Counting problems on the hypercube

Guillermo Morales-Luna
Computer Science Section
CINVESTAV-IPN
gmorales@cs.cinvestav.mx

November 2, 2000

Abstract

The finite-dimensional hypercube has served as model of several processes in Electrical Engineering: from the most basic of circuit design till prototypical NP-complete problems, needless to mention its importance regarding communicactions and multi-processor computing. Any set in the hypercube can be realized as the set of satisfying assignments of a Boolean form. However, if we consider just conjunctive forms with clauses of a fixed length, the satisfying assignments form a *simple* set, i.e. a set which is the union of subspaces of a given dimension. In our approach we look towards the calculation of the cardinalities of simple sets. In the most general case, we have posed recurrence relations whose solutions are the simple sets and the sets in the hypercube that contain no a subspase of a given dimension.

Keywords: Complexity, optimization, hypercube geometry.

Resumen

El hipercubo de dimensión finita se ha presentado como modelo de diversos procesos en Ingeniería Eléctrica: desde los más elementales tocantes al diseño de circuitos, hasta la formulación de problemas completos-NP, ya sin mencionar la importancia del hipercubo en el anĺisis y el diseño de programas y dispositivos de cómputo en paralelo o distribuído. Cualquier conjunto en el hipercubo puede ser representado como el conjunto de asignaciones satisfactoras de alguna forma booleana. Sin embargo, si sólo se considera a formas conjuntivas cuyas cláusulas son de una longitud fija entonces los correspondientes conjuntos de asignaciones satisfactoras son *simples*, es decir, son uniones de subespacios de una misma longitud. Aquí planteamos el problema de calcular las cardinalidades de conjuntos simples. En el caso más general, planteamos ecuaciones de recurrencia cuyas soluciones son las cardinalidades buscadas y las cardinalidade de conjuntos en el hipercubo que no contienen a espacios de una dimensión determinada.

Palabras claves: Teoría de la complejidad, optimización, geometría del hipercubo.

1 Introduction

The *Satisfiability problem* (SAT) decides whether any given Boolean form is satisfiable. Its counting version (#SAT) calculates the number of satisfying assignments of an input Boolean form. Any such form determines a function whose domain is the n-dimensional hypercube where n is the number of variables appearing in the form and its counterimage is the set of truth values $\{0,1\}$. For any Boolean form $\phi(\mathbf{X})$ the set of satisfying assignments is $\phi^{-1}(1)$ and the set of falsifying assignments is $\phi^{-1}(0)$. SAT decides whether $\phi^{-1}(1) \neq \emptyset$ and #SAT asks for the value of $\mathrm{card}\phi^{-1}(1)$.

#SAT has been treated by several approaches ([1], [2], [6]), and they already constitute classical treatments. Most of the procedures have followed an approach based on the inclusion-exclusion counting formula ([3], [4], [5]). We pose in this paper the problem to characterize the numbers that can appear as solutions of #SAT restricted to conjunctive forms with an uniform clause length. We initiate our analysis by posing recurrent relations whose solutions are the sought values.

2 Preliminaries

Let Two $= \{0,1\}$ be the basic set of Boolean truth-values, and for any $n \geq 1$, let the *n-dimensional hypercube* be the Cartesian power Two^n. Naturally, the hypercube is the valuation set of any Boolean form over n variables. For any given set of *variables* $\mathbf{X} = [X_j]_{j=1,\dots,n}$ let us introduce the following notions:

Literals Variables or negated variables:

$$L \text{ literal if } \exists X \in \mathbf{X} : (L = X) \vee (L = \neg X).$$

Phrases Conjunctions of literals: $\forall \boldsymbol{\epsilon} \in$ Three^n, where Three $=$ Two $\cup \{\hat{\ }\}$, let

$$\mathrm{Phrase}(\boldsymbol{\epsilon}) = \bigwedge_{1 \leq j \leq n} X_j^{\epsilon_j}$$

where

$$\forall j = 1,\dots,n : X_j^{\epsilon_j} = \begin{cases} X_j & \text{if } \epsilon_j = 1 \\ 0 & \text{if } \epsilon_j = \hat{\ } \\ \neg X_j & \text{if } \epsilon_j = 0 \end{cases}$$

Thus

$$\epsilon_j = \begin{cases} 1 & \text{if } X_j \text{ appears in Phrase}(\boldsymbol{\epsilon}), \\ 0 & \text{if } \neg X_j \text{ appears in Phrase}(\boldsymbol{\epsilon}), \\ \hat{\ } & \text{if neither } X_j \text{ nor } \neg X_j \text{ appear in Phrase}(\boldsymbol{\epsilon}). \end{cases}$$

Evidently $\forall \boldsymbol{\delta} \in$ Two^n :

$$\mathrm{Phrase}(\boldsymbol{\epsilon})(\boldsymbol{\delta}) = 1 \iff \boldsymbol{\epsilon} = \boldsymbol{\delta} \qquad (1)$$

in other words $\{\boldsymbol{\epsilon}\} = \mathrm{Phrase}(\boldsymbol{\epsilon})^{-1}(1)$.

Disjunctive forms Disjunctions of phrases. If $DF = \bigvee_{1 \leq i \leq m} Ph_i = \bigvee_{1 \leq i \leq m} \bigwedge_{1 \leq j \leq n} L_{ij}$ let us write $DF = [L_{ij}]_{i,j}^{D}$. A DF is True if at least one of its phrases is True. The DF is False if each of its phrases is False. In symbols:

$$\begin{aligned} DF^{-1}(1) &= \bigvee_{i=1}^{m} Ph_i^{-1}(1) \\ DF^{-1}(0) &= \bigwedge_{i=1}^{m} Ph_i^{-1}(0) \end{aligned} \qquad (2)$$

Clauses Disjunctions of literals: $\forall \boldsymbol{\epsilon} \in$ Three^n, let

$$\mathrm{Clause}(\boldsymbol{\epsilon}) = \bigvee_{1 \leq j \leq n} \overline{X_j^{\epsilon_j}}.$$

Thus

$$\epsilon_j = \begin{cases} 0 & \text{if } X_j \text{ appears in Clause}(\boldsymbol{\epsilon}), \\ 1 & \text{if } \neg X_j \text{ appears in Clause}(\boldsymbol{\epsilon}), \\ \hat{\ } & \text{if neither } X_j \text{ nor } \neg X_j \text{ appear in Clause}(\boldsymbol{\epsilon}). \end{cases}$$

Evidently $\forall \boldsymbol{\delta} \in$ Two^n :

$$\mathrm{Clause}(\boldsymbol{\epsilon})(\boldsymbol{\delta}) = 0 \iff \boldsymbol{\epsilon} = \boldsymbol{\delta} \qquad (3)$$

in other words $\{\boldsymbol{\epsilon}\} = \mathrm{Clause}(\boldsymbol{\epsilon})^{-1}(0)$.

Conjunctive forms Conjunctions of clauses. If $CF = \bigwedge_{1\leq i\leq m} C_i = \bigwedge_{1\leq i\leq m}\bigvee_{1\leq j\leq n} L_{ij}$ let us write $CF = [L_{ij}]^C_{i,j}$. A CF is False if at least one of its clauses is False. The CF is True if each of its clauses is True.

$$\begin{aligned} CF^{-1}(0) &= \bigvee_{i=1}^m C_i^{-1}(0) \\ CF^{-1}(1) &= \bigwedge_{i=1}^m C_i^{-1}(1) \end{aligned} \quad (4)$$

Boolean forms Any conjunctive or disjunctive form.

Each Boolean form F determines a function $F : \text{Two}^n \to \text{Two}$. A Boolean form F is *satisfiable* if $F \neq \mathbf{0}$, *tautological* if $F = \mathbf{1}$, *refutable* if $F \neq \mathbf{1}$ and a *contradiction* if $F = \mathbf{0}$.

Given a set of points $A \subset \text{Two}^n$, the *disjunctive form associated to* A is

$$DF_A(\mathbf{X}) = \bigvee_{\boldsymbol{\epsilon}\in A} \text{Phrase}(\boldsymbol{\epsilon})(\mathbf{X}). \quad (5)$$

Clearly, $A = DF_A^{-1}(1)$, i.e. A consists of the points that satisfy its associated DF. Similarly, the *conjunctive form associated to* A is

$$CF_A(\mathbf{X}) = \bigwedge_{\boldsymbol{\epsilon}\in A} \text{Clause}(\boldsymbol{\epsilon})(\mathbf{X}). \quad (6)$$

Clearly, $A = CF_A^{-1}(0)$, i.e. A consists of the points that falsify its associated CF.

The following are classical problems:

Problem SAT

Instance: A CF F.

Solution: $\begin{cases} \textit{Yes} & \text{if } F \text{ is satisfiable} \\ \textit{No} & \text{otherwise} \end{cases}$

Problem #SAT

Instance: A CF F.

Solution: $\text{card}\{\boldsymbol{\epsilon} \in \text{Two}^n | F(\boldsymbol{\epsilon}) = 1\}$

Problem MaxSAT

Instance: A CF F.

Solution: An assignment $\boldsymbol{\epsilon}_0 \in \text{Two}^n$ such that $\forall \boldsymbol{\epsilon} \in \text{Two}^n$:

$$\begin{aligned} & \text{card}\{c \in F | c(\boldsymbol{\epsilon}_0) = 1\} \\ \geq\ & \text{card}\{c \in F | c(\boldsymbol{\epsilon}) = 1\} \end{aligned}$$

Problem co-SAT

Instance: A DF F.

Solution: $\begin{cases} \textit{Yes} & \text{if } F \text{ is a tautology} \\ \textit{No} & \text{otherwise} \end{cases}$

Let $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n) \in \text{Three}^n$. Let the set of *instantiated variables by* $\boldsymbol{\epsilon}$ be the set of indexes defined as:

$$\begin{aligned} \text{Def}(\boldsymbol{\epsilon}) &= [\![1, n]\!] - \boldsymbol{\epsilon}^{-1}(\hat{\ }) \\ &= \{i \in [\![1, n]\!] | \epsilon_i \in \text{Two}\} \end{aligned}$$

If $k = \text{card Def}(\boldsymbol{\epsilon})$ then $\boldsymbol{\epsilon}$ determines a hypercube

$$H(\boldsymbol{\epsilon}) = \{\boldsymbol{\delta} \in \text{Two}^n | \forall i \in \text{Def}(\boldsymbol{\epsilon})\, [\delta_i = \epsilon_1]\},$$

and it is a subspace whose dimension, as an affine variety in Two^n, is $n - k$. In this case, we will say that $\boldsymbol{\epsilon} \in \text{Three}^n$ is a k-*assignment*. Evidently

$$\text{Phrase}(\boldsymbol{\epsilon})^{-1}(1) = H(\boldsymbol{\epsilon}) = \text{Clause}(\boldsymbol{\epsilon})^{-1}(0).$$

A k-*phrase* (resp. k-*clause*) is of the form $\text{Phrase}(\boldsymbol{\epsilon})$ (resp. $\text{Clause}(\boldsymbol{\epsilon})$), where $\boldsymbol{\epsilon}$ is a k-assignment. In Circuit Design, a k-phrase is called also a k-*miniterm*.

A k-DF (resp.k-CF), $k \geq 1$, is a DF (resp. CF) consisting just of k-phrases (resp. k-clauses). kSAT, #kSAT, Max$k$SAT and k-co-SAT are the corresponding problems restricted to k-forms.

Thus the sets of assignments satisfying a k-DF (resp. falsifying a k-CF) is the union of $(n-k)$-dimensional subspaces.

In this paper we will deal with the problem to count the number of sets that can be expressed as the union of k_1-dimensional subspaces, for any given $k_1 \in [\![1,n]\!]$, or, stated equivalently, are the satisfying sets of k_2-DF's, or falsifying sets of k_2-CF's, with $k_1 + k_2 = n$.

3 Some counting problems

Let n be the number of variables, $k \leq n$ be the clauses length, and $C_n^{(k)}$ be the set of k-clauses over n variables. There are $c_{nk} = \binom{n}{k}2^k$ elements in $C_n^{(k)}$.

3.1 Pairs of clauses with common literals

First of all, let us deal with the following

Problema 3.1 *Given $l \in [\![0,k]\!]$, calculate the number p_{lkn} of pairs in $\left(C_n^{(k)}\right)^{(2)}$ whose components share exactly l literals.* In other words: *Count the pairs of clauses sharing exactly l literals.*

Let us consider first the case where $k \in [\![1, \frac{n}{2}[\![$.

If $\{\mathbf{c}_1, \mathbf{c}_2\} \in \left(C_n^{(k)}\right)^{(2)}$ is such a pair let us denote by $\mathbf{d}$ the list of the common literals. Then we may write

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{d} \vee \mathbf{c}'_1 \\ \mathbf{c}_2 &= \mathbf{d} \vee \mathbf{c}'_2 \end{aligned}$$

where $\mathbf{c}'_1$ and $\mathbf{c}'_2$ are $(k-l)$-clauses. Clearly,

1. there are $\binom{n}{l}2^l$ possibilities to select the common l-clause $\mathbf{d}$,
2. there are $\binom{n-l}{k-l}2^{k-l}$ possibilities to select the first remaining $(k-l)$-clause $\mathbf{c}'_1$,
3. there are $\binom{n-k}{k-l}2^{k-l}$ possibilities to select the second remaining $(k-l)$-clause $\mathbf{c}'_2$, and
4. for $l < k$ each pair of the remaining $(k-l)$-clauses $\{\mathbf{c}'_1, \mathbf{c}'_2\}$ is counted exactly twice by the above counting process.

Thus, let

$$\begin{aligned} q_{lkn} &= \binom{n}{l}2^l \cdot \binom{n-l}{k-l}2^{k-l} \cdot \binom{n-k}{k-l}2^{k-l} \\ &= \binom{n}{l}\binom{n-l}{k-l}\binom{n-k}{k-l}2^{2k-l} \end{aligned}$$

then

$$p_{lkn} = \begin{cases} \frac{1}{2}q_{lkn} & \text{if } l < k \\ q_{kkn} & \text{if } l = k \end{cases}$$

indeed, $q_{kkn} = c_{nk}$.

In table 1 we present some values of q_{lkn} with $0 \leq q \leq k$ and $1 \leq k < \frac{n}{2}$. In table 2, the prime factorization of these numbers is shown.

Now, if $k \in [\![\frac{n}{2}, n]\!]$ then p_{lkn} may be computed as before whenever $n-k \geq k-l$. Thus, for the current case one must have

$$p_{lkn} = \begin{cases} 0 & \text{if } 0 \leq l < 2k-n \\ \frac{1}{2}q_{lkn} & \text{if } 2k-n \leq l < k \\ q_{kkn} & \text{if } l = k \end{cases}$$

3.2 k-representability

3.2.1 Basic geometrical notions

Let $0 \leq k \leq n$. Let us say that a set $A \subset Two^n$

$$n=2 \quad : \quad \left[\begin{array}{c|cc} k\backslash l & 0 & 1 \\ \hline 1 & 4 & 4 \end{array}\right]$$

$$n=3 \quad : \quad \left[\begin{array}{c|cc} k\backslash l & 0 & 1 \\ \hline 1 & 12 & 6 \end{array}\right]$$

$$n=4 \quad : \quad \left[\begin{array}{c|ccc} k\backslash l & 0 & 1 & 2 \\ \hline 1 & 24 & 8 & \\ 2 & 48 & 96 & 24 \end{array}\right]$$

$$n=5 \quad : \quad \left[\begin{array}{c|ccc} k\backslash l & 0 & 1 & 2 \\ \hline 1 & 40 & 10 & \\ 2 & 240 & 240 & 40 \end{array}\right]$$

$$n=6 \quad : \quad \left[\begin{array}{c|cccc} k\backslash l & 0 & 1 & 2 & 3 \\ \hline 1 & 60 & 12 & & \\ 2 & 720 & 480 & 60 & \\ 3 & 640 & 2880 & 1440 & 160 \end{array}\right]$$

$$n=7 \quad : \quad \left[\begin{array}{c|cccc} k\backslash l & 0 & 1 & 2 & 3 \\ \hline 1 & 84 & 14 & & \\ 2 & 1680 & 840 & 84 & \\ 3 & 4480 & 10080 & 3360 & 280 \end{array}\right]$$

$$n=8 \quad : \quad \left[\begin{array}{c|ccccc} k\backslash l & 0 & 1 & 2 & 3 & 4 \\ \hline 1 & 112 & 16 & & & \\ 2 & 3360 & 1344 & 112 & & \\ 3 & 17920 & 26880 & 6720 & 448 & \\ 4 & 8960 & 71680 & 80640 & 17920 & 1120 \end{array}\right]$$

$$n=9 \quad : \quad \left[\begin{array}{c|ccccc} k\backslash l & 0 & 1 & 2 & 3 & 4 \\ \hline 1 & 144 & 18 & & & \\ 2 & 6048 & 2016 & 144 & & \\ 3 & 53760 & 60480 & 12096 & 672 & \\ 4 & 80640 & 322560 & 241920 & 40320 & 2016 \end{array}\right]$$

$$n=10 \quad : \quad \left[\begin{array}{c|cccccc} k\backslash l & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 180 & 20 & & & & \\ 2 & 10080 & 2880 & 180 & & & \\ 3 & 134400 & 120960 & 20160 & 960 & & \\ 4 & 403200 & 1075200 & 604800 & 80640 & 3360 & \\ 5 & 129024 & 1612800 & 3225600 & 1612800 & 201600 & 8064 \end{array}\right]$$

Table 1: Numbers of k-clause pairs, over n variables, with l-common literals.

$$n=2 \quad : \quad \begin{bmatrix} 2^2 & 2^2 \end{bmatrix}$$

$$n=3 \quad : \quad \begin{bmatrix} 2^2 \cdot 3^1 & 2^1 \cdot 3^1 \end{bmatrix}$$

$$n=4 \quad : \quad \begin{bmatrix} 2^3 \cdot 3^1 & 2^3 & \\ 2^4 \cdot 3^1 & 2^5 \cdot 3^1 & 2^3 \cdot 3^1 \end{bmatrix}$$

$$n=5 \quad : \quad \begin{bmatrix} 2^3 \cdot 5^1 & 2^1 \cdot 5^1 & \\ 2^4 \cdot 3^1 \cdot 5^1 & 2^4 \cdot 3^1 \cdot 5^1 & 2^3 \cdot 5^1 \end{bmatrix}$$

$$n=6 \quad : \quad \begin{bmatrix} 2^2 \cdot 3^1 \cdot 5^1 & 2^2 \cdot 3^1 & & \\ 2^4 \cdot 3^2 \cdot 5^1 & 2^5 \cdot 3^1 \cdot 5^1 & 2^2 \cdot 3^1 \cdot 5^1 & \\ 2^7 \cdot 5^1 & 2^6 \cdot 3^2 \cdot 5^1 & 2^5 \cdot 3^2 \cdot 5^1 & 2^5 \cdot 5^1 \end{bmatrix}$$

$$n=7 \quad : \quad \begin{bmatrix} 2^2 \cdot 3^1 \cdot 7^1 & 2^1 \cdot 7^1 & & \\ 2^4 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^3 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^2 \cdot 3^1 \cdot 7^1 & \\ 2^7 \cdot 5^1 \cdot 7^1 & 2^5 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^5 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^3 \cdot 5^1 \cdot 7^1 \end{bmatrix}$$

$$n=8 \quad : \quad \begin{bmatrix} 2^4 \cdot 7^1 & 2^4 & & & \\ 2^5 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^1 \cdot 7^1 & 2^4 \cdot 7^1 & & \\ 2^9 \cdot 5^1 \cdot 7^1 & 2^8 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 7^1 & \\ 2^8 \cdot 5^1 \cdot 7^1 & 2^1 1 \cdot 5^1 \cdot 7^1 & 2^8 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^9 \cdot 5^1 \cdot 7^1 & 2^5 \cdot 5^1 \cdot 7^1 \end{bmatrix}$$

$$n=9 \quad : \quad \begin{bmatrix} 2^4 \cdot 3^2 & 2^1 \cdot 3^2 & & & \\ 2^5 \cdot 3^3 \cdot 7^1 & 2^5 \cdot 3^2 \cdot 7^1 & 2^4 \cdot 3^2 & & \\ 2^9 \cdot 3^1 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^3 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^3 \cdot 7^1 & 2^5 \cdot 3^1 \cdot 7^1 & \\ 2^8 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^1 0 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^8 \cdot 3^3 \cdot 5^1 \cdot 7^1 & 2^7 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^5 \cdot 3^2 \cdot 7^1 \end{bmatrix}$$

$$n=10 \quad : \quad \begin{bmatrix} 2^2 \cdot 3^2 \cdot 5^1 & 2^2 \cdot 5^1 & & & & \\ 2^5 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^2 \cdot 5^1 & 2^2 \cdot 3^2 \cdot 5^1 & & & \\ 2^8 \cdot 3^1 \cdot 5^2 \cdot 7^1 & 2^7 \cdot 3^3 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^6 \cdot 3^1 \cdot 5^1 & & \\ 2^8 \cdot 3^2 \cdot 5^2 \cdot 7^1 & 2^1 1 \cdot 3^1 \cdot 5^2 \cdot 7^1 & 2^7 \cdot 3^3 \cdot 5^2 \cdot 7^1 & 2^8 \cdot 3^2 \cdot 5^1 \cdot 7^1 & 2^5 \cdot 3^1 \cdot 5^1 \cdot 7^1 & \\ 2^1 1 \cdot 3^2 \cdot 7^1 & 2^1 0 \cdot 3^2 \cdot 5^2 \cdot 7^1 & 2^1 1 \cdot 3^2 \cdot 5^2 \cdot 7^1 & 2^1 0 \cdot 3^2 \cdot 5^2 \cdot 7^1 & 2^7 \cdot 3^2 \cdot 5^2 \cdot 7^1 & 2^7 \cdot 3^2 \cdot 7^1 \end{bmatrix}$$

Table 2: Prime factorization of numbers of k-clause pairs, over n variables, with l-common literals.

- is *k-representable* if there is a collection $S_1, \ldots, S_m$ of k-subspaces such that $A = \bigcup_{i=1}^{m} S_i$, and
- is *k-thin* if no k-subspace is contained in A.

The following relations are self-evident:

1. The empty set $\emptyset$ is k-representable, since it is represented as an empty union of k-subspaces, and it is also k-thin.
2. The whole space Two^n is k-representable.
3. Any non-k-representable set is the disjoint union of a k-representable set and a non-empty k-thin set.
4. If A is k-representable then it has at least 2^k points, i.e. $\mathrm{card}(A) \geq 2^k$.
5. Since Two^n can be partitioned as the union of 2^{n-k} pairwise independent k-subspaces $S_0^{(k)}, \ldots, S_{2^{n-k}-1}^{(k)}$, any k-thin set A can be partitioned as a collection of 2^{n-k} pairwise disjoint sets, each with less than 2^k points, thus

$$\begin{array}{ll} & \exists \{l_i\}_{i=1}^{2^{n-k}-1} : \\ & \left[0 \leq l_0 \leq l_1 \leq \cdots \leq l_{2^{n-k}-1} < 2^k\right] \\ \& & \left[\mathrm{card}(A) = \sum_{i=1}^{2^{n-k}-1} l_i\right]. \end{array} \tag{7}$$

Let us define

$$\begin{array}{rcl} \mathcal{R}_{kn} & = & \{A \subset Two^n | A \text{ is } k\text{-representable}\} \\ \rho_{kn} & = & \mathrm{card}(\mathcal{R}_{kn}), \\ \mathcal{N}_{kn} & = & \{A \subset Two^n | A \text{ is not } k\text{-representable}\} \\ \eta_{kn} & = & \mathrm{card}(\mathcal{N}_{kn}), \\ \mathcal{T}_{kn} & = & \{A \subset Two^n | A \text{ is } k\text{-thin}\} \\ \tau_{kn} & = & \mathrm{card}(\mathcal{T}_{kn}). \end{array}$$

For $k = 0$ the following relations hold:

$$\begin{array}{rcl} \mathcal{R}_{0n} & = & \mathcal{P}(Two^n), \\ \mathcal{T}_{0n} & = & \emptyset, \\ \mathcal{T}_{nn} & = & \mathcal{P}(Two^n) - \{Two^n\}, \end{array}$$

thus

$$\begin{array}{rcl} \rho_{0n} & = & 2^{2^n}, \\ \tau_{0n} & = & 0, \\ \tau_{nn} & = & 2^{2^n} - 1. \end{array}$$

Besides, after 7 we get an estimation of τ_{kn}:

$$\tau_{kn} = \sum_{a=1}^{2^n-1} \sum_{\substack{l_0+\cdots+l_{2^{n-k}-1}=a \\ 0 \leq l_i < 2^k}} \binom{2^k}{l_0} \cdots \binom{2^k}{l_{2^{n-k}-1}}. \tag{8}$$

Let us make some remarks about the summands and the values of the above sum. Let

$$\begin{array}{rcl} M_{akn} & = & \left\{ (l_0, \ldots, l_{2^{n-k}-1}) \in I\!N^{2^{n-k}} | \right. \\ & & \left. \sum_{j=0}^{2^{n-k}-1} l_j = a \right\}, \\ L_{akn} & = & \left\{ (l_0, \ldots, l_{2^{n-k}-1}) \in M_{akn} | \right. \\ & & \left. \forall j \leq (2^{n-k} - 1) : l_j < 2^k \right\}. \end{array}$$

1. If $a > (2^k - 1) \cdot 2^{n-k}$ then $L_{akn} = \emptyset$. Thus 8 can be written as

$$\tau_{kn} = \sum_{a=1}^{(2^k-1)\cdot 2^{n-k}} \sum_{\substack{l_0+\cdots+l_{2^{n-k}-1}=a \\ 0 \leq l_i < 2^k}} \binom{2^k}{l_0} \cdots \binom{2^k}{l_{2^{n-k}-1}}. \tag{9}$$

2. For $k = 1$, a tuple $(l_0, \ldots, l_{2^{n-1}-1})$ is in the set L_{a1n} iff it is the characteristic function of a subset with a elements in

$\{0,\ldots,2^{n-1}-1\}$. Thus, 9 is equivalent to the equation

$$\tau_{1n} = \sum_{a=1}^{2^{n-1}} \binom{2^{n-1}}{a} \binom{2}{1}^a.$$

Hence, $\tau_{1n} = 3^{2^{n-1}} - 1$.

3. If $1 \le a \le (2^k - 1)$ then $L_{akn} = M_{akn}$.

4. For any a, k, n the set M_{akn} can be expressed as the disjoint union of

   $\binom{a-1}{s-1}$: number of possibilities to express a as the sum of s non-nul summands,

   sets, each consisting of

   $\binom{2^{n-k}}{s}$: number of possibilities to put s non-nul summands into a sum of 2^{n-k} non-negative summands.

   elements. s varies from 1 to a.

3.2.2 Orders

For any set non-empty set $A \subset Two^n$ let the *order* of A be $o(A) = \mathrm{Min}\{k | A \in \mathcal{T}_{kn}\} - 1$. Let

$$\begin{array}{rcl} \mathcal{U}_{kn} & = & \{A \subset Two^n | o(A) = k\} \\ \omega_{kn} & = & \mathrm{card}(\mathcal{U}_{kn}), \\ \forall l \le 2^n : \mathcal{V}_{lkn} & = & \{A \in \mathcal{U}_{kn} | \mathrm{card}(A) = l\} \\ \upsilon_{lkn} & = & \mathrm{card}(\mathcal{V}_{lkn}). \end{array}$$

The following relations hold:

1. $\{\mathcal{V}_{lkn}\}_{1\le l\le 2^n}$ is a partition of $\mathcal{U}_{kn}$, thus $\omega_{kn} = \sum_{i=1}^{2^n} \upsilon_{lkn}$.

2. For $k = 0$,

   $\upsilon_{10n} = 2^n$
   : Any singleton has order 0.

   $\upsilon_{2^{n-1}0n} = 2$
   : The only way to choose 2^{n-1} points that form no an edge is by selecting opposite points along the main diagonal on 2-dimensional cubes.

   $l > 2^{n-1} \Rightarrow \upsilon_{l0n} = 0$
   : Any set with more than 2^{n-1} points has at least an edge.

3. For any k with $0 < k \le n$,

   $l < 2^k \Rightarrow \upsilon_{lkn} = 0$
   : Any set of order k should have, at least, 2^k points.

   $\upsilon_{2^k kn} = 2^{n-k}\binom{n}{k}$
   : This is the number of k-dimensional cubes in the n-dimensional cube.

   $l > 2^{n-k}(2^k - 1) \Rightarrow \upsilon_{lkn} = 0$
   : Any set with more than $2^{n-k}(2^k - 1)$ points has at least a k-dimensional cube.

4. For any $k < n$, $\upsilon_{2^{n-k}(2^k-1),k,n} = 2^k$.

4 Final remarks

In tis paper we have posed several problems related to counting representable sets as well as sets of a given order.

Just a few results are presented here. Indeed the whole calculation of the defined parameters has not been completed.

Such calculation may provide an approach to solving #SAT decision problems: Given a CF ϕ and a $N \in I\!N$ we could design a procedure to decide whether $N \leq M$, where M is the number of assignments falsifying ϕ, by looking at the geometric properties of this last set.

There is still a long way to follow-up in order to complete the calculation of the parameters introduced here.

References

[1] Creignou N., Hermann M., *Complexity of Generalized Satisfiability Counting Problems*, Information and Computation 125, 1-12, 1996.

[2] Dubois Olivier, *Counting the number of solutions for instances of satisfiability*, Theor. Comp. Sc. 81, 49-64, 1991.

[3] Kahn J., Linial N., Samorodnitsky A., *Inclusion-exclusion: exact and approximate*, Combinatorica 16(4), 465-477, 1996.

[4] Linial N., Nissan N., *Approximate inclusion-exclusion*, Combinatorica 10(4), 349-365, 1990.

[5] Lozinskii E.L., *Counting propositional models*, Information Processing Letters 41, 327-332, 1992.

[6] Toda S., *On the computational power of PP and* $\bigoplus P$, Proc. 30th IEEE Ann. Symp. on Foundations of Computer Science, 514-519, 1989.