Inclusion-Exclusion for k-CNF Formulas

Kazuo Iwama† and Akihiro Matsuura†

†School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan
†{iwama, matsu}@kuis.kyoto-u.ac.jp

Abstract

We show that the number of satisfying assignments of a k-CNF formula is determined uniquely from the numbers of unsatisfying assignments for clause-sets of size up to $\lfloor \log k \rfloor + 2$. The information of this size is also shown to be necessary.

key words: combinatorial problems; SAT; k-CNF formula; counting; inclusion-exclusion

1 Introduction

The Satisfiability Problem (SAT) is one of the well-known NP-complete problems and has been studied extensively from various viewpoints. #SAT (or #k-SAT) is the problem of counting the number of satisfying assignments of a CNF (or k-CNF, resp.) formula and is shown to be #P-complete [6]. Thus they are believed be quite hard to solve exactly. A naive way for this counting is possible by the so-called *inclusion-exclusion* formula. Namely, suppose that a CNF formula has m clauses and A_i is the set of truth assignments that falsify the ith clause of the formula. Then the total number of unsatisfying assignments can be computed as follows:

$$|\bigcup_{i=1}^{m} A_i| = \sum_{i=1}^{m} |A_i| - \sum_{i<j} |A_i \bigcap A_j| + \sum_{i<j<k} |A_i \bigcap A_j \bigcap A_k| - \cdots + (-1)^{m-1} |\bigcap_{i=1}^{m} A_i|, \tag{1}$$

where $|A_i|$ is the number of unsatisfying assignments for the ith clause. Since there are 2^m terms in the formula, the direct computation can take exponential time. On the average case, it is known to be possible in polynomial or pseudopolynomial time under some reasonable assumptions [1], [4]. Furthermore, when approximation is permitted, almost tight error bounds for approximating the value $|\bigcup_{i=1}^{m} A_i|$ using only the numbers $\{|\bigcap_{i\in S} A_i|\}$'s over all S such that $|S| \leq l$ are known [2], [3]. In general, there can be some error even when such numbers are used up to $l = m - 1$ for the approximation. Remarkably, however, when the number of satisfying assignments of CNF (or equivalently, DNF) formulas with n variables is targeted, Kahn et al. showed [2] that the information over all S such that $|S| \leq \log n + 1$ already uniquely determines the number of satisfying assignments of the formula. In other words, if the numbers of unsatisfying assignments of clause-sets of size up to $\log n + 1$ coincide at any two CNF formulas, these formulas have exactly the same number of satisfying assignments. Though the original result was only existential, the procedure for computing $|\bigcup_{i}^{m} A_i|$ from $\{|\bigcap_{i\in S} A_i|\}$'s ($|S| \leq \log n + 1$) was later shown by A. A. Melkman et al. [5].

In this paper, we extend this problem of uniquely determining the number of satisfying assignments to k-CNF formulas for any constant $k \geq 2$. Then we show that the information for clause-sets of only *constant* size, i.e., size up to $\lfloor \log k \rfloor + 2$ is necessary and sufficient. We also briefly mention dual representations of hypergraphs as possible application of the results for k-CNF formulas.

The organization of the paper is as follows: In Section 2, we summarize the previous work on uniquely determining the number satisfying assignments and make some definitions. Section 3 is devoted to the main theorem and its proof. In Section 4, we relate our results on k-CNF formulas to dual representations of hypergraphs. Finally, concluding remarks are given in Section 5.

2 Previous Work and Definitions

As is mentioned in the Introduction, exactly counting the number of satisfying assignments using the inclusion-exclusion formula can take exponential time because of the number of terms in it. On the other hand, when approximation is permitted, a nice result in a more general setting is known.

Proposition 2.1. ([2]) *Let $A_1, \cdots, A_m$ and $B_1, \cdots, B_m$ be collections of events in some probability space where*

$$Pr(\bigcap_{i \in S} A_i) = Pr(\bigcap_{i \in S} B_i)$$

for every subset S in $[m] = \{1, 2, \cdots, m\}$ such that $|S| \leq l$. Then

$$\left| Pr(\bigcup_{i=1}^{m} A_i) - Pr(\bigcup_{i=1}^{m} B_i) \right| = e^{-\Omega(\frac{l^2}{m \log m})}.$$

When $k = m - 1$, this error bound is improved to be $(1/2)^{m-1}$ [2], [3]. Thus, if we naively apply this approximation result to CNF formulas, there might be some error even when the information up to $m - 1$ clauses is used. Remarkably, however, Kahn et al. have shown that the number of unsatisfying assignments of a CNF formula is already uniquely determined by the numbers $\{|\bigcap_{i \in S} A_i|\}$'s over all $S \subset [m]$ such that $|S| \leq \log n + 1$. The original theorem was stated for DNF formulas. Here we restate the theorem in terms of CNF formulas in the equivalent way.

Theorem 2.2. ([2]) *Let ϕ be a CNF formula with n variables and m clauses. Then the numbers $|\bigcap_{i \in S} A_i|$ where S ranges over all subsets of $[m]$ such that $|S| \leq \log n + 1$ already uniquely determine $|\bigcup_{i=1}^{m} A_i|$.*

Motivated by this, we extend the problem to k-CNF formulas for any constant $k \geq 2$ and consider the minimum size of clause-sets needed to determine the number of satisfying assignments uniquely. The definition is as follows.

Definition 2.3. *Let ϕ and ψ be k-CNF formulas with n variables and m clauses. Let A_i and B_i be the numbers of unsatisfying assignments of the ith clause of ϕ and ψ, respectively.*

Then, $\lambda_k(n)$ is defined as follows:

$$\lambda_k(n) = \min_{\phi,\psi} \left\{ l \;\middle|\; |\bigcap_{i\in S} A_i| = |\bigcap_{i\in S} B_i| \textit{ for } \forall S \textit{ s.t. } |S| \le l \textit{ result in } |\bigcup_{i=1}^{m} A_i| = |\bigcup_{i=1}^{m} B_i|. \right\}$$

Note by Theorem 2.2 that $\lambda_k(n)$ does not depend on m and $\lambda_n(n) \le \log n + 1$. For brevity, we denote $\lambda_k(n)$ by λ_k.

3 Main Results

In this section, we show that the information for clause-sets of only *constant* size uniquely determine the number of solutions. Namely, we show the following main theorem.

Theorem 3.1. *For any constant $k \ge 2$, $\lambda_k = \lfloor \log k \rfloor + 2$.*

3.1 The Upper Bounds

Let $\phi = c_1 \cdot c_2 \cdots c_m$ be a k-CNF formula with m clauses and n variables. Let $\bigcup_{i\in S} c_i$ be the set of literals contained in c_i's and let $\bigcap_{i\in S} c_i$ be the intersection of the sets of literals. Let $|c_i|$ be the number of literals contained in c_i (Note that this uses the same notation with $|A_i|$ but the types of sets are different). We first present three fundamental lemmas on A_i and c_i.

Lemma 3.2. *$|\bigcap_{i\in S} A_i| = 0$ holds iff there exists a variable x and $i, j \in S$ such that $x \in c_i$ and $\overline{x} \in c_j$.*

Proof. If $x \in c_i$ and $\overline{x} \in c_j$, $|A_i \bigcap A_j| = 0$. Therefore, $|\bigcap_{i\in S} A_i| = 0$. If $|\bigcap_{i\in S} A_i| = 0$, there exist $i, j \in S$ such that $|A_i \bigcap A_j| = 0$. Therefore, there exists x such that $x \in c_i$ and $\overline{x} \in c_j$. □

Lemma 3.3. *If every variable appear in $\{c_i\}_{i\in S}$ either as positive or negative literal, then $|\bigcap_{i\in S} A_i| = 2^{n-|\bigcup_{i\in S} c_i|}$.*

The following lemma holds by the using inclusion-exclusion formula for sets of clauses.

Lemma 3.4. *$|\bigcup_{i\in S} c_i|$ can be computed from $\{|\bigcap_{i\in T} c_i|\}_{T\subset S}$. Conversely, $|\bigcap_{i\in S} c_i|$ can be computed from $\{|\bigcup_{i\in T} c_i|\}_{T\subset S}$.*

By Lemmas 3.3 and 3.4, we can utilize $\{|\bigcap_{i\in S} c_i|\}$'s instead of $\{|\bigcap_{i\in S} A_i|\}$'s for obtaining the upper bound of λ_k. The key claim is the following:

Claim 3.5. *Let $\phi = c_1 \cdot c_2 \cdot \cdots \cdot c_m$ be a CNF formula with n variables. Then for any $T \subset [m]$ such that $|T| \ge \lfloor \log k \rfloor + 3$, $|\bigcap_{i\in T} c_i|$ is uniquely determined by the numbers $\{|\bigcap_{i\in S} c_i|\}$'s such that $|S| \le \lfloor \log k \rfloor + 2$.*

Proof. Let $\psi = c'_1 \cdot c'_2 \cdot \cdots \cdot c'_m$ be the other k-CNF formula, and suppose that for some $T \subset [m]$ such that $|T| = l+3$, ϕ and ψ satisfies the following equations: (i) $|\bigcap_{i\in S} c_i| = |\bigcap_{i\in S} c'_i|$ for any $S \subset T$ such that $|S| \le l+2$; and (ii) $|\bigcap_{i\in T} c_i| \ne |\bigcap_{i\in T} c'_i|$. Then what we should prove is that l must satisfy $l + 2 < \lfloor \log k \rfloor + 2$, i.e., $l < \lfloor \log k \rfloor$.

The situation can be further simplified as follows. We fix any $S_0 \subset T$ such that $|S_0| = l+2$. Then since an equality $|\bigcap_{i\in S} c_i| = |\bigcap_{i\in S} c'_i|$ holds for any $S \subset S_0$, we can identify $\{c_i\}_{i\in S_0}$ with $\{c'_i\}_{i\in S_0}$ w.l.g. by corresponding the variables between the sets of clauses. We denote $\{c_i\}_{i\in T} = \{c_i\}_{i\in S_0}\bigcup\{d\}$ and $\{c'_i\}_{i\in T} = \{c_i\}_{i\in S_0}\bigcup\{d'\}$. Let us also denote that $D_S = |(\bigcap_{i\in S} c_i)\bigcap d|$ and $D'_S = |(\bigcap_{i\in S} c_i)\bigcap d'|$ for $S \subset S_0$. Then (i) means that $D_S = D'_S$ for all $S \subset T$ such that $|S| \leq l+2$ and (ii) means that $D_T \neq D'_T$. The idea of the proof is to make use of the equality $D_S = D'_S$ in the descending order on $|S|$ $(= l+2, l+1, \cdots, 1)$ and reconstruct d and d'.

For $|S| = l+2$, i.e., for $S = S_0$, we can write by the assumption that $D_T = t$ and $D'_T = t'$ where $t - t' \geq 1$. Then c_i's $(i \in T)$ and d commonly contain a set of t (≥ 1) variables, say, X_1. Similarly, c_i's $(i \in T)$ and d' commonly contain a set of t' (≥ 0) variables, Y_1. Note that $X_1 \bigcap Y_1$ can be non-empty.

Next, for each $S \subset T$ such that $|S| = l+1$, we consider the equality $D_S = D'_S$. Since the t variables in X_1 are contained in c_i's and d, they contribute to the value D_S, which results in $D_S \geq t$. Similarly, t' variables in Y_1 contribute to D'_S for t'. Therefore, there are some gap for D_S and D'_S to coincide. To bridge this gap, d' must contain at least $t - t'$ variables which are not contained in d. Since there is such a set of variables for each S such that $|S| = l+1$, d' must contain at least $(t-t')\binom{l+2}{l+1} = (t-t')(l+2)$ variables which are not in d. We denote this set of variables by Y_2.

Thirdly, we consider the equality $D_S = D'_S$ for S such that $|S| = l$. Then the variables in Y_1 contribute to D'_S with the magnitude of t'. Furthermore, the variables in Y_2 contribute to D'_S with the magnitude of $(t-t')\binom{(l+2)-l}{(l+2)-(l+1)} = 2(t-t')$. Therefore, $D'_S \geq s + 2(t-t') = 2t - t'$. On the other hand, $D_S \geq t$. Since there are $\binom{l+2}{l}$ S's such that $|S| = l$, there must be at least $(2t - t' - t)\binom{l+2}{l} = (t-t')\binom{l+2}{l}$ variables in d which are not in d'. We denote this set of variables by X_2.

We reconstruct d and d' in this way. Finally, we see that when l is odd, d contains at least $(l+1)/2+1$ sets of variables $X_1, X_2, \cdots, X_{(l+1)/2+1}$, and d' contains at least $(l+1)/2+1$ sets of variables $Y_1, Y_2, \cdots, Y_{(l+1)/2+1}$. Here $|X_i| = (t-t')\binom{l+2}{l+2-2(i-1)} = (t-t')\binom{l+2}{l+4-2i}$ and $|Y_i| = (t-t')\binom{l+2}{l+2-2(i-1)+1} = (t-t')\binom{l+2}{l+5-2i}$ for $i \geq 2$. When l is even, d contains $l/2+1$ sets of variables $X_1, X_2, \cdots, X_{l/2+1}$ and d' contains $l/2+2$ sets of variables $Y_1, Y_2, \cdots, Y_{l/2+2}$, where $|X_i| = (t-t')\binom{l+2}{l+2-2(i-1)} = (t-t')\binom{l+2}{l+4-2i}$ and $|Y_i| = (t-t')\binom{l+2}{l+2-2(i-1)+1} = (t-t')\binom{l+2}{l+5-2i}$ for $i \geq 2$.

When l is odd, we evaluate the size of d. Then

$$\begin{aligned}
|d| &\geq \sum_{i=1}^{\frac{l+1}{2}+1} |X_i| = t + (t-t')\sum_{i=2}^{\frac{l+1}{2}+1}\binom{l+2}{l+4-2i} \\
&= t + (t-t')\sum_{i=2}^{\frac{l+1}{2}+1}\left\{\binom{l+1}{l+4-2i} + \binom{l+1}{l+4-2i-1}\right\} \\
&= t + (t-t')\sum_{j=0}^{l}\binom{l+1}{j} = t + (t-t')(2^{l+1}-1) \\
&\geq 2^{l+1} \text{ (since } t \geq 1 \text{ and } t-t' \geq 1). \qquad (2)
\end{aligned}$$

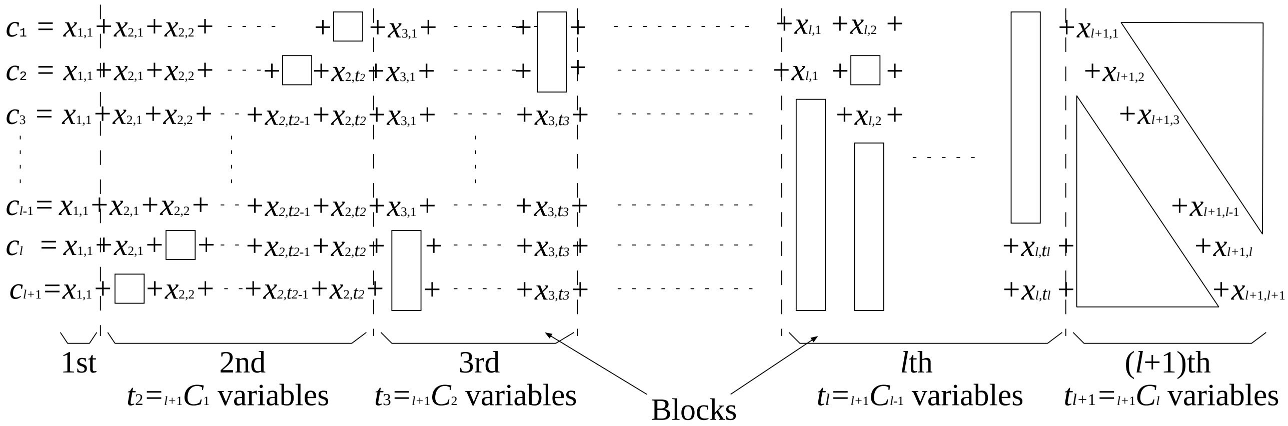

Figure 1: Definition of clauses $c_1,\ c_2,\ \cdots,\ c_l$, and c_{l+1}

Since the formula is in k-CNF, $2^{l+1} \leq |d| \leq k$, i.e., $l < \lfloor \log k \rfloor$. Thus Claim 3.5 is proven in this case. When l is even, we evaluate the size of d'. Then similarly to (2), we obtain

$$\begin{aligned}|d'| \quad &\geq \quad \sum_{i=1}^{\frac{l}{2}+2} |Y_i| = t' + (t-t') \sum_{i=2}^{\frac{l}{2}+2} \binom{l+2}{l+5-2i} = t' + (t-t')2^{l+1} \\ &\geq \quad 2^{l+1} \ \text{(since } t' \geq 0 \text{ and } t - t' \geq 1). \end{aligned} \tag{3}$$

$|d'| \leq k$ again leads to $l < \lfloor \log k \rfloor$. This completes the proof of Claim 3.5. □

3.2 The Lower Bounds

Recall that The lower bound for Theorem 3.1 is proven by showing that there exists a pair of k-CNF formulas for which the numbers of unsatisfying assignments for clause-sets of size up to $\lfloor \log k \rfloor + 1$ coincide, but those of size $\lfloor \log k \rfloor + 2$ differ. It can be done by constructing a pair of formulas which achieves the boundary value for Claim 3.5.

First, let us consider the case $k = 2^l$ for integer $l \geq 1$. We define $\log k + 1 = l + 1$ clauses $\{c_1, c_2, \cdots, c_{l+1}\}$ as shown in Fig. 1. We also define two more clauses as follows.

$$\begin{aligned} d &= \sum_{j:\text{odd}} \sum_{i=1}^{t_j} x_{j,i} = x_{1,1} + (x_{3,1} + x_{3,2} + \cdots + x_{3,t_3}) + (x_{5,1} + x_{5,2} + \cdots + x_{5,t_5}) + \cdots, \quad \text{and} \\ d' &= \sum_{j:\text{even}} \sum_{i=1}^{t_j} x_{j,i} = (x_{2,1} + x_{2,2} + \cdots + x_{2,t_2}) + (x_{4,1} + x_{4,2} + \cdots + x_{4,t_4}) + \cdots. \end{aligned}$$

When l is even, variables $\{x_{l,1}, x_{l,2}, \cdots, x_{l,t_l}\}$ and $\{x_{l+1,1}, x_{l+1,2}, \cdots, x_{l+1,t_{l+1}}\}$ are added at the end to d' and d, respectively. When l is odd, they are added to d and d', respectively.

Main properties of c_i's are as follows: (i) All the literals are positive; (ii) literals in c_i's are partitioned into $l+1$ blocks, where the ith block contains t_i variables such that each of them is added to $l-i+2$ clauses among the $l+1$ variables; and (iii) the blanks in the figure means that the variable in the column is not added to the clauses with them. For example, $x_{1,1}$ in the first block is added to all of the $l+1$ clauses, and each of the t_2 $(=l)$ variables $\{x_{2,1}, x_{2,2}, \cdots, x_{2,t_2}\}$ in the second block are added to l clauses. By the definition of t_i $t_i = \binom{l+1}{l-i+2}$ for $1 \le i \le l+1$. As for the sizes of c_i's, since each variable appears $l-i+2$ times in the ith block, each clause equally contains $t_i(l-i+2)/(l+1) = \binom{l+1}{l-i+2}(l-i+2)/(l+1) = \binom{l}{i-1}$ variables in the ith block. Therefore, $|c_i| = \sum_{i=1}^{l+1} \binom{l}{i-1} = 2^l = k$. The total number of variables contained in $\{c_1, c_2, \cdots, c_{l+1}\}$ is $\sum_{i=1}^{l+1} t_i = \sum_{i=1}^{l+1} \binom{l+1}{l-i+2} = 2^{l+1} - 1$.

As for the sizes of d and d', similarly to (2) and (3), we obtain $|d| = 2^l$ and $|d'| = 2^l - 1$ when l is even, and $|d| = 2^l - 1$ and $|d'| = 2^l$ when l is odd. From now on, we assume that l is even. (In the even case, a quite similar argument is possible.)

Before defining a pair of k-CNF formulas for the lower bound of λ_k, we define two subformulas:

$$\phi = c_1 \cdot c_2 \cdots c_{l+1} \cdot d \cdot (\overline{x}_{1,1} + y) \quad \text{and} \quad \psi = c_1 \cdot c_2 \cdots c_{l+1} \cdot (d' + y) \cdot (\overline{x}_{1,1} + \overline{y}).$$

Here the variable y and the last clauses of ϕ and ψ are introduced to make both formulas contain the same number of, i.e., 2^{l+1} variables without affecting the essential difference in the number of satisfying assingments of both formulas. (Furthermore, in order to make the last clauses have exactly k literals, we may fill them with any literals which have already appeared in the formulas.) Then the key lemma is the following.

Lemma 3.6. *For any $S \subset [l+1]$ such that $|S| < l+1$, $|(\bigcap_{i \in S} c_i) \bigcap d| = |(\bigcap_{i \in S} c_i) \bigcap (d'+y)| = 2^{l-|S|}$. For $S = [l+1]$, $|(\bigcap_{i=1}^{l+1} c_i) \bigcap d| = 1$ and $|(\bigcap_{i=1}^{l+1} c_i) \bigcap (d'+y)| = 0$ hold.*

Proof. We count the number of variables commonly contained in $\{c_i\}_{i \in S}$ and d (or $d'+y$) at each ith block. Then we can check that d has one common variable $x_{1,1}$ with $\{c_i\}_{i \in S}$ for any S, no common variable at the second block, $\binom{(l+1)-|S|}{2}$ common variables at the third block, and so forth. Therefore,

$$|(\bigcap_{i \in S} c_i) \bigcap d| = \sum_{j=0}^{\lfloor \frac{(l+1)-|S|}{2} \rfloor} \binom{(l+1)-|S|}{2j} = 2^{l-|S|}.$$

$|(\bigcap_{i \in S} c_i) \bigcap d'| = 2^{l-|S|}$ is proven in the similar way. Finally, it is easy to check that $|(\bigcap_{i=1}^{l+1} c_i) \bigcap d| = 1$ and $|(\bigcap_{i=1}^{l+1} c_i) \bigcap (d'+y)| = 0$. □

Let $A(\phi)$ and $A(\psi)$ be the numbers of unsatisfying assingments of ϕ and ψ, respectively. Then by the inclusion-exclusion formula (1) and Lemma 3.6 (also recall Lemmas 3.3 and Lemma 3.4 for the relation between $|\bigcap c_i|$ and $|\bigcap A_i|$), we obtain $A(\phi) < A(\psi)$.

Now we are ready to define two k-CNF formulas with n variables which yield the lower bound of λ_k. Let $\{\phi_i, \psi_i\}$ $(1 \le i \le n/2^{l+1})$ be $n/2^{l+1}$ copies of $\{\phi, \psi\}$ such that each of them has its own set of variables of size 2^{l+1}; namely, any two of the copies have disjoint sets of variables. The two k-CNF formulas with n variables are defined as follows.

$$\widetilde{\phi} = \phi_1 \cdot \phi_2 \cdot \cdots \cdot \phi_{n/2^{l+1}} \quad \text{and} \quad \widetilde{\psi} = \psi_1 \cdot \psi_2 \cdot \cdots \cdot \psi_{n/2^{l+1}}.$$

Since ϕ_i and ψ_i satisfy Lemma 3.6 for all i, $A(\widetilde{\phi}) < A(\widetilde{\psi})$. Therefore, $\lambda_k \ge \log k + 2$ for $k = 2^l$.

For a general k, suppose that k is written as $k = 2^l + j$ $(0 \leq j \leq 2^l - 1)$. Since λ_k is monotonically increasing, $\lambda_k = \lambda_{2^l+j} \geq \lambda_{2^l} \geq l + 2 = \lfloor \log k \rfloor + 2$. This completes the proof of Theorem 3.1. $\square$

Remark 3.7. (1) When $k = 2$, the 2CNF formulas which achieve this upper bound are given as follows. We define $\phi_1 = c_1 \cdot c_2 \cdot c_3 \cdot c_4$ such that $c_1 = x_1 + x_2$, $c_2 = x_1 + x_3$, $c_3 = x_1 + x_4$, and $c_4 = \overline{x}_2 + \overline{x}_4$. We also define $\phi_2 = d_1 \cdot d_2 \cdot d_3 \cdot d_4$ such that $d_1 = x_1 + x_2$, $d_2 = x_1 + x_3$, $d_3 = x_2 + x_3$, and $d_4 = \overline{x}_2 + \overline{x}_4$. Note that c_4 and d_4 are added to make the numbers of variables coincide (without making difference between the information on the number of satisfying assingments for both formulas). We can easily check that the numbers of unsatisfying assignments for each of the two clauses are all the same. However, $|c_1 \bigcap c_2 \bigcap c_3| = 1$ and $|d_1 \bigcap d_2 \bigcap d_3| = 0$. Actually, the numbers of unsatisfying assignments of ϕ_1 and ϕ_2 are different, i.e., 10 and 11.

(2) When $k = n/2$, $\lambda_n \geq \lambda_k \geq \lfloor \log k \rfloor + 2 = \lfloor \log n \rfloor + 1$. Since $\lambda_n \leq \log n + 1$ by Theorem 2.2, we obtain as a special case of Theorem 3.1 that $\lambda_n = \lfloor \log n \rfloor + 1$.

4 Dual Representations for Hypergraphs

As is well known, there is a close relation between 2CNF formulas ϕ and graphs G. Namely, suppose that there is an edge between v_i and v_j in G. We associate this edge with a clause $x_i + x_j$ in ϕ. Then a satisfying assignment of ϕ corresponds to a vertex cover of G by inserting v_i into the cover iff $x_i = 1$ in the assignment. Our result for 2CNF formulas (Remark 3.7) means that if we are given all the values (0 or 1) $|\bigcap_{i \in S} e_i|$ (0 or 1) for all S such that $|S| \leq 3$, then we can determine the number of different vertex covers uniquely. Here we denote $|e_i \bigcap e_j| = 1$ ($= 0$, resp.) if two edges share a (no, resp.) vertex.

Actually, however, all the values of $|\bigcap_{i \in S} e_i|$ contain more information; namely, one can easily see that we can reconstruct the original graph itself rather than the number of vertex covers. (For example, suppose that $|e_1 \bigcap e_2| = |e_2 \bigcap e_3| = |e_3 \bigcap e_1| = 1$. Then, e_1, e_2, and e_3 construct a triangle if $|e_1 \bigcap e_2 \bigcap e_3| = 0$ and a star if $|e_1 \bigcap e_2 \bigcap e_3| = 1$.) Thus a graph has dual representations; one is the vertex-based conventional representation and the other is the edge-based representation as described above.

This argument is extended to hypergraphs. Let a k-graph ($k \geq 2$) be a hypergraph such that each hyperedge consists of exactly k vertices. For any $k \geq 2$, we can consider the same correspondence as above between k-CNF formulas and k-graphs. Then the obvious question is until which k the above reconstructability of the original graph is possible. Interestingly, it is valid for any integer $k \geq 2$ even if the number of different vertex covers has apparently less amount of information than the k-graph itself. The reason is simple as follows. Given $|\bigcap_{i \in S} e_i|$ over all S such that $|S| \leq \lfloor \log k \rfloor + 2$, then Theorem 3.1 implies that the numbers $|\bigcap_{i \in S} e_i|$ over all subsets $S \subset [m]$ (m is the number of hyperedges) are determined automatically. Then the k-graph can be reconstructed using these numbers $|\bigcap_{i \in S} e_i|$ $(1 \leq |S| \leq m)$ and is determined uniquely up to isomorphism. This observation means that k-graphs have relatively efficient dual representations, which might have some interesting applications.

5 Concluding Remarks

We considered a condition for uniquely determining the number of satisfying assignments of k-CNF formulas. We have shown that for a k-CNF formula, the information for clause-sets of size up to $\lfloor \log k \rfloor + 2$ is necessary and sufficient. This generalizes the previous result for general CNF formulas. We also mentioned the relation of our results for k-CNF formulas with dual representations for hypergraphs. Our proof of the upper bounds for k-CNF formulas is still existential, Therefore, to find an explicit procedure for obtaining the information of clause-sets of size more than $\lfloor \log k \rfloor + 2$ from smaller parts should be future work.

Acknowledgments

The second author was supported in part by the grant of JSPS Research Fellowships for Young Scientists (00133161).

References

[1] K. Iwama, CNF satisfiability test by counting and polynomial average time, SIAM J. Comput., vol. 18, no. 2 (1989) 385-391.

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

[3] N. Linial and N. Nisan, Approximate inclusion-exclusion, Combinatorica 10 (1990) 349-365.

[4] E. L. Lozinskii, Computing propositional models, Inform. Process. Lett., vol. 41 (1992) 327-332.

[5] A. A. Melkman and S. E. Shimony, A note on approximate inclusion-exclusion, Discrete Appl. Math., 73 (1997) 23-26.

[6] L. G. Valiant, The complexity of computing the permanent, Theor. Comput. Sci., 8 (1979) 189-201.