [15]. Finally, many thanks to Christopher Calzonetti as well as the Math Faculty Computing Facility at the University of Waterloo for their computing support.

REFERENCES

[1] Abdi, A.: A code for generating all non-isomorphic strictly non-polar sets of fixed dimension and bounded degree. (2019) GitHub repository, https://github.com/ahmad-abdi/cuboids-code/

[2] Abdi, A.: Ideal clutters. Ph.D. Dissertation, University of Waterloo (2018)

[3] Abdi, A., Cornuéjols, G., Lee, D.: Resistant sets in the unit hypercube. To appear in Math. Oper. Res.

[4] Abdi, A., Cornuéjols, G., Pashkovich, K.: Ideal clutters that do not pack. Math. Oper. Res. **43**(2), 533–553 (2018)

[5] Barahona, F. and Grötschel, M.: On the cycle polytope of a binary matroid. J. Combin. Theory Ser. B **40**(1), 40–62 (1986)

[6] Bouchet, A.: Greedy algorithm and symmetric matroids. Math. Prog. **38**(2), 147–159. (1987)

[7] Conforti, M. and Cornuéjols, G.: Clutters that pack and the max-flow min-cut property: a conjecture. The Fourth Bellairs Workshop on Combinatorial Optimization (1993)

[8] Conforti, M., Cornuéjols, G., Zambelli, G.: Integer programming. Springer (2014)

[9] Cornuéjols, G.: Combinatorial optimization, packing and covering. SIAM, Philadelphia (2001)

[10] Cornuéjols, G., Guenin, B., Margot, F.: The packing property. Math. Program. Ser. A **89**(1), 113–126 (2000)

[11] Cornuéjols, G. and Novick, B.: Ideal 0,1 matrices. J. Combin. Theory Ser. B **60**, 145–157 (1994)

[12] Edmonds, J. and Fulkerson, D.R.: Bottleneck extrema. J. Combin. Theory Ser. B **8**, 299–306 (1970)

[13] Edmonds, J. and Giles, R.: A min-max relation for submodular functions on graphs. Annals of Discrete Math. **1**, 185–204 (1977)

[14] Edmonds, J. and Johnson, E.L.: Matchings, Euler tours and the Chinese postman problem. Math. Prog. **5**, 88–124. (1973)

[15] Flatt, M. and PLT. Reference: Racket. Technical Report PLT-TR-2010-1, PLT Design Inc. (2010) https://racket-lang.org/tr1/

[16] Flores, A., Gitler, I., Reyes, E.: On 2-partitionable clutters and the MFMC property. Adv. Appl. Discrete Math. **2**(1), 59–84. (2008)

[17] Fulkerson, D.R.: Blocking and anti-blocking pairs of polyhedra. Math. Program. **1**, 168–194 (1971)

[18] Guenin, B.: A short proof of Seymour's characterization of the matroids with the max-flow min-cut property. J. Combin. Theory Ser. B **86**(2), 273–279 (2002)

[19] Guenin, B.: Perfect and ideal $0,\pm 1$ matrices. Math. Oper. Res. **23**(2), 322–338 (1998)

[20] Hoffman, A.J.: A generalization of max flow-min cut. Math. Prog. **6**(1), 352–359 (1974)

[21] Isbell, J.R.: A class of simple games. Duke Math. J. **25**(3), 423–439 (1958)

[22] Lee, J.: Cropped cubes. J. Combin. Optimization **7**(2), 169–178. (2003)

[23] Lehman, A.: A solution of the Shannon switching game. Society for Industrial Appl. Math. **12**(4), 687–725 (1964)

[24] Lehman, A.: On the width-length inequality. Math. Program. **17**(1), 403–417 (1979)

[25] Lehman, A.: The width-length inequality and degenerate projective planes. DIMACS Vol. **1**, 101–105 (1990)

[26] Lovász, L.: Minimax theorems for hypergraphs. Lecture Notes in Mathematics **411**, Springer-Verlag 111–126 (1972)

[27] Mantel, W.: Vraagstuk XXVIII. Wiskundige Opgaven **10**, 60–61 (1910)

[28] Nobili, P. and Sassano, A.: $(0,\pm 1)$ ideal matrices. Math. Program. **80**(3), 265–281 (1998)

[29] Novick, B. and Sebő, A.: On combinatorial properties of binary spaces. Integer Programming and Combinatorial Optimization, Vol. **4**, 212–227 (1995)

[30] Oxley, J.: Matroid theory, second edition. Oxford University Press, New York (2011)

[31] Schrijver, A.: A counterexample to a conjecture of Edmonds and Giles. Discrete Math. **32**, 213–214 (1980)

[32] Seymour, P.D.: Decomposition of regular matroids. J. Combin. Theory Ser. B **28**, 305–359 (1980)

[33] Seymour, P.D.: Matroids and multicommodity flows. Europ. J. Combinatorics **2**, 257–290 (1981)

[34] Seymour, P.D.: On Lehman's width-length characterization. DIMACS Vol. **1**, 107–117 (1990)

[35] Seymour, P.D.: Sums of circuits. Graph Theory and Related Topics (Bondy, J.A. and Murty, U.S.R., eds), Academic Press, New York 342–355. (1979)

[36] Seymour, P.D.: The matroids with the max-flow min-cut property. J. Combin. Theory Ser. B **23**, 189–222 (1977)

[37] Szekeres, G.: Polyhedral decomposition of cubic graphs. Bull. Austral. Math. Soc. **8**, 367–387. (1973)

THE STRICTLY NON-POLAR SETS OF DEGREE AT MOST 4 AND DIMENSION AT MOST 7

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
	(0,3)
1	(“100" "010" "001" "111")
	(2,4)
2	(“0001" "1000" "0100" "0010" "1100" "0110" "1010" "1111")
	(2,5)
3	(“10110" "00100" "00001" "01010" "01100" "11000" "11001" "00011" "10100" "11111" "10111" "11010" "10001" "01111" "00010" "01101")
	(3,3)
4	(“100" "010" "001")
5	(“100" "010" "001" "000")
	(3,4)
6	(“0010" "1110" "1001" "0101")
7	(“1111" "1101" "0001" "1011" "0110" "1010")
8	(“1111" "0011" "1101" "0001" "1011" "0110" "1010")
9	(“1111" "0011" "1101" "0001" "1011" "0111" "0110" "1010")
	(3,5)
10	(“00001" "01100" "10010" "11001" "10100" "10111" "11010" "01001" "01111" "00010" "10101" "01110")
11	(“10110" "00001" "00111" "01010" "01100" "00101" "11011" "10010" "11001" "10111" "10001" "00010" "11100" "01011")
12	(“10110" "00001" "00111" "01010" "01100" "00101" "11011" "10010" "11001" "11111" "10111" "10001" "01111" "00010" "11100" "01011")
	(3,6)
13	("110110" "010001" "011111" "010011" "101101" "011010" "101010" "010111" "111000" "100010" "100100" "010100" "100001" "001111" "011100" "100110" "101001" "111111" "001100" "110001" "000011" "011000" "000001" "001101" "111001" "001010" "110111" "000010" "111010" "000100" "101111" "110100")
14	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "011001" "100010" "100100" "010100" "110011" "001111" "011100" "101001" "011110" "111111" "110001" "000011" "000101" "101000" "011000" "000001" "111001" "000111" "001010" "110111" "000010" "000100" "101111" "110100" "110010")
	(4,4)
	not minimally non-polar
15	(“0011" "1101" "1110")
16	(“1111" "0011" "1101" "1110")
17	(“0011" "1011" "1101" "0110")
18	(“1111" "0011" "1011" "1101" "0110")
19	(“1111" "0011" "1011" "1101" "1110" "1001")
	minimally non-polar
20	(“0011" "1011" "1101" "1010" "0110")
21	(“0011" "1011" "1101" "1110" "1001" "0101")
22	(“1111" "0011" "1011" "1101" "1010" "0110")
23	(“1111" "0011" "1011" "1101" "1110" "1001" "0101")
24	(“0011" "1011" "1101" "1110" "1001" "0111" "0101")
25	(“1111" "0011" "1011" "1101" "1110" "1001" "0111" "0101")
	(4,5)
	not minimally non-polar
26	(“10110" "11101" "01010" "10010" "00001" "11010" "00110" "01110")
	minimally non-polar, not critically non-polar
27	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "11100" "11111")
28	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "00110" "11100" "11111")
29	("10110" "11101" "10010" "00001" "11011" "11000" "11010" "00110" "10001" "11100" "11111")
30	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "00110" "10111" "11100" "11111")
31	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "00110" "10001" "11100" "11111")
32	("10110" "11101" "10010" "00001" "11011" "11000" "11010" "10101" "00110" "10001" "11100" "11111")
33	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "11100" "11111")
34	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "00110" "10111" "10001" "11100" "11111")
35	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "10001" "11100" "11111")
36	("10110" "11101" "01010" "10010" "00001" "10011" "00111" "11011" "11010" "00110" "10111" "00011" "01110" "01011" "01111")
37	("10110" "11101" "01010" "10010" "00001" "10011" "00111" "11011" "11010" "00110" "10111" "00011" "01110" "01011" "01111" "11111")
	critically non-polar
38	("10110" "11101" "10010" "00001" "11000" "11010" "00110" "01110" "01111" "11100" "11111")
39	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "01110" "11100")
40	("10110" "11101" "10010" "00001" "10011" "11000" "11010" "00110" "01110" "01111" "11100" "11111")
41	("10110" "11101" "10010" "00001" "11000" "11010" "10101" "00110" "01110" "01011" "01111" "11100")
42	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "01110" "11100" "11111")
43	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "10111" "01110" "11100")
44	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10001" "00011")
45	("10110" "11101" "00001" "11000" "01100" "11010" "10101" "00110" "01101" "10001" "00011" "01011")
46	("10110" "11101" "10010" "00001" "11000" "11010" "10101" "00110" "01110" "01011" "01111" "11100" "11111")
47	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "10111" "01110" "11100" "11111")
48	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "10111" "01110" "01111" "11100")
49	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10001" "00011" "01011")
50	("10110" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10111" "10001" "00011" "01011")
51	("10110" "11101" "10010" "00001" "10011" "11000" "11010" "10101" "00110" "10111" "01110" "01111" "11100" "11111")
52	("10110" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "01110" "01111" "11100" "11111")
53	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "01110" "01011" "11100")
54	("10110" "11101" "00001" "10011" "11011" "11000" "11010" "00110" "01101" "10111" "01110" "01111" "11100" "11111")
55	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10111" "10001" "00011" "11111")
56	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10111" "10001" "00011" "01011")
57	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "01110" "01011" "11100" "11111")
58	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "01110" "01011" "01111" "11100")
59	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10001" "00011" "01011" "01111" "11111")
60	("10110" "11101" "10010" "00001" "10011" "11011" "11000" "11010" "10101" "00110" "10111" "01110" "01011" "01111" "11100" "11111")
61	("10110" "11101" "00001" "11011" "11000" "01100" "11010" "10101" "00110" "01101" "10111" "10001" "00011" "01011" "01111" "11111")
62	("00101" "11101" "00001" "10011" "11011" "11000" "10101" "00110" "01101" "10111" "10001" "00011" "01011" "01001" "01111" "11111")
	(4,6)
	size 27
63	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "111100")
64	("101100" "001110" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
65	("101100" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
66	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
67	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "111100")
68	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
69	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
70	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
71	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "110100" "000110" "110010" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
72	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "011001" "100111" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "110100" "000110" "110010" "111100")
	size 28
73	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
74	("101100" "001110" "010001" "100011" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
75	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
76	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
77	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "111100")
78	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
79	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
80	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
81	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
82	("101100" "110110" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
83	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
84	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
85	("101100" "110110" "001110" "010001" "100011" "011111" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
86	("101100" "001110" "010001" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
87	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
88	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
89	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
90	("101100" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
91	("101100" "001110" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
92	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
93	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "010110" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
94	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
95	("101100" "001110" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
96	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
97	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
98	("101100" "001110" "010001" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
99	("101100" "001110" "010001" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
100	("101100" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
101	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
102	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "110100" "110010" "111100")
103	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
104	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
105	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
106	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
107	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
108	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
109	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
110	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
111	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "111100")
112	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
113	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
114	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
115	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
116	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
117	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
118	("101100" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
119	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
120	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
121	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
122	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
123	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "111100")
124	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
125	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
126	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
127	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
128	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
129	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
130	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
131	("101100" "010001" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "010010" "011100" "110000" "100110" "101001" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
132	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
133	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "110111" "000010" "101111" "110100" "000110" "111100")
134	("101100" "010001" "011111" "100101" "101010" "011001" "100111" "101011" "100010" "100100" "010010" "011100" "101001" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "000111" "000010" "111010" "110100" "000110" "110010" "111100")
135	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
136	("101100" "001110" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "111111" "001100" "000101" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
137	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
138	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "110000" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "110100" "000110" "110010" "111100")
139	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "011001" "100111" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "110100" "000110" "110010" "111100")
140	("101100" "110110" "001110" "010101" "011111" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
141	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011011" "011010" "011001" "100111" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "110100" "000110" "110010" "111100")
142	("101100" "001110" "010001" "010101" "100011" "011011" "011010" "010111" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
143	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
144	("101100" "010001" "010101" "011111" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
145	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "110100" "001100" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
	size 29
146	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
147	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
148	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
149	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
150	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
151	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
152	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
153	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
154	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
155	("101100" "001110" "010001" "100011" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
156	("101100" "001110" "010001" "100011" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
157	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
158	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
159	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
160	("101100" "110110" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
161	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
162	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
163	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
164	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
165	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
166	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
167	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
168	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
169	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
170	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
171	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
172	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
173	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
174	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
175	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
176	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
177	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
178	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
179	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
180	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
181	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
182	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
183	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
184	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
185	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
186	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
187	("101100" "001110" "010001" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
188	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
189	("101100" "110110" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
190	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
191	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
192	("101100" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
193	("101100" "001110" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
194	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
195	("101100" "001110" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
196	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "010110" "010010" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
197	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "010110" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
198	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
199	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "111100")
200	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
201	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
202	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
203	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
204	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
205	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
206	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
207	("101100" "001110" "010001" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
208	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
209	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
210	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
211	("101100" "001110" "010001" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
212	("101100" "001110" "010001" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
213	("101100" "001110" "010001" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
214	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
215	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
216	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
217	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
218	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
219	("101100" "001110" "010101" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
220	("101100" "001110" "010101" "100011" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
221	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "110100" "110010" "111100")
222	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
223	("101100" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
224	("101100" "110110" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
225	("101100" "110110" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
226	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
227	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "110100" "000110" "110010" "111100")
228	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
229	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
230	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
231	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "111100")
232	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "100010" "100100" "100001" "001111" "010010" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
233	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
234	("101100" "001110" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
235	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
236	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
237	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
238	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "000101" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
239	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
240	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
241	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
242	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
243	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
244	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "000101" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
245	("101100" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
246	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
247	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
248	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
249	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
250	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
251	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
252	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
253	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
254	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
255	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
256	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
257	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
258	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
259	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
260	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
261	("101100" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
262	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
263	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
264	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
265	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
266	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
267	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
268	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
269	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
270	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
271	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
272	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
273	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
274	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
275	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
276	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
277	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "111100")
278	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
279	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
280	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
281	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
282	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
283	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010110" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
284	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
285	("101100" "001110" "010101" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "111011" "000001" "001010" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
286	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "000010" "110100" "000110" "110010" "111100")
287	("101100" "110110" "001110" "010101" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "001100" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "110100" "000110" "111100")
288	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
289	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
290	("101100" "010001" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
291	("101100" "011111" "011101" "100101" "011011" "101010" "100111" "111000" "101011" "100001" "010110" "010010" "011100" "110000" "001001" "101110" "111111" "001100" "110001" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
292	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
293	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "110111" "000010" "101111" "110100" "000110" "111100")
294	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
295	("101100" "001110" "010001" "011101" "100101" "101010" "010111" "011001" "101011" "100100" "100001" "010110" "001111" "010010" "011100" "001001" "001100" "110101" "111011" "000001" "000111" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
296	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "000010" "110100" "000110" "110010" "111100")
297	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "110100" "000110" "110010" "111100")
298	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
299	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
300	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
301	("101100" "001110" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
302	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
303	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "110100" "000110" "110010" "111100")
304	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
305	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "011001" "100111" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "110100" "000110" "110010" "111100")
306	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
307	("101100" "001110" "010001" "010101" "100011" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
308	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
309	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "001100" "110001" "110101" "000001" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
310	("101100" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "000010" "110100" "000110" "111100")
311	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "000010" "110100" "000110" "111100")
312	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "111011" "001001" "111001" "101111" "110100" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
313	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "000101" "110100" "001100" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
	size 30
314	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
315	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
316	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
317	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
318	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
319	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
320	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
321	("101100" "110110" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
322	("101100" "110110" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
323	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
324	("101100" "110110" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
325	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
326	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
327	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
328	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "100110" "011110" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
329	("101100" "110110" "010001" "100011" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
330	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
331	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
332	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
333	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
334	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
335	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
336	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
337	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
338	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
339	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
340	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
341	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
342	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
343	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
344	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
345	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
346	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
347	("101100" "110110" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
348	("101100" "110110" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
349	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
350	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
351	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
352	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
353	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
354	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
355	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
356	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
357	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
358	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
359	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
360	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
361	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
362	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
363	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
364	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
365	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
366	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
367	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
368	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
369	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
370	("101100" "110110" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
371	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
372	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
373	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
374	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
375	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
376	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
377	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
378	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
379	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
380	("101100" "001110" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
381	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
382	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "010110" "010010" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
383	("101100" "010001" "010101" "100011" "011111" "101101" "011010" "010111" "111000" "101011" "100010" "100100" "010110" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
384	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
385	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
386	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
387	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
388	("101100" "001110" "010001" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
389	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "000010" "101111" "110100" "111100")
390	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
391	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
392	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
393	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
394	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
395	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
396	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
397	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
398	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
399	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
400	("101100" "001110" "010001" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "111010" "101111" "110100" "111100")
401	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
402	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
403	("101100" "001110" "010001" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
404	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
405	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
406	("101100" "001110" "010001" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
407	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
408	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
409	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
410	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
411	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
412	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
413	("101100" "001110" "010101" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
414	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
415	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
416	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
417	("101100" "001110" "010101" "100011" "011111" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
418	("101100" "110110" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
419	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
420	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
421	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
422	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
423	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "110010" "111100")
424	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
425	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
426	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "111100")
427	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
428	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "000010" "101111" "110100" "111100")
429	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
430	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
431	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
432	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
433	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
434	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
435	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "000010" "101111" "110100" "111100")
436	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
437	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
438	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
439	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
440	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
441	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
442	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
443	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
444	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "000101" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
445	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
446	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
447	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
448	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
449	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
450	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
451	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
452	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "111100")
453	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
454	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
455	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
456	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
457	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
458	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
459	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
460	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
461	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
462	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
463	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
464	("101100" "001110" "010001" "010101" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
465	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
466	("101100" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
467	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
468	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
469	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
470	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
471	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
472	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
473	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "110100" "000110" "111100")
474	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "110111" "000010" "101111" "110100" "000110" "111100")
475	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "111100")
476	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
477	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
478	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
479	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
480	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
481	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
482	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
483	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
484	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
485	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
486	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
487	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
488	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
489	("101100" "010101" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
490	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
491	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100001" "010110" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
492	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "000010" "110100" "000110" "110010" "111100")
493	("101100" "110110" "001110" "010101" "011101" "011011" "011010" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
494	("101100" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "110100" "000110" "110010" "111100")
495	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "110100" "000110" "111100")
496	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "111000" "101011" "100001" "010010" "011100" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
497	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
498	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
499	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
500	("101100" "011101" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100001" "010110" "010010" "011100" "110000" "001001" "101110" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
501	("101100" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100001" "010110" "010010" "011100" "110000" "001001" "101110" "111111" "001100" "110001" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
502	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
503	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "100111" "101011" "100010" "100100" "010010" "011100" "101001" "011110" "001001" "111111" "001100" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
504	("101100" "001110" "010001" "011101" "100101" "101010" "010111" "011001" "101011" "100100" "100001" "010110" "001111" "010010" "011100" "001001" "111111" "001100" "110101" "111011" "000001" "000111" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
505	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "110111" "000010" "110100" "000110" "110010" "111100")
506	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
507	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
508	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
509	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
510	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "110111" "000010" "101111" "110100" "110010" "111100")
511	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "110111" "000010" "101111" "110100" "110010" "111100")
512	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
513	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
514	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
515	("101100" "001110" "010001" "010101" "011111" "011101" "100101" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
516	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "110100" "000110" "110010" "111100")
517	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "011001" "100111" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
518	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "000111" "110111" "000010" "111010" "110100" "000110" "110010" "111100")
519	("101100" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000111" "110111" "000010" "110100" "110010" "111100")
520	("101100" "110110" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
521	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "000111" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
522	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
523	("101100" "110110" "001110" "010101" "011111" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
524	("101100" "010001" "010101" "011111" "101101" "011011" "011010" "010111" "101011" "100010" "100001" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "000111" "110111" "000010" "101111" "110100" "000110" "111100")
525	("101100" "010001" "010101" "011111" "101101" "011011" "011010" "010111" "101011" "100010" "100001" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000101" "111011" "011000" "000001" "001101" "110111" "000010" "101111" "110100" "000110" "111100")
526	("101100" "001110" "010001" "010101" "100011" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
527	("101100" "001110" "010001" "010101" "100011" "011011" "011010" "010111" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
528	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011011" "011010" "010111" "011001" "100111" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "110100" "000110" "110010" "111100")
529	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011011" "011010" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
530	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011011" "011010" "010111" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
531	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "010110" "010010" "110000" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "000010" "111010" "101111" "110100" "000110" "111100")
532	("101100" "001110" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
533	("101100" "001110" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "111011" "000001" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
534	("101100" "010001" "010101" "100011" "011111" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
535	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
536	("101100" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
537	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000001" "001101" "110111" "000010" "101111" "110100" "000110" "111100")
538	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "000010" "111010" "101111" "110100" "000110" "111100")
539	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
540	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "101101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "000010" "110100" "000110" "111100")
541	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "110100" "000110" "111100")
542	("101100" "010001" "011111" "011101" "100101" "101010" "010111" "011001" "100111" "111000" "101011" "100100" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "111001" "001010" "110111" "000010" "111010" "101111" "110100" "111100")
543	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011101" "100101" "010111" "011001" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "111111" "001100" "000101" "000001" "111001" "001010" "110111" "000010" "101111" "110100" "110010" "111100")
544	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011010" "011100" "101010" "100110" "010110" "110110" "110000" "011101" "110111" "111011" "111001" "101111" "110100" "111111" "001100" "111100" "001010" "000001" "101110" "001101" "000010" "110001")
545	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011010" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "111011" "001001" "111001" "101111" "110100" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
546	("100101" "010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "111011" "001001" "111001" "101111" "110100" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
547	("100101" "010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "000101" "110100" "001100" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
	size 31
548	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
549	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
550	("101100" "001110" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
551	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
552	("101100" "110110" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
553	("101100" "110110" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
554	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
555	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
556	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
557	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
558	("101100" "110110" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
559	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
560	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
561	("101100" "110110" "010001" "100011" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
562	("101100" "110110" "010001" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
563	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
564	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
565	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
566	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
567	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
568	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
569	("101100" "001110" "010001" "100011" "011111" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
570	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
571	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
572	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
573	("101100" "110110" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
574	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
575	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
576	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
577	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
578	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
579	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "111010" "101111" "110100" "111100")
580	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
581	("101100" "110110" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
582	("101100" "001110" "010001" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
583	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
584	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
585	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
586	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
587	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
588	("101100" "001110" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
589	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
590	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
591	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
592	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
593	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
594	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
595	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
596	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
597	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
598	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
599	("101100" "001110" "010001" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
600	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
601	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "000010" "101111" "110100" "110010" "111100")
602	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "110100" "111100")
603	("101100" "001110" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
604	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
605	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
606	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
607	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
608	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "000010" "101111" "110100" "110010" "111100")
609	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "111100")
610	("101100" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
611	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
612	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
613	("101100" "110110" "001110" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
614	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
615	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
616	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
617	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
618	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
619	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
620	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
621	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
622	("101100" "110110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
623	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
624	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
625	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
626	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
627	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
628	("101100" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
629	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
630	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
631	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
632	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
633	("101100" "110110" "001110" "010001" "010101" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
634	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
635	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
636	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
637	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
638	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "110100" "110010" "111100")
639	("101100" "110110" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
640	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
641	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
642	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "000010" "111010" "101111" "110100" "000110" "111100")
643	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
644	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
645	("101100" "010101" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
646	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010110" "010010" "011100" "110000" "100110" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
647	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
648	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
649	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
650	("101100" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
651	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "110100" "000110" "110010" "111100")
652	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "110100" "000110" "111100")
653	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
654	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
655	("101100" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
656	("101100" "011111" "011101" "100101" "011011" "101010" "011001" "100111" "111000" "101011" "100001" "010110" "010010" "011100" "110000" "001001" "101110" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
657	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
658	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "000010" "101111" "110100" "000110" "110010" "111100")
659	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
660	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "111001" "001010" "110111" "000010" "101111" "110100" "110010" "111100")
661	("101100" "001110" "010001" "010101" "100011" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
662	("101100" "110110" "001110" "010101" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "000111" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
663	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
664	("101100" "001110" "010001" "010101" "100011" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
665	("101100" "001110" "010101" "100011" "011111" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
666	("101100" "010001" "010101" "100011" "011111" "101101" "011011" "011010" "010111" "101011" "100010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000101" "111011" "011000" "000001" "001101" "000111" "110111" "000010" "101111" "110100" "000110" "111100")
667	("101100" "010001" "010101" "011111" "101101" "011011" "011010" "010111" "101011" "100010" "100001" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000101" "111011" "011000" "000001" "001101" "000111" "110111" "000010" "101111" "110100" "000110" "111100")
668	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
669	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "010110" "010010" "110000" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
670	("101100" "001110" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
671	("101100" "001110" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "111011" "000001" "001010" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
672	("101100" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
673	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "000010" "111010" "101111" "110100" "000110" "111100")
674	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
675	("101100" "010001" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
676	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
677	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010010" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
678	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "101101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "000010" "101111" "110100" "000110" "111100")
679	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "110100" "000110" "111100")
680	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "101101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "110100" "000110" "111100")
681	("101100" "010001" "100011" "011111" "011101" "100101" "101010" "010111" "011001" "100111" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "111100")
682	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011010" "011100" "101010" "100110" "010110" "110110" "110000" "011101" "110111" "111011" "111001" "101111" "110100" "111111" "001100" "111100" "001010" "011110" "000001" "101110" "001101" "000010" "110001")
683	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "101111" "110100" "111111" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
	size 32
684	("101100" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
685	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
686	("101100" "110110" "001110" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
687	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
688	("101100" "110110" "010001" "100011" "011111" "101101" "011010" "101010" "010111" "111000" "101011" "100010" "100100" "010110" "001111" "100110" "011110" "111111" "001100" "110001" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
689	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
690	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
691	("101100" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
692	("101100" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
693	("101100" "110110" "001110" "010001" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
694	("101100" "110110" "001110" "010001" "100011" "011111" "011010" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
695	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
696	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
697	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
698	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "010110" "001111" "010010" "011110" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
699	("101100" "110110" "001110" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
700	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
701	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
702	("101100" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "001001" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
703	("101100" "110110" "001110" "100011" "011111" "100101" "101010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "010110" "001111" "010010" "101110" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "001101" "111001" "110111" "000010" "101111" "110100" "111100")
704	("101100" "110110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110001" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
705	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
706	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
707	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "100110" "101001" "111111" "001100" "110101" "111011" "011000" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
708	("101100" "110110" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "111111" "001100" "000101" "111011" "011000" "000001" "001101" "111001" "001010" "110111" "000010" "101111" "110100" "111100")
709	("101100" "001110" "010001" "010101" "100011" "011111" "100101" "010111" "011001" "111000" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "000101" "111011" "011000" "000001" "111001" "110111" "000010" "101111" "110100" "110010" "111100")
710	("101100" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "111011" "000001" "001101" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
711	("101100" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
712	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
713	("101100" "001110" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
714	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
715	("101100" "110110" "001110" "010101" "011111" "011101" "011011" "011010" "010111" "100111" "111000" "101011" "100001" "010010" "011100" "110000" "100110" "101001" "101110" "111111" "001100" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
716	("101100" "010001" "010101" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "011100" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
717	("101100" "011111" "011101" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100001" "010110" "010010" "011100" "110000" "001001" "101110" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
718	("101100" "001110" "010001" "010101" "011111" "100101" "010111" "011001" "100111" "101011" "100010" "100100" "001111" "010010" "011100" "101001" "011110" "001001" "111111" "001100" "110101" "111011" "000001" "000111" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
719	("101100" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
720	("101100" "110110" "001110" "010101" "100011" "011111" "100101" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
721	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
722	("101100" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "011001" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
723	("101100" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "011001" "100111" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "000111" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
724	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "011001" "100111" "101011" "100100" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
725	("101100" "001110" "010001" "010101" "100011" "011111" "011010" "010111" "011001" "101011" "100010" "100100" "100001" "001111" "010010" "101001" "001001" "111111" "001100" "110101" "111011" "011000" "000001" "000111" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
726	("101100" "001110" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
727	("101100" "110110" "001110" "010101" "100011" "011111" "011011" "011010" "010111" "011001" "101011" "100010" "100001" "010010" "110000" "101001" "101110" "111111" "001100" "000101" "111011" "011000" "000001" "000111" "001010" "110111" "000010" "101111" "110100" "000110" "110010" "111100")
728	("101100" "010001" "010101" "100011" "011111" "101101" "011011" "011010" "010111" "101011" "100010" "100001" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "000101" "111011" "011000" "000001" "001101" "000111" "110111" "000010" "101111" "110100" "000110" "111100")
729	("101100" "001110" "010001" "010101" "100011" "011111" "011101" "011011" "011010" "010111" "011001" "100111" "101011" "100001" "010010" "110000" "101001" "001001" "111111" "001100" "111011" "000001" "000111" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
730	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "010110" "010010" "110000" "100110" "011110" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
731	("101100" "001110" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "111011" "000001" "001010" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
732	("101100" "010001" "010101" "100011" "011111" "011011" "011010" "010111" "100111" "111000" "101011" "100010" "100001" "010010" "110000" "100110" "101001" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
733	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "110111" "000010" "111010" "101111" "110100" "000110" "110010" "111100")
734	("101100" "010001" "100011" "011111" "100101" "011011" "101010" "010111" "011001" "100111" "111000" "101011" "100010" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "110101" "111011" "000001" "001101" "110111" "000010" "111010" "101111" "110100" "000110" "111100")
735	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
736	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "101101" "011010" "010111" "111000" "101011" "100100" "100001" "010110" "110000" "100110" "111111" "001100" "110001" "000101" "111011" "011000" "000001" "001101" "001010" "110111" "000010" "101111" "110100" "000110" "111100")
737	("101100" "010001" "100011" "011111" "011101" "100101" "101010" "010111" "011001" "100111" "111000" "101011" "100100" "100001" "010110" "010010" "110000" "001001" "111111" "001100" "110001" "111011" "000001" "001101" "111001" "001010" "110111" "000010" "111010" "101111" "110100" "111100")
738	("101100" "110110" "010001" "010101" "100011" "011111" "011101" "011010" "010111" "100111" "111000" "101011" "100100" "100001" "010010" "110000" "100110" "101001" "111111" "001100" "110001" "000101" "111011" "000001" "001101" "111001" "001010" "110111" "000010" "101111" "110100" "111100")
739	("100101" "010010" "010101" "101011" "010111" "100111" "101001" "111000" "011111" "100100" "101100" "011100" "100110" "110000" "011101" "100001" "110111" "111011" "001001" "111001" "000101" "101111" "110100" "111111" "001100" "111100" "001010" "000001" "101110" "001101" "000010" "110001")
740	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011010" "011100" "101010" "100110" "010110" "110110" "110000" "011101" "110111" "111011" "111001" "101111" "110100" "111111" "001100" "111010" "111100" "001010" "011110" "000001" "101110" "001101" "000010" "110001")
741	("100101" "010010" "101011" "010111" "011001" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "010110" "001110" "110110" "110000" "011101" "110111" "111011" "111001" "110010" "101111" "110100" "111111" "001100" "111010" "111100" "001010" "011110" "000001" "101110" "000010")
742	("010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011010" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "101111" "110100" "111111" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
743	("100101" "010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "110000" "011101" "110111" "111011" "001001" "111001" "101111" "110100" "111111" "001100" "111010" "111100" "001010" "011110" "000001" "001101" "000010" "110001")
744	("100101" "010010" "101011" "010111" "100111" "111000" "011111" "100100" "101100" "011100" "101010" "100110" "010110" "010001" "100010" "110000" "110111" "111011" "001001" "111001" "000101" "101111" "110100" "111111" "001100" "111100" "001010" "011110" "000001" "001101" "000010" "110001")

	The strictly non-polar sets of degree at most 4 and dimension at most 7, ordered according to (degree, dimension) and by size
	(4,7)
	size 64
745	("1000101" "1101010" "0010110" "1111000" "1110110" "0011000" "0000100" "1100100" "0001010" "0000110" "0011101" "0100101" "1011111" "0111111" "1001101" "0000001" "0111110" "0011001" "1100001" "1110111" "0110001" "0010010" "1101000" "0111100" "0101011" "1100000" "1011100" "1110010" "1101111" "0010011" "1001011" "1000100" "0000101" "1110000" "1010110" "1100101" "0111000" "1001100" "0100100" "1001010" "0110110" "0101010" "0100110" "0000011" "1011000" "1111111" "1011101" "0000010" "1110011" "0100001" "0010001" "1110001" "1010010" "0101111" "0111101" "0111001" "1010111" "1001111" "1001000" "1010011" "1101011" "0001011" "0011100" "0101110")