A-25

k-CNF 式に対する Inclusion-Exclusion 公式における非充足解数計算手法

Computing the Number of Unsatisfying Assignments for k-CNF Formulas Via the Inclusion-Exclusion Formula

松浦 昭洋*
Akihiro Matsuura

1 はじめに

k-CNF 式の非充足解の総数を求める際、Inclusion-Exclusion 公式を用いるのは標準的な方法である。公式中、全ての項の組合せ、すなわち項数を m としたとき、2^m 個の項集合に対して、それらの項集合を非充足にする変数割り当ての数を評価する必要があるが、最近 [2] において、$\lfloor \log k \rfloor + 2$ 個までの項の非充足解の情報が分かれば、元の式の非充足解の数は既に一意に定まっているという事実が示された。しかし、その論文中で非充足解の数がある数に一意に決まっていることが "存在定理" として示されたものの、$\lfloor \log k \rfloor + 2$ 個までの項に関する非充足解の数の情報から、それ以上の項数に関する非充足解の数を実際に計算する手法については分かっていなかった。本稿では、一般の CNF 式に関する結果を拡張することにより、そのようなアルゴリズムが存在することを示す。

2 準備

(k-)CNF 式の充足解の総数を求める問題は、NP 完全問題とも深い関連があり、活発に研究が行われている。1970 年代後半に Valiant によってその $\#P$ 完全性が証明されており、実用時間内で厳密に解くことは困難であると推測されている。一方、その数を求めるためのナイーブな手法としては、以下に説明する Inclusion-Exclusion 公式が知られている ([1]-[4])。

今、$\phi = c_1 \cdot c_2 \cdot \cdots \cdot c_m$ を n 変数の (k-)CNF 式とし、A_i $(i = 1, 2, \cdots, m)$ を項 c_i を非充足にするような変数割り当てとする。(例えば、3 変数 2CNF 式 $(x_1 + \bar{x}_2)(x_1 + x_3)$ においては、第一項を非充足にする変数割り当ての集合は $A_1 = \{(0,1,0),(0,1,1)\}$ となる) このとき、ϕ の非充足解の総数は以下の Inclusion-Exclusion 公式から求められる。

$$\begin{aligned} \Big|\bigcup_{i=1}^{m} A_i\Big| &= \sum_{i=1}^{m}|A_i| - \sum_{i<j}|A_i \bigcap A_j| \\ &+ \sum_{i<j<k}|A_i \bigcap A_j \bigcap A_k| \\ &- \cdots \\ &+ (-1)^{m-1}\Big|\bigcap_{i=1}^{m} A_i\Big|. \end{aligned} \tag{1}$$

ここで、$|A_i|$ は項 c_i を非充足にするような変数割り当ての総数を表す。式 (1) 中には計算項が 2^m 個存在するため、その直接的な計算は指数爆発を起こしてしまうが、未だ効率的な計算法は知られていない。

*京都大学大学院情報学研究科

従来研究

アルゴリズムの提案ではないものの、1996 年 Kahn 等 [3] により、Inclusion-Exclusion 公式で計算される CNF 式の (非) 充足解の総数は、全ての S に関して非充足解の数の情報 $\{|\bigcap_{i\in S} A_i|\}$ が分からなくても、$|S| \leq \log n + 1$ であるような S に関する非充足解の数の情報から一意に定まる、という興味深い結果が導かれた。彼等の証明は充足解の総数の一意性 (つまり存在定理) に対するものだったが、後に Melkman 等 [5] によって、$|S| \leq \log n + 1$ である全ての $S \subset \{1, 2, \cdots, m\}$ に対する $|\bigcap_{i\in S} A_i|$ の値の情報から $|S| \geq \log n + 2$ に対する $|\bigcap_{i\in S} A_i|$ の値を実際に求める方法が示された

さらに最近、k-CNF 式 (各項がちょうど k 個のリテラルを持つ) に対して、ちょうど $\lfloor \log k \rfloor + 2$ 個の項の和集合に関して非充足解の数を知ることが、元の式の充足解の数を一意に定めるための必要十分条件であることが [2] において示された。これにより、例えば 3CNF 式において充足解の総数は、高々 $|S| \leq 3$ であるような $|\bigcap_{i\in S} A_i|$ の値から一意に定まっていることが分かる。しかし、[2] における証明も充足解の総数の一意性を保証するにとどまり、$\lfloor \log k \rfloor + 3 \leq |S| \leq m$ に対する $|\bigcap_{i\in S} A_i|$ を実際に求める計算手法は分かっていなかった。

CNF 式に対する従来の結果 [5] を利用するのは自然なことと思われるので、まずはその概略を述べる。$\{c_i\}_{i=1}^{l}$ と $\{d_i\}_{i=1}^{l}$ をそれぞれ項集合とする。$|\bigcap_{i\in S} c_i|$ を $\{c_i\}_{i\in S}$ に含まれる共通変数の数とし、$|\bigcup_{i\in S} c_i|$ を $\{c_i\}_{i\in S}$ に含まれる変数の和集合のサイズとする。また、$\{c_i\}_{i=1}^{l}$ と $\{d_i\}_{i=1}^{l}$ は次の条件「$|S| \leq l-1$ である全ての部分集合 S に対して $|\bigcap_{i\in S} c_i| = |\bigcap_{i\in S} d_i|$(*)」を満たすとする。このとき、[5] において次の結果が得られた。

$$\max\left\{\Big|\,|\bigcup_{i=1}^{l} c_i| - |\bigcup_{i=1}^{l} d_i|\,\Big|\right\} = \left\lfloor \frac{N}{2^{l-1}} \right\rfloor. \tag{2}$$

ここで、$N = \max\{|\bigcup_{i=1}^{l} c_i|, |\bigcup_{i=1}^{l} d_i|\}$。一般の CNF 式においては $N \leq n$ であるので、考えられる $\{|\bigcup_{i=1}^{l} c_i|\}$ の値が一意であるための条件は式 (2) より $\lfloor n/2^{l-1} \rfloor < 1$、すなわち $l > \log n + 1$ ならば良い。しかし k-CNF 式では $N \leq kl$ より $\lfloor kl/2^{l-1} \rfloor < 1$ しか言えず、タイトな結果 ($l > \lfloor \log k \rfloor + 2$) が求められない。

今回の結果

そこで我々は $\{|\bigcup_{i=1}^{l} c_i|\}$ の代りに $\{|\bigcap_{i=1}^{l} c_i|\}$ について考えることで、次の結果を得た。

定理. For $l \geq 2$,

$$\max\left\{\left|\,|\bigcap_{i=1}^{l} c_i| - |\bigcap_{i=1}^{l} d_i|\,\right|\right\} = \left\lfloor \frac{k}{2^{l-2}} \right\rfloor. \qquad (3)$$

ここで、maxは上の条件(*)を満たす全ての$\{c_i\}_{i=1}^{l}$と$\{d_i\}_{i=1}^{l}$に関する最大値を取る。

定理において、特に$l = \lfloor \log k \rfloor + 3$とすると(2)は$\lfloor k/2k \rfloor = 0$となる。したがって、$l = \lfloor \log k \rfloor + 3$個までの項の和集合のサイズが分かれば、それ以上の項数に関する和集合のサイズは一意に定まる。また、項の和集合のサイズ$\{|\bigcap_{i\in S} c_i|\}$から非充足解の数$\{|\bigcap_{i\in S} A_i|\}$が求められることは[2][3]等において既知であるため、Inclusion-Exclusion公式より非充足解の総数も一意に定まる。

3 定理の証明

基本となるアルゴリズムは一般のCNF式に対するもの[5]と同様である(pseudo-codeは右上図1)。解析における違いは、各項の大きさ$|c_i|$が定数kに制限されるため、各ステップにおける$|\bigcap_{i\in S} c_i|$、$|\bigcup_{i\in S} c_i|$の値の変化が異なることである。今Nを$|\bigcap_{i\in S}^{l} c_i|$ ($|S| \leq l-1$)の値が既知であるときの$|\bigcup_{i=1}^{l} c_i|$の最大値とする。以下、$|c_l| = k$の大きさで場合分けして$|\bigcap_{i=1}^{l} c_i|$の取りうる値の上下限を求めていく。

(i) $|c_l| = k \leq N/2$のとき、$c_{i,1} = c_i \bigcap c_l$, $i = 1, \cdots, l-1$なる$l-1$個の項の集合$\{c_{i,1}\}_{i=1}^{l-1}$を考える。この集合に関して

$$|\bigcap_{i=1}^{l} c_i| = |\bigcap_{i=1}^{l-1} c_{i,1}| \qquad (4)$$

が成り立つので、以下$\{c_{i,1}\}_{i=1}^{l-1}$の共通集合のサイズを求めればよい。なお、$\bigcup_{i=1}^{l-1} c_{i,1} \subset c_l$より、$|\bigcup_{i=1}^{l-1} c_{i,1}| \leq |c_l| = k$を満たす。

(ii) $N/2 < |c_l| = k \leq N$のとき、$c_{i,1} = c_i - c_l$, $i = 1, \cdots, l-1$なる$l-1$個の項の集合を考えると

$$|\bigcap_{i=1}^{l} c_i| = |\bigcap_{i=1}^{l-1} c_i| - |\bigcap_{i=1}^{l-1} c_{i,1}| \qquad (5)$$

が成り立つので、$\{c_{i,1}\}$の共通集合のサイズを求めた後、$|\bigcap_{i=1}^{l-1} c_i|$から減じることで、所望の$|\bigcap_{i=1}^{l} c_i|$が求まる。なお、$\{c_{i,1}\}_{i=1}^{l-1}$の和集合に関して、$|\bigcup_{i=1}^{l-1} c_{i,1}| < N - |c_l| \leq N/2 < k$が成り立つ。

(i)(ii)いずれの場合も、次のステップでは$|\bigcup_{i=1}^{l-1} c_{i,1}| \leq k$を満たす項集合$\{c_{i,1}\}_{i=1}^{l-1}$に対して、(i)(ii)同様の場合分けをして、再帰的な計算を行う。例えば上記(i)において、もし$|c_{l-1,1}| \leq \lfloor k/2 \rfloor$ならば、$c_{i,2} = c_{i,1} \bigcap c_{l-1,1}$とおき、$\{c_{i,2}\}_{i=1}^{l-2}$の共通集合を考えていく。なお、$\bigcup_{i=1}^{l-2} c_{i,2} \subset c_{l-1,1}$より、$|\bigcup_{i=1}^{l-2} c_{i,2}| \leq |c_{l-1,1}| \leq \lfloor k/2 \rfloor$を満たしている。一方で$\lfloor k/2 \rfloor < |c_{l-1,1}| \ (\leq k)$ならば$c_{i,2} = c_{i,1} - c_{l-1,1}$とおき、$\{c_{i,2}\}_{i=1}^{l-2}$の共通集合を考えていく。この場合も、$|\bigcup_{i=1}^{l-2} c_{i,2}| < k - |c_{l-1}| < \lfloor k/2 \rfloor$となり、次の再帰計算は和集合のサイズが高々$\lfloor k/2 \rfloor$であるような項集合$\{c_{i,2}\}_{i=1}^{l-2}$について、$c_{l-2,2}$のサイズで場合分けを行い、共通集合のサイズを求めていく。

procedure *bounds*$(\{c_i\}_{i=1}^{l}; N)$: (lower_bd, upper_bd);
  if $l = 2$, then return
    lower_bd = max $\{|c_1| + |c_2| - \lfloor N \rfloor, 0\}$;
    upper_bd = min $\{|c_1|, |c_2|\}$;
  else if $|c_l| \leq N/2$, for all $i \in \{1, \cdots, l-1\}$,
    $c_i = c_i \bigcap c_l$;
    return *bounds*$(\{c_i\}_{i=1}^{l-1}; |c_l|)$;
  else ($N/2 < |c_l| \leq N$), for all $i \in \{1, \cdots, l-1\}$,
    $c_i = c_i - c_l$;
    (lower_bd, upper_bd):= *bounds*$(\{c_i\}_{i=1}^{l-1}; N - |c_l|)$;
    return $|\bigcup_{i=1}^{l-1} c_i|$−upper_bd, $|\bigcup_{i=1}^{l-1}|$−lower_bd;

図1. $|\bigcap_{i=1}^{l} c_i|$の上下限を求めるアルゴリズム

このように再帰的な計算を行っていくと、$\{c_{i,1}\}_{i=1}^{l-1}$, $\{c_{i,2}\}_{i=1}^{l-2}$, $\cdots$, $\{c_{i,l-2}\}_{i=1}^{2}$なる項集合の列が生成され、最終的に$\{c_{i,l-2}\}_{i=1}^{2}$に関して$c_{1,l-2} \bigcap c_{1,l-2}$のサイズの情報が分かれば、式(4)(5)より$|c_{1,l-2} \bigcap c_{1,l-2}|$の値の上下限の差はそのまま$|\bigcap_{i=1}^{l} c_i|$の値の上下限の差となる。まず$|c_{1,l-2} \bigcup c_{2,l-2}| \leq \lfloor k/2^{l-3} \rfloor$となることが、上の(i)(ii)と同様の解析で求められる。さらに、$|c_{1,l-2} \bigcap c_{2,l-2}|$の上下限については、下限が$\max\{|c_{1,l-2}| + |c_{2,l-2}| - \lfloor k/2^{l-3} \rfloor, 0\}$、上限が$\min\{|c_{1,l-2}|, |c_{2,l-2}|\}$であることから、それらの差は高々$\lfloor k/2^{l-2} \rfloor$であることが分かる。以上より、$|\bigcap_{i=1}^{l} c_i|$の上下限の差は高々$\lfloor k/2^{l-2} \rfloor$であり、図1がそのような上下限を求める手続きであることが示された。一方で、$|\bigcap_{i=1}^{l} c_i|$の上下限の差がこれ以上改善されないことが、[2]で示された二つの特別なk-CNF式の存在から容易に示される(ここでは紙面の都合により割愛する)。以上より定理が証明された。

4 まとめ

本稿では、k-CNF式において非充足解の総数を求めるため、あるサイズまでの項集合に対する充足解の数の情報を用いて、それ以上のサイズの項集合に対する非充足解の数の上下限を求める手法について述べた。これにより、特に、$\lfloor \log k \rfloor + 2$個までの項に関する非充足解の数の情報が分かれば、そのCNF式の非充足解の総数を計算することが可能となる。

謝辞

本研究は日本学術振興会特別研究員奨励費(13003161)による補助を受けています。

参考文献

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

[2] K. Iwama and A. Matsuura, Inclusion-exclusion for k-CNF formulas, *Proc. SAT2002*, pp. 182-189, 2002.

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

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

[5] A. A. Melkman and S. E. Shimony, A note on approximate inclusion-exclusion, *Disc. Appl. Math.*, 73, pp. 23-26, 1997.

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