On Berge Multiplication for Monotone Boolean Dualization *

Endre Boros[1], Khaled Elbassioni[2], and Kazuhisa Makino[3]

[1] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003;
boros@rutcor.rutgers.edu
[2] Max-Planck-Institut für Informatik, 66111 Saarbrücken, Germany;
elbassio@mpi-sb.mpg.de
[3] Department of Mathematical Informatics, University of Tokyo, Tokyo, 113-8656, Japan;
makino@mist.i.u-tokyo.ac.jp

Abstract. Given the prime CNF representation ϕ of a monotone Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$, the dualization problem calls for finding the corresponding prime DNF representation ψ of f. A very simple method (called *Berge multiplication* [3, Page 52–53]) works by multiplying out the clauses of ϕ from left to right in some order, simplifying whenever possible using *the absorption law*. We show that for any monotone CNF ϕ, Berge multiplication can be done in subexponential time, and for many interesting subclasses of monotone CNF's such as CNF's with bounded size, bounded degree, bounded intersection, bounded conformality, and read-once formula, it can be done in polynomial or quasi-polynomial time.

1 Introduction

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. A function is called *monotone* (also called *positive*) if for every pair of vectors $x, y \in \{0,1\}^n$, $x \leq y$ (i.e., $x_i \leq y_i$ for all i) always implies $f(x) \leq f(y)$. Any monotone function f has a unique *prime conjunctive normal form* (*CNF*) expression

$$\phi(x) = \bigwedge_{C \in \mathcal{C}} \Big(\bigvee_{i \in C} x_i \Big), \tag{1}$$

where $\mathcal{C}$ is *Sperner* (i.e., $I \not\subseteq J$ holds for $I, J \in \mathcal{F}$ with $I \neq J$). It is well-known that $\mathcal{C}$ corresponds to the set of all *prime implicates* of f. The well-known *monotone Boolean dualization problem* is to find the corresponding *prime disjunctive normal form* (DNF) representation of f:

$$\psi(x) = \bigvee_{D \in \mathcal{D}} \Big(\bigwedge_{i \in D} x_i \Big), \tag{2}$$

* The first author is thankful for the partial support by NSF (CBET-0735910). The second and third authors thank the partial support by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science.

where $\mathcal{D}$ is Sperner and corresponds to the set of all *prime implicants* of f. Equivalently, the problem is to compute, for an explicitly given hypergraph $\mathcal{C} \subseteq 2^V$, the transversal hypergraph $\mathcal{D}$, consisting of all minimal transversals D of $\mathcal{H}$ (i.e., all minimal subsets $D \subseteq V$ such that $D \cap C \neq \emptyset$ for all $C \in \mathcal{C}$). This problem has received considerable attention in the literature (see e.g., [4, 11, 13, 29, 32]), since it is known to be polynomially or quasi-polynomially equivalent with many problems in various areas, such as artificial intelligence (e.g., [11, 22]), database theory (e.g., [31]), distributed systems (e.g., [18, 20]), machine learning and data mining (e.g., [1, 7, 19]), mathematical programming (e.g., [5, 24]), matroid theory (e.g., [27, 25]), and reliability theory (e.g., [8, 33]).

While the size of the output DNF ψ can be exponential in the size of ϕ, it is open (for more than 25 years now, e.g., [4, 12, 21, 28, 29, 32]) whether ψ can be computed in *output-polynomial* (or *polynomial total*) *time*, i.e., in time polynomial in the combined size of ϕ *and* ψ. Any such algorithm for the monotone dualization problem would significantly advance the state of the art of the problems in the application areas mentioned above. This is witnessed by the fact that these problems are cited in a rapidly growing body of literature and have been referenced in various survey papers and complexity theory retrospectives, e.g. [12, 14, 21, 29, 30, 32].

In 1996, Fredman and Khachiyan [17] established a remarkable result that the monotone dualization problem can be solved in quasi-polynomial time $O(nN) + N^{o(\log N)}$, where $N = |\phi| + |\psi|$, thus putting the problem somewhere between polynomiality and NP-completeness. They achieved this by presenting a quasi-polynomial time algorithm for the decision-version of the problem: given two monotone Boolean formulae ϕ and ψ in CNF and DNF respectively, is $\phi \equiv \psi$? Furthermore, for several special classes of monotone formulae ϕ, the problem is known to be solvable in polynomial time, e.g., when every clause has bounded-size [6, 11], when every variable has bounded degree [9, 13, 32], when clauses have bounded intersection-size [26], for read-once formulae [15], etc.

A very simple method to solve the monotone dualization problem, called *left-to-right multiplication*, or sometimes *Berge multiplication* (see [3, Page 52–53]), works by traversing the clauses of the input CNF in some order, say $j = 1, \ldots, m = |\phi|$, multiplying out clause C_j with the DNF obtained for $C_1 \wedge \ldots \wedge C_{j-1}$, and simplifying the DNF's using *the absorption law* (i.e., the identity $x \vee (x \wedge y) = x$ for all Boolean x, y) whenever possible (see Figure 1). We remark that many practical algorithms for monotone dualization are obtained from the left-to-right multiplication by putting several heuristic ideas (see e.g., [2, 10, 23, 35]).

It is not difficult to come up with examples for which this method exhibits an *exponential blow-up* in the input-output size, e.g., the intermediate DNFs are exponential in the input size, while the final output is polynomially-bounded. Consider for instance, a CNF $\phi = \bigwedge_{1 \leq i,j \leq n}(x_i \vee y_j)$ on the set of $2n$ variables $\{x_1, \ldots, x_n, y_1, \ldots, y_n\}$. One can easily check that the corresponding prime DNF is $(x_1 \wedge \ldots \wedge x_n) \vee (y_1 \wedge \ldots \wedge y_n)$. On the other hand, if we start by multiplying the clauses $(x_1 \vee y_1), \ldots, (x_n \vee y_n)$, then we get 2^n clauses, which will be canceled out later in the process. More interestingly, Takata [34] gave an example for which the left-to-right multiplication method exhibits a *superpolynomial* blow-up, under *any ordering* of the clauses of the input CNF. He also suggested a generalization of the multiplication method which was shown to be output quasi-polynomial in [16].

In view of this result, it is natural to ask whether there is an example where an *exponential* blow-up is unavoidable under any ordering of the clauses. In this paper, we answer this question in the *negative*. Namely, we show that, for any monotone CNF, there is an ordering of the clauses such that the size of the intermediate DNF at any stage of the left-to-right multiplication is bounded by a *subexponential* in the input-output size. Furthermore, we show that, for several interesting well-known classes of monotone CNF formulae such as read-once, bounded degree, bounded clause-size, etc., there are orderings of the clauses that guarantee (quasi-)polynomial blow-up's. The only result we are aware of this type is the one for bounded degree formulae [9].

To formally state our results, let us consider a monotone CNF $\phi = C_1 \wedge \cdots \wedge C_m$, and let $\pi \in \mathbb{S}_m$ be a permutation of the clauses, where $\mathbb{S}_m$ denotes the set of all permutations of m elements. For $j = 1, \ldots, m$, let ϕ_j^π denote the CNF having the first j clauses in ϕ according to the ordering π, i.e., $\phi_j^\pi \stackrel{\text{def}}{=} \bigwedge_{l=1}^{j} C_{\pi(l)}$. For a CNF (resp., DNF) φ, we denote by $|\varphi|$ the number of clauses (resp., terms). Denote by $\nu(\pi)$ the size of a maximum intermediate DNF produced during the left-right multiplication, i.e.,

$$\nu(\pi) \stackrel{\text{def}}{=} \max_{1 \le j \le m} |(\phi_j^\pi)^*|,$$

where, for a monotone CNF φ, φ^* denotes the prime DNF corresponding to φ. Then we have the following theorem.

Theorem 1. *Let ϕ be a prime monotone CNF with n variables and m clauses. Then*

(i) *If ϕ has bounded clause-size, bounded degree, bounded intersection-size, or bounded degeneracy, then there exists a permutation π of the clauses in ϕ such that $\nu(\pi) = |\phi^*|^{O(1)}$.*
(ii) *If ϕ has bounded conformality or read-once representation, then there exists a permutation π of the clauses in ϕ such that $\nu(\pi) = |\phi^*|^{O(\log m)}$.*
(iii) *For any prime monotone CNF ϕ, there exists a permutation π of the clauses in ϕ such that $\nu(\pi) \le n^{\sqrt{n}+1}|\phi^*|^{\sqrt{n}\ln m}$.*

Furthermore, such permutations can be found in polynomial time in n and m.

The formal definitions of the types of CNF's stated in (i) and (ii) will be given in Sections 3 and 4. We remark that there is a prime monotone CNF ϕ with read-once representation such that $\nu(\pi) = |\phi^*|^{\Omega(\log\log m)}$ holds for any permutation π of clauses in ϕ [34].

It is easy to see that, for a given permutation π, the left-to-right multiplication takes polynomial time in n, m, and $\nu(\pi)$, where more careful analysis can be found in Section 2. Thus, the theorem above gives an upper bound on the running time of the left-to-right multiplication procedure.

Corollary 1. *The following three statements hold.*

(i) *If ϕ is a prime monotone CNF that has bounded clause-size, bounded degree, bounded intersection-size, or bounded degeneracy, then the left-to-right multiplication for ϕ can be done in output-polynomial time.*

(ii) *If ϕ is a prime monotone CNF that has bounded conformality or read-once representation, then the left-to-right multiplication for ϕ can be done in output-quasi-polynomial time.*
(iii) *For any prime monotone CNF, the left-to-right multiplication can be done in output-subexponential time.*

The rest of the paper is organized as follows. In the next section, we state our notation and present several properties of left-to-right multiplication used in the following sections. In Section 3, we show that the left-to-right multiplication based on reverse lexicographic ordering of clauses is an efficient way of dualizing monotone CNF's with bounded clause-size, bounded degree, or bounded clause-intersections. In Section 4, we present a more general technique for ordering the clauses of an input CNF, and derive from it the above stated results for general monotone CNF's and for some special classes.

2 Preliminaries

Let $\phi = \phi(x_1, \dots, x_n)$ be a formula. We denote by $V(\phi)$ the set of variables in ϕ. For convenience, if ϕ is a monotone CNF (resp. DNF) and C is a clause (resp., term) in ϕ, we shall write $C \in \phi$, and view C also as the index set $C \subseteq V(\phi)$ of the variables that it contains. This way, one can also view ϕ as a subfamily of $2^{V(\phi)}$, each of which represents a clause (resp., term), and thus use ordinary set operations on it. A monotone CNF ϕ is *prime* if for all $C, C' \in \phi$, $C \subseteq C'$ implies that $C = C'$ (see (1)). If ϕ is a monotone CNF formula, we denote by ϕ^* a prime DNF formula representing the same monotone Boolean function as ϕ (see (2)). As mentioned in the Introduction, any monotone function has a unique prime CNF (DNF) expression. In this paper we consider the following problem:

Problem MONOTONE BOOLEAN DUALIZATION
Input: The prime CNF ϕ of a monotone Boolean function.
Output: The prime DNF ϕ^*.

We shall assume that a given monotone CNF ϕ satisfies $n = |V(\phi)|$ and $m = |\phi|$.

The left-to-right multiplication given in Figure 1 is one of the simplest procedure to solve the MONOTONE BOOLEAN DUALIZATION. Here function Min($\cdot$) takes the conjunction of a monotone prime DNF ρ and a monotone clause C, and returns a prime monotone DNF ρ' that is equivalent to $\rho \wedge C$.
It is not difficult to see that for all $j = 1, \dots, m$, ψ_j in Figure 1 satisfies $\psi_j = (\phi_j^\pi)^*$, and hence the left-to-right multiplication correctly computes ϕ^* $(= \psi_m)$. Let us then consider its time complexity.

Proposition 1. *For a prime monotone CNF ϕ and a permutation $\pi \in \mathbb{S}_m$, Procedure* LR-Mult(ϕ, π) *can be implemented to run in $O(nm\nu(\pi)\min\{m, \nu(\pi)\})$ time.*

Procedure LR-Mult(ϕ, π)**:**
Input: The prime CNF $\phi = \wedge_{j=1}^{m} C_j$ of a monotone Boolean function and a permutation $\pi \in \mathbb{S}_m$.
Output: The prime DNF ϕ^*.

$\psi_0 := \emptyset$
for $j = 1, \ldots, m$
  $\psi_j := \text{Min}(\psi_{j-1} \wedge C_{\pi(j)})$
return ψ_m and **halt**

Fig. 1. The left-to-right multiplication

For a monotone CNF ϕ and $i \in V(\phi)$, we denote by $\phi_{(i)}$ the subformula of ϕ consisting of all clauses containing variable x_i, and let $\deg_\phi(i) = |\phi_{(i)}|$ be the degree of x_i in ϕ. For a subset $S \subseteq V(\phi)$ of variables, denote by ϕ_S the CNF formula obtained form ϕ by fixing $x_i = 1$ for all $i \in V(\phi) \setminus S$. Equivalently, $\phi_S = \bigwedge_{C \in \phi:\, C \subseteq S} \big(\bigvee_{i \in C} x_i\big)$. Thus we call ϕ_S the *projection* of ϕ on S. The reason that we are interested in projections is the following.

Proposition 2 ([28]). *Let ϕ be a monotone CNF. For any $S \subseteq V(\phi)$, we have $|\phi_S^*| \leq |\phi^*|$.*

Clearly, we have $|(\phi \wedge \phi')^*| \leq |\phi^*||(\phi')^*|$ for any CNF's ϕ and ϕ', and thus the above proposition implies the following claims.

Lemma 1. *Let ϕ ba a monotone CNF. If $\phi' = \phi_{S_1} \wedge \phi_{S_2} \wedge \cdots \wedge \phi_{S_k}$ for some subsets $S_\ell \subseteq V(\phi)$, $\ell = 1, .., k$, then we have $|(\phi')^*| \leq |\phi^*|^k$.*

Lemma 2. *Let $\phi = \bigwedge_{j=1}^{m} C_j$ be a monotone CNF, and let $\pi \in \mathbb{S}_m$ be a permutation of the clauses of ϕ such that for every $j = 1, \ldots, m$ there exists some subsets $S_{j,\ell} \subseteq V$, $\ell = 1, \ldots, k_j$ such that*

$$\phi_j^\pi = \phi_{S_{j,1}} \wedge \phi_{S_{j,2}} \wedge \cdots \wedge \phi_{S_{j,k_j}} \tag{3}$$

holds. Let $k = \max\{k_1, \ldots, k_m\}$. Then we have $\nu(\pi) \leq |\phi^*|^k$, and thus LR-Mult(ϕ, π) *computes ϕ^* in $O(nm|\phi^*|^k \min\{m, |\phi^*|^k\})$ time.*

In the following sections we show various techniques to find such an ordering π of ϕ which guarantees a *small* k in the above statement.

3 Reverse Lexicographic Orderings

Assume that $V = V(\phi)\,(= \{1, 2, \ldots, n\})$ and for subsets $A, B \subseteq V$ let us denote by $L = L(A, B)$ their *last common elements*, i.e., L is the maximal subset $L \subseteq A \cap B$ such that for all $i_1 \in (A \cup B) \setminus L$ and $i_2 \in L$ we have $i_1 < i_2$. We say that A *precedes*

B if $\max(A \setminus L(A,B)) < \max(B \setminus L(A,B))$. For example, if $A = \{1,3,5,6\}$ and $B = \{4,5,6\}$, then $L(A,B) = \{5,6\}$, $\max(A \setminus L(A,B)) = 3$, $\max(B \setminus L(A,B)) = 4$, and A precedes B. On the other hand, if $A = \{1,3,5\}$ and $B = \{1,5,6\}$, then $L(A,B) = \emptyset$, $\max(A \setminus L(A,B)) = 5$, $\max(B \setminus L(A,B)) = 6$, and A precedes B. Finally, we say that $\{C_1, C_2, \ldots, C_m\}$ is the *reverse lexicographic labeling* of ϕ (or that the clauses of ϕ are in *reverse lexicographic order*), if C_{j_1} precedes C_{j_2} for all $1 \le j_1 < j_2 \le m$. Clearly, the reverse lexicographic order of the clauses is determined uniquely by the ordering of the *variable* indices in V. To denote this dependence, let us use $L_\sigma(A,B)$ for the last common elements of A and B, when V is ordered by a permutation $\sigma \in \mathbb{S}_n$, and call the corresponding ordering of the clauses of ϕ the *σ-reverse lexicographic order* of ϕ, denoted by π_σ.

Given a permutation $\sigma \in \mathbb{S}_n$, let us introduce

$$\mu_\sigma(\phi) \stackrel{\text{def}}{=} \max_{1 \le j < m} |L_\sigma(C_{\pi_\sigma(j)}, C_{\pi_\sigma(j+1)})|.$$

Clearly, given a permutation σ, the value of $\mu_\sigma(\phi)$ can be computed in $O(nm)$ time.

To simplify our notations, let us assume that $\sigma = (1, \ldots, n)$ and $\pi_\sigma = (1, \ldots m)$, i.e., $\{C_1, \ldots, C_m\}$ is the σ-reverse lexicographic labeling of ϕ. Given an index $1 \le j < m$, let us introduce $L_j = L_\sigma(C_j, C_{j+1})$, $\lambda = |L_j|$, and $\phi_j = \phi_j^{\pi_\sigma}$ $(= C_1 \wedge \cdots \wedge C_j)$. By definition, we have $\lambda \le \mu_\sigma(\phi)$. Furthermore, let $L_j = \{i_1, i_2, \ldots, i_\lambda\}$, where $i_1 < \cdots < i_\lambda$, and i_0 is the largest element in $C_{j+1} \setminus L_j$. Clearly, $\{i_0, \ldots, i_\lambda\}$ is the last $\lambda + 1$ elements of L_{j+1}.

Let $[i] = \{1, \ldots, i\}$ and consider the following subsets of V:

$$S_\ell = [i_\ell - 1] \cup \bigcup_{k=\ell+1}^{\lambda} \{i_k\} \quad \text{for all} \quad \ell = 0, \ldots, \lambda. \tag{4}$$

Lemma 3. *For all $1 \le j < m$ we have $\phi_j = \phi_{S_0} \wedge \cdots \wedge \phi_{S_\lambda}$.*

Lemma 4. *For every $j = 1, 2, \ldots, m$ we have $k\,(\le 1 + \mu_\sigma(\phi))$ subsets $S_{j,1}$, $S_{j,2}$, $\ldots$, $S_{j,k}$ of V such that (3) holds.*

Theorem 2. *For every CNF ϕ and permutation σ of V, we have $|\nu(\pi_\sigma)| \le |\phi^*|^{1+\mu_\sigma(\phi)}$, and thus* LR-Mult *computes ϕ^* in $O(nm|\phi^*|^{1+\mu_\sigma(\phi)} \min\{m, |\phi^*|^{1+\mu_\sigma(\phi)}\})$ time.*

Proof. The theorem follows from Proposition 1 and Lemma 4. □

We shall show in the next subsections that even with $\sigma = (1, 2, \ldots, n)$, the class of CNF's ϕ for which $\mu_\sigma(\phi)$ is a fixed constant includes several well-known classes, proving that LR-Mult provides an efficient dualization for all these cases. Before turning to special types of CNF's, let us observe a useful property of the sets introduced in (4).

Lemma 5. *For every $\ell = 0, \ldots, \lambda$, the sets in $(\phi_{S_0} \cup \phi_{S_1} \cup \cdots \cup \phi_{S_\ell}) \setminus (\phi_{S_{\ell+1}} \cup \cdots \cup \phi_{S_\lambda})$ all contain $L = \{i_{\ell+1}, \ldots, i_\lambda\}$ as their last elements according to π_σ.*

Unless otherwise stated, let us assume in the sequel that $\sigma = (1, 2, \ldots, n)$ and eliminate it from our notations, and let $\pi = \pi_\sigma$.

3.1 Degenerate CNF's

Given a CNF ϕ, let us denote by $\Delta(\phi) = \max_{i \in V} \deg_\phi(i)$ the maximum degree of a variable in ϕ. For a given k, we say that ϕ has *bounded occurrences* if $\Delta(\phi) \leq k$. More generally, a CNF ϕ is said to be *k-degenerate* [13], for an integer $k \in \mathbb{Z}_+$, if for any $S \subseteq V$, $\min_{i \in S} \deg_{\phi_S}(i) \leq k$. Equivalently, ϕ is k-degenerate if and only if there exists a permutation $\sigma \in \mathbb{S}_n$ of the variables such that, for all $i = 1, \ldots, n$, $\deg_{\phi_{[i]}}(i) \leq k$. Here we note that such a permutation can be computed in $O(nm)$ time [13]. This class includes for instance formulae of bounded occurrences, bounded hypertree-width; see [13]. The following statement thus generalizes the results of [9].

Theorem 3. *If ϕ is a k-degenerate CNF and σ is a permutation of variables such that $\deg_{\phi_{[i]}}(i) \leq k$ for all $i = 1, \ldots, n$, then we have $\nu(\pi_\sigma) \leq |\phi^*| n^{k-1}$, and thus* LR-Mult *computes ϕ^* in $O(n^k m |\phi^*| \min\{m, n^{k-1}|\phi^*|\})$ time.*

Proof. Assume without loss of generality that $\sigma = (1, \ldots, n)$ is a permutation of variables such that $\deg_{\phi_{[i]}}(i) \leq k$ for all $i = 1, \ldots, n$. Let j be an integer in $[m-1]$. If $L_j = \emptyset$, then $\phi_j = \phi_{S_0}$ and hence $|(\phi_j)^*| \leq |\phi^*|$. On the other hand, if $L_j \neq \emptyset$, then by Lemma 5, the clauses in $\left(\phi_{S_0} \cup \phi_{S_1} \cup \cdots \cup \phi_{S_{\lambda-1}}\right) \setminus \phi_{S_\lambda}$ all contain i_λ as their last element, and we cannot have more than $k-1$ such clauses, since $\deg_{\phi_{[i_\lambda]}}(i_\lambda) \leq k$ and $i_\lambda \in C_{j+1}$. This implies $|(\phi_j)^*| \leq n^{k-1} |(\phi_{S_\lambda})^*| \leq n^{k-1} |\phi^*|$. □

We remark that for CNFs with bounded occurrences, any ordering σ of variables produces a good left-to-right multiplication.

3.2 CNF's with bounded (k, r)-intersections

Given a CNF ϕ, let $D_1(\phi)$ and $D_2(\phi)$ respectively denote the *dimension* and *intersection size* of ϕ, i.e., $D_1(\phi) = \max_{C \in \phi} |C|$ and $D_2(\phi) = \max_{C, C' \in \phi, C \neq C'} |C \cap C'|$. For a given r we say that ϕ has *bounded dimension* and *intersections* if $D_1(\phi) \leq r$ and $D_2(\phi) \leq r$, respectively.

We generalize classes of monotone CNF's with bounded occurrences, bounded dimension, and bounded intersection as follows. Let $k \geq 1$ and $r \geq 0$ be integers. We denote by $\mathbb{A}(k, r)$ the class of of monotone CNF formulae with (k, r)-*bounded intersections* [26]: $\phi \in \mathbb{A}(k, r)$ if for any k distinct clauses of ϕ, $C_{j_1}, \ldots, C_{j_k}$, we have $|\bigcap_{\ell=1}^{k} C_{j_\ell}| \leq r$. Note that

$$\Delta(\phi) \leq k \text{ iff } \phi \in \mathbb{A}(k+1, 0), \quad D_1(\phi) \leq r \text{ iff } \phi \in \mathbb{A}(1, r), \quad \text{and} \quad D_2(\phi) \leq r \text{ iff } \phi \in \mathbb{A}(2, r),$$

and hence, the class $\mathbb{A}(k, r)$ contains the bounded size, bounded degree, and bounded intersections CNF's as subclasses.

Lemma 6. *Let $\phi \in \mathbb{A}(k, r)$ and let σ be an arbitrary permutation of variables. Then, for any index j with $1 \leq j < m$,*

$$|(\phi_j^{\pi_\sigma})^*| \leq \begin{cases} |\phi^*|^r & \text{if } \lambda < r \\ |\phi^*|^{r+1} & \text{if } \lambda = r \\ n^{k-2} |\phi^*|^{r+1} & \text{if } \lambda > r, \end{cases}$$

where $\lambda = |L_j|$ $(= |L_\sigma(C_{\pi_\sigma(j)}, C_{\pi_\sigma(j+1)})|)$.

Lemma 7. *Let* $\phi \in \mathbb{A}(k, r)$ *and let* σ *be an arbitrary permutation of variables. Then, for any index* j *with* $1 \leq j < m$, $\lambda < r$ *holds for* $k = 1$, *and* $\lambda \leq r$ *holds for* $k = 2$, *where* $\lambda = |L_j| \, (= |L_\sigma(C_{\pi_\sigma(j)}, C_{\pi_\sigma(j+1)})|)$.

From Lemmas 6 and 7, we have the following theorem.

Theorem 4. *Let* $\phi \in \mathbb{A}(k, r)$ *and let* σ *be an arbitrary permutation of variables. Then we have*

$$\nu(\pi_\sigma) \;\leq\; \begin{cases} |\phi^*|^r & \text{if } k = 1 \\ |\phi^*|^{r+1} & \text{if } k = 2 \\ n^{k-2}|\phi^*|^{r+1} & \text{if } k \geq 3, \end{cases}$$

and thus LR-Mult *computes* ϕ^* *in*

$$\begin{array}{ll} O(nm|\phi^*|^r \min\{m, |\phi^*|^r\}) \text{ time} & \text{if } k = 1, \\ O(nm|\phi^*|^{r+1} \min\{m, |\phi^*|^{r+1}\}) \text{ time} & \text{if } k = 2, \text{ and} \\ O(n^{k-1}m|\phi^*|^{r+1} \min\{m, n^{k-2}|\phi^*|^{r+1}\}) \text{ time} & \text{if } k \geq 3. \end{array}$$

As a corollary, for prime monotone CNFs ϕ with bounded degree $\Delta(\phi) \leq k$, LR-Mult computes ϕ^* in $O(n^k m|\phi^*| \min\{m, n^{k-1}|\phi^*|\})$ time, which matches Theorem 3.

4 Multiplication-Tree Orderings

Given a monotone CNF formula ϕ, we build a binary tree $\mathbf{T}$, which we call a *multiplication* tree, each node v of which is associated with a monotone CNF $\phi(v)$ as follows:

(I) if v is a leaf then $\phi(v)$ is an individual clause of ϕ and every clause of ϕ appears uniquely in a leaf of $\mathbf{T}$;

(II) if v is an internal node, then it has two children u and w such that $\phi(v) = \phi(u) \wedge \phi(w)$, i.e., $\phi(v)$ is the conjunction of the subset of clauses of ϕ appearing in the leaves of the subtree of $\mathbf{T}$ rooted at v.

For a binary multiplication tree $\mathbf{T}$, we fix a planar embedding of $\mathbf{T}$ and let $\pi_{\mathbf{T}}$ be the order of clauses defined by the *left-to-right traversal* of the leaves of $\mathbf{T}$. Namely, $\pi_{\mathbf{T}}$ is obtained in the depth-first search from the root of $\mathbf{T}$ in which at each node, the left child is visited before the right one.

Note that any ordering π of clauses in ϕ can be represented by $\pi = \pi_{\mathbf{T}}$ for some multiplication tree. Denote by $\mathcal{N}(\mathbf{T})$ the set of nodes of the tree $\mathbf{T}$. For a node $v \in \mathcal{N}(\mathbf{T})$, let ϕ^v be the subformula of ϕ obtained by the left-to-right traversal of the leaves of $\mathbf{T}$ upto the right-most leaf of the subtree rooted at v: $\phi^v = \phi_r^{\pi_{\mathbf{T}}} \; (= \bigwedge_{i=1}^r C_{\pi_{\mathbf{T}}(i)})$, where r is the number of leaves, counted from the left-most leaf of $\mathbf{T}$, up to the right-most leaf of the subtree rooted at v. In what follows we denote by $\nu(\mathbf{T})$ the size of a maximum intermediate DNF produced during LR-Mult$(\phi, \pi_{\mathbf{T}})$:

$$\nu(\mathbf{T}) = \nu(\pi_{\mathbf{T}}) = \max_{v \in \mathcal{N}(\mathbf{T})} \{|(\phi^v)^*|\}.$$

Procedure Construct-Tree-A(ϕ, v)**:**
Input: A prime monotone CNF ϕ and a node v of the tree.
Output: A proper binary multiplication tree for ϕ rooted at v.

$\phi(v)$:= ϕ
if $|\phi(v)| > 1$
  Construct the left and right children left(v) and right(v) of v
  i:= argmin$\{\deg_\phi(i) :\ i \in V(\phi)\}$
  Call Construct-Tree-A($\phi_{V(\phi)\setminus\{i\}}$, left($v$))
  Call Construct-Tree-A($\phi_{(i)}$, right(v))
halt

Fig. 2. Procedure **Construct-Tree-A** to construct a proper multiplication tree for ϕ

We denote respectively by $p(v)$, left(v), and right(v), the parent, left and right children of node $v \in \mathcal{N}(\mathbf{T})$.

A binary multiplication tree $\mathbf{T}$ is called *proper* if for every $v \in \mathcal{N}(\mathbf{T})$, the set ϕ(left(v)) is a projection of $\phi(v)$, i.e., there exists a set $S \subseteq V(\phi)$ such that $\phi(v)_S = \phi(\text{left}(v))$. Call a node $v \in \mathcal{N}(\mathbf{T})$ an L-node (resp., R-node) if v is the left (resp., right) child of its parent in $\mathbf{T}$ (see Figure 4 in the appendix). Define the *right-depth* of $v \in \mathcal{N}(\mathbf{T})$, denoted by $\mathrm{d}(v)$, to be one plus the number of R-nodes in the path from the root $r(\mathbf{T})$ of $\mathbf{T}$ to v, and define the right-depth of $\mathbf{T}$, by $d(\mathbf{T}) = \max_{v\in\mathcal{N}(\mathbf{T})} \mathrm{d}(v)$.

Theorem 5. *Let ϕ be a monotone CNF. If $\mathbf{T}$ be a proper binary multiplication tree of ϕ, then we have $\nu(\mathbf{T}) \leq |\psi^*|^{d(\mathbf{T})}$.*

4.1 Quasi-Polynomial Cases

Conformal CNF's. There are several equivalent definitions for conformal CNF's (see [3, Page 90]). The most convenient for our purposes is the following: For an integer $k \geq 1$, a monotone CNF ϕ is called *k-conformal* if for every subset of variables $X \subseteq V(\phi)$, X is contained in a clause of ϕ whenever each subset of X of cardinality at most k is contained in a clause of ϕ. One can easily verify that $\phi \in \mathbb{A}(k, r)$ implies that ϕ is $(k + r)$-conformal. Thus the class of CNF's with bounded conformality includes as a special case the CNF's with bounded intersections considered in the previous section.

Although the prime DNF representation of a k-conformal CNF can be computed in polynomial time if k is constant [26], we can only show a quasi-polynomial bound for the left-to-right multiplication.

Lemma 8. *Let $\phi = \bigwedge_{j=1}^m C_j$ be a k-conformal prime monotone CNF. Then there exists a proper binary multiplication tree $\mathbf{T}$ with $\mathrm{d}(\mathbf{T}) \leq k \ln m + 1$.*

Proof. We use a simple procedure shown in Figure 2, combined with the following claim.

Claim (C1). Let $\phi' \subseteq \phi$ be a subformula of ϕ such that $|\phi'| > 1$. Then there exists an infrequent variable $i \in V(\phi')$: $|\phi'_{(i)}| \leq (1 - \frac{1}{k})|\phi'|$.

We now argue that the right-depth of $\mathbf{T}$ is logarithmic. Consider a node $v \in \mathcal{N}(\mathbf{T})$, and let $u_1, \ldots, u_h$ be the R-nodes in the path from the root $\mathbf{r}(\mathbf{T})$ to v, ordered by increasing distance from $\mathbf{r}(\mathbf{T})$. Then by the selection of the branching variable, $|\phi(u_\ell)| \leq (1-1/k)|\phi(p(u_\ell))|$ for all $\ell = 1, \ldots, h$. It follows that $|\phi(u_1)| \leq (1-1/k)|\phi| = (1-1/k)m$ and $|\phi(u_{\ell+1})| \leq (1-1/k)|\phi(u_\ell)|$ for $\ell = 1, \ldots, h$, and hence $|\phi(u_h)| \leq (1-1/k)^h m$. Since $|\phi(u_h)| \geq 1$, we get $h \leq k \ln m$. □

Theorem 6. *Let $\phi = \bigwedge_{j=1}^m C_j$ be a k-conformal prime monotone CNF. Then Procedure* **Construct-Tree-A** *produces a permutation $\pi_\mathbf{T}$ of the clauses such that $\nu(\pi_\mathbf{T}) \leq |\phi^*|^{k \ln m+1}$, and thus* LR-Mult *computes ϕ^* in $O(nm|\phi^*|^{k \ln m+1} \min\{m, |\phi^*|^{k \ln m+1}\})$ time.*

CNF's of read-once expressions. A formula φ is called *read-once* if it can be written as an $\wedge - \vee$ formula in which every variable in $V(\varphi)$ appears exactly once. A well-known equivalent definition is that ϕ is a prime monotone CNF which can be represented by a read-once expression if and only if

$$|C \cap t| = 1 \quad \text{for every clause } C \in \phi \text{ and every term } t \in \phi^*. \tag{5}$$

Lemma 9. *Let $\phi = \bigwedge_{j=1}^m C_j$ be a prime monotone CNF with a read-once expression. Then there exists a proper binary multiplication tree* $\mathbf{T}$ *with* $\mathrm{d}(\mathbf{T}) \leq \log m + 1$.

Proof. We use the following claim to construct a tree $\mathbf{T}$ by the procedure shown in Figure 2.

Claim (C2). Let $\phi' \subseteq \phi$ be a subformula of ϕ such that $|\phi'| > 1$. Then there exists an infrequent variable $i \in V(\phi')$: $|\phi'_{(i)}| \leq \frac{1}{2}|\phi'|$.

The rest of the proof is the same as in Lemma 8. □

Theorem 7. *Let $\phi = \bigwedge_{j=1}^m C_j$ be a prime monotone CNF which can be represented by a read-once expression. Then Procedure* **Construct-Tree-A** *produces a permutation $\pi_\mathbf{T}$ of the clauses such that $\nu(\pi_\mathbf{T}) \leq |\phi^*|^{\log m+1}$, and thus* LR-Mult *computes ϕ^* in $O(nm|\phi^*|^{\log m+1} \min\{m, |\phi^*|^{\log m+1}\})$ time.*

4.2 General Monotone CNF's

In this section, we consider general monotone CNFs, and show that by use of the procedure in Figure 3, the left-to-right multiplication can always be done in subexponential time. The procedure constructs a proper binary multiplication tree for ϕ which is almost identical to the procedure in Figure 2, except that the minimum-degree variable is computed with respect to the CNF ϕ' containing only small clauses of ϕ.

Let us begin with the following two simple lemmas.

Lemma 10. *Let ϕ be a prime monotone CNF, and let k be a positive integer with $k < n/2$. If every clause of ϕ has size at least $n - k$, then any permutation π has $\nu(\pi) \leq n^{k+1}$.*

Procedure Construct-Tree-B(ϕ, v)**:**
Input: A prime monotone CNF ϕ and a node v of the tree.
Output: A proper binary multiplication tree for ϕ rooted at v.

$\phi(v)$:= ϕ
if $|\phi(v)| > 1$
Construct the left and right children left(v) and right(v) of v
ϕ':= $\bigwedge\{C \in \phi : \ |C| \leq |V(\phi)| - \sqrt{|V(\phi)|}\}$

i:= $\mathrm{argmin}\{\deg_{\phi'}(i) : \ i \in V(\phi)\}$

Call Construct-Tree-B$(\phi_{V(\phi)\setminus\{i\}}, \mathrm{left}(v))$
Call Construct-Tree-B$(\phi_{(i)}, \mathrm{right}(v))$
halt

Fig. 3. Procedure **Construct-Tree-B** to construct a proper multiplication tree for ϕ

Lemma 11. *Let ϕ be a prime monotone CNF, let k be a positive integer, and let ϕ' be a subformula of ϕ. If every clause of ϕ' has size at most $n-k$, then there exists an infrequent variable $i \in V(\phi)$ with respect to ϕ': $|\phi'_{(i)}| \leq (1 - \frac{k}{n})|\phi'|$.*

Let us now show that the procedure in Figure 3 produces a multiplication tree with small right-depth.

Theorem 8. *Let $\phi = \bigwedge_{j=1}^m C_j$ be a prime monotone CNF. Then Procedure* **Construct-Tree-B** *produces a permutation $\pi_{\mathbf{T}}$ of the clauses such that $\nu(\pi_{\mathbf{T}}) \leq n^{\sqrt{n}+1}|\phi^*|^{\sqrt{n}\ln m}$, and thus* LR-Mult *computes ϕ^* in $O(n^{\sqrt{n}+2}m|\phi^*|^{\sqrt{n}\ln m} \quad \min\{m, n^{\sqrt{n}+1}|\phi^*|^{\sqrt{n}\ln m}\})$ time.*

Proof. Consider any leaf $v \in \mathcal{N}(\mathbf{T})$ and let $\mathbf{P}$ be the path from the root $\mathbf{r}(\mathbf{T})$ to v. For a node w in $\mathbf{P}$, let $V(w) = V(\phi(w))$ and $\phi'(w) = \bigwedge\{C \in \phi(w) : \ |C| \leq |V(\phi(w))| - \sqrt{|V(\phi(w))|}\}$. Note that there is a node w of $\mathbf{P}$ such that $\phi'(w) = \emptyset$. Let w_0 be the closest such node to the root, and let $u_1, u_2, \ldots, u_h$ be the R-nodes in the path $\mathbf{P}$ between $\mathbf{r}(\mathbf{T})$ and w_0, ordered by increasing distance from $\mathbf{r}(\mathbf{T})$.

For $\ell = 0, 1, \ldots, h$, let $n_\ell = |V(u_\ell)|$, where we assume $u_0 = \mathbf{r}(\mathbf{T})$. Note that

$$V(u_\ell) \subseteq V(p(u_\ell)) \subseteq V(u_{\ell-1}) \quad \text{and} \quad \phi'(u_\ell) \subseteq \phi'(p(u_\ell)) \subseteq \phi'(u_{\ell-1}),$$

for $\ell = 1, \ldots, h$. In particular, Lemma 11 implies $|\phi'(u_\ell)| \leq (1 - \frac{1}{\sqrt{n_{\ell-1}}})|\phi'(u_{\ell-1})|$, for $\ell = 1, \ldots, h$, and thus $|\phi'(u_h)| \leq (1 - 1/\sqrt{n_0})^h |\phi'(u_0)|$, Since $|\phi'(u_h)| \geq 1$, we conclude that $\mathrm{d}(w_0) = l + 1 \leq \sqrt{n}\ln m + 1$, where $n = n_0$.

From Theorem 5, we know that $|(\phi^v)^*| \leq |\phi^*|^{\mathrm{d}(w_0)-1}|(\phi'')^*|$, where $\phi'' \subseteq \phi(w_0)$ consists of clauses in $\phi(w_0) \cap \phi^v$. By definition of w_0, we have $|\phi'(w_0)| = 0$ and thus $\phi(w_0)$ consists only of clauses of size at least $|V(\phi(w_0))| - \sqrt{|V(\phi(w_0))|}$. Thus $|(\phi'')^*| \leq n^{\sqrt{n}+1}$ by Lemma 10, and hence

$$|(\phi^v)^*| \ \leq \ |\phi^*|^{\mathrm{d}(w_0)-1} n^{\sqrt{n}+1} \ \leq \ n^{\sqrt{n}+1}|\phi^*|^{\sqrt{n}\ln m}.$$

□

References

1. M. Anthony and N. Biggs. *Computational learning theory: an introduction*. Cambridge University Press, New York, NY, USA, 1992.
2. J. Bailey, T. Manoukian, and K. Ramamohanarao. A fast algorithm for computing hypergraph transversals and its application in mining emerging patterns. In *ICDM*, pages 485–488, 2003.
3. C. Berge. *Hypergraphs*. Elsevier-North Holand, Amsterdam, 1989.
4. J. C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive boolean functions. *Information and Computation*, 123(1):50–63, 1995.
5. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. Dual-bounded generating problems: All minimal integer solutions for a monotone system of linear inequalities. *SIAM Journal on Computing*, 31(5):1624–1643, 2002.
6. E. Boros, V. Gurvich, and P.L. Hammer. Dual subimplicants of positive boolean functions. *Optim. Methods Softw.*, 10:147–156, 1998.
7. E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On maximal frequent and minimal infrequent sets in binary matrices. *Annals of Mathematics and Artificial Intelligence*, 39(3):211–221, 2003.
8. C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, Inc., New York, NY, USA, 1987.
9. C. Domingo, N. Mishra, and L. Pitt. Efficient read-restricted monotone cnf/dnf dualization by learning with membership queries. *Machine Learning*, 37(1):89–110, 1999.
10. G. Dong and J. Li. Mining border descriptions of emerging patterns from dataset pairs. *Knowl. Inf. Syst.*, 8(2):178–202, 2005.
11. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.
12. T. Eiter and G. Gottlob. Hypergraph transversal computation and related problems in logic and ai. In *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, pages 549–564, London, UK, 2002. Springer-Verlag.
13. T. Eiter, G. Gottlob, and K. Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM Journal on Computing*, 32(2):514–537, 2003.
14. T. Eiter, K. Makino, and G. Gottlob. Computational aspects of monotone dualization: A brief survey. KBS Research Report INFSYS RR-1843-06-01, Institute of Information Systems, Vienna University of Technology, 2006, to appear in *Discrete Applied Mathematics*.
15. Thomas Eiter. Exact transversal hypergraphs and application to Boolean μ-functions. *Journal of Symbolic Computation*, 17(3):215–225, 1994.
16. K. Elbassioni. On the complexity of the multiplication method for monotone CNF/DNF dualization. In *ESA '06: Proceedings of the 14th Annual European Symposium on Algorithms*, pages 340–351, 2006.
17. M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
18. H. Garcia-Molina and D. Barbará. How to assign votes in a distributed system. *Journal of the Association for Computing Machinery*, 32(4):841–860, 1985.
19. D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen. Data mining, hypergraph transversals, and machine learning (extended abstract). In *PODS '97: Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 209–216, New York, NY, USA, 1997. ACM Press.
20. T. Ibaraki and T. Kameda. A theory of coteries: Mutual exclusion in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 4(7):779–794, 1993.

21. D. S. Johnson. Open and closed problems in NP-completeness. In *Lecture given at the International School of Mathematics "G. Stampacchia": Summer School "NP-Completeness: The First 20 Years"*, Erice (Sicily), Italy, 20 - 27 June 1991.
22. D. J. Kavvadias, C. H. Papadimitriou, and M. Sideri. On horn envelopes and hypergraph transversals. In *ISAAC '93: Proceedingsof the 4th International Symposium on Algorithms and Computation*, pages 399–405, 1993.
23. Dimitris J. Kavvadias and Elias C. Stavropoulos. An efficient algorithm for the transversal hypergraph generation. *J. Graph Algorithms Appl.*, 9(2):239–264, 2005.
24. L. Khachiyan. Transversal hypergraphs and families of polyhedral cones. In *Advances in Convex Analysis and Global Optimization, honoring the memory of K. Carathéodory*, pages 105–118, Dordrecht/Boston/London, 2000. Kluwer Academic Publishers.
25. L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, and K. Makino. Enumerating spanning and connected subsets in graphs and matroids. In *ESA*, pages 444–455, 2006.
26. L. Khachiyan, E. Boros, K. Elbassioni, and V. Gurvich. On the dualization of hypergraphs with bounded edge-intersections and other related classes of hypergraphs. *Theor. Comput. Sci.*, 382(2):139–150, 2007.
27. L. Khachiyan, E. Boros, K. Elbassioni, V. Gurvich, and K. Makino. On the complexity of some enumeration problems for matroids. *SIAM Journal on Discrete Mathematics*, 19(4):966–984, 2005.
28. E. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558–565, 1980.
29. L. Lovász. Combinatorial optimization: some problems and trends. DIMACS Technical Report 92-53, Rutgers University, 2000.
30. H. Mannila. Local and global methods in data mining: Basic techniques and open problems. In *ICALP*, pages 57–68, 2002.
31. H. Mannila and K. J. Räihä. Design by example: An application of armstrong relations. *Journal of Computer and System Sciences*, 33(2):126–141, 1986.
32. C. H. Papadimitriou. Np-completeness: A retrospective. In *ICALP '97: Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, pages 2–6, London, UK, 1997. Springer-Verlag.
33. K. G. Ramamurthy. *Coherent Structures and Simple Games*. Kluwer Academic Publishers, 1990.
34. K. Takata. On the sequential method for listing minimal hitting sets. In *DM & DM 2002: Proceedings of Workshop on Discrete Mathematics and Data Mining, 2nd SIAM International Conference on Data Mining*, pages 109–120, 2002.
35. T. Uno. A practical fast algorithm for enumerating minimal set coverings. In *Proceedings 83rd SIGAL Conference of the Information Processing Society of Japan (IPSJ), (in Japanese)*, pages 9–16, Toyko, March 15, 2002.

Appendix

Proof of Proposition 1.

The proof follows from Lemmas 12 and 13 below.

Lemma 12. *For a prime monotone CNF ϕ and a permutation $\pi \in \mathbb{S}_m$, Procedure* LR-Mult(ϕ, π) *can be done in $O(nm\nu(\pi)^2)$ time.*

Proof. Let us show that, for each j, ψ_j can be computed in $O(n\nu(\pi)^2)$ time from a prime monotone DNF ψ_{j-1} and a monotone clause $C_{\pi(j)}$, which completes the proof, since we have m such j's.

Note that

$$\psi_j \equiv \bigvee_{t \in \psi_{j-1}} \bigvee_{i \in C_{\pi(j)}} (t \wedge x_i).$$

Thus if $t \in \psi_{j-1}$ contains some $i \in C_{\pi(j)}$, then $t \in \psi_j$ and we have

$$\psi_j \equiv \bigvee_{\substack{t \in \psi_{j-1}: \\ t \cap C_{\pi(j)} \neq \emptyset}} t \;\vee \bigvee_{\substack{t \in \psi_{j-1}: \\ t \cap C_{\pi(j)} = \emptyset}} \bigvee_{i \in C_{\pi(j)}} (t \wedge x_i), \tag{6}$$

since $t \subseteq t \wedge x_i$ for all $i \in C_{\pi(j)}$ and t is contained in prime ψ_{j-1}. Moreover, for a term $t \in \psi_{j-1}$ with $t \cap C_{\pi(j)} = \emptyset$ and an index $i \in C_{\pi(j)}$, we claim that $t \wedge x_i \in \psi_j$ if and only if there is no $t' \in \psi_{j-1}$ such that $t' \setminus t = \{i\}$.

For the only-if part, let t' be a term in ψ_{j-1} such that $t' \setminus t = \{i\}$. Note that $t' \in \psi_j$ holds by $t' \cap C_{\pi(j)} \neq \emptyset$, and $t' \subsetneq t \wedge x_i$ holds, since $t' \setminus t = \{i\}$ and $t \setminus t' \neq \emptyset$ (by the primality of ψ_{j-1}). Therefore, $t \wedge x_i \notin \psi_j$.

For the if part, let us assume that no $t' \in \psi_{j-1}$ satisfies $t' \setminus t = \{i\}$. Then all t' $(\neq t)$ in ψ_{j-1} satisfy $t' \not\subseteq t \wedge x_i$, since $t' \setminus t \neq \emptyset$. This implies $t \wedge x_i \in \psi_j$.

Since $|\psi_{j-1}|, |\psi_j| \leq \nu(\pi)$, it follows from our claim that ψ_j can be computed in $O(n\nu(\pi)^2)$ time. □

Lemma 13. *For a prime monotone CNF ϕ and a permutation $\pi \in \mathbb{S}_m$, Procedure* LR-Mult(ϕ, π) *can be done in $O(nm^2\nu(\pi))$ time.*

Let us show that, for each j, ψ_j can be computed in $O(nm\nu(\pi))$ time from a prime monotone DNF ψ_{j-1} and a monotone clause $C_{\pi(j)}$, which completes the proof, since we have m such j's.

By the discussion in the proof of Lemma 13, $t \in \psi_j$ holds for any $t \in \psi_{j-1}$ with $t \cap C_{\pi(j)} \neq \emptyset$, and ψ_j can be represented by (6). Let t be a term in ψ_{j-1} with $t \cap C_{\pi(j)} = \emptyset$, and for an $\ell \in t$, let $\mathcal{C}_\ell = \{C \in \phi^\pi_{j-1} \mid C \cap t = \{\ell\}\}$. By definition, $\mathcal{C}_\ell \cap \mathcal{C}_{\ell'} = \emptyset$ for any ℓ and ℓ' with $\ell \neq \ell'$, and $\mathcal{C}_\ell \neq \emptyset$ for any $\ell \in t$, since $\psi_{j-1} = (\phi^\pi_{j-1})^*$. We now claim that $t \wedge x_i \in \psi_j$ for $i \in C_{\pi(j)}$ if and only if no $\ell \in t$ satisfies $\mathcal{C}_\ell = \{C \in \mathcal{C}_\ell \mid C \ni i\}$.

Fo the only-if part, let ℓ be an index in t with $\mathcal{C}_\ell = \{C \in \mathcal{C}_\ell \mid C \ni i\}$. Then no $C \in \phi^\pi_{j-1}$ satisfies $C \cap (t \wedge x_i) = \{\ell\}$, and hence $t \wedge x_i$ is not a minimal transversal of ϕ^π_j, which means $t \wedge x_i \notin \psi_j$ $(= (\phi^\pi_j)^*)$.

Fo the if part, let us assume that no ℓ in t satisfies $\mathcal{C}_\ell = \{C \in \mathcal{C}_\ell \mid C \ni i\}$. Then for each $\ell \in t \wedge x_i$, there exists a clause C in ϕ_j^π such that $C \cap (t \wedge x_i) = \{\ell\}$. This implies that $t \wedge x_i$ is a minimal transversal of ϕ_j^π, which completes the proof of the claim.

Note that $\bigcup_{\ell \in t} \mathcal{C}_\ell \subseteq \phi_{j-1}^\pi$ and $\mathcal{C}_\ell \cap \mathcal{C}_{\ell'} = \emptyset$ for any ℓ, ℓ' with $\ell \neq \ell'$. Thus from the claim, it is not difficult to see that ψ_j can be computed in $O(nm\nu(\pi))$ time. □

Proof of Lemma 6. For simplicity, let $\sigma = (1, \dots, n)$ and $\pi_\sigma = (1, \dots, m)$. For an index j with $1 \leq j < m$, let $L_j = \{i_1, i_2, \dots, i_\lambda\}$, where $i_1 < \cdots < i_\lambda$, and let i_0 be the largest element in $C_{j+1} \setminus L_j$.

By Lemmas 1 and 3, we have $|(\phi_j^{\pi_\sigma})^*| \leq |\phi^*|^{\lambda+1}$, and thus the statement in the lemma holds for $\lambda \leq r$. If $\lambda > r$, it follows from Lemma 5 that

$$U \ \stackrel{\text{def}}{=} \ \left(\phi_{S_0} \cup \phi_{S_1} \cup \cdots \cup \phi_{S_{\lambda-r-1}}\right) \setminus \left(\phi_{S_{\lambda-r}} \cup \cdots \cup \phi_{S_\lambda}\right)$$

contains $L = \{i_{\lambda-r}, \dots, i_\lambda\}$. Thus we have

$$|U| \ \leq \ k-2,$$

since $|L| = r+1$, $C_{j+1} \supseteq L$, and $C_{j+1} \not\in U$. This implies

$$|(\phi_j^{\pi_\sigma})^*| \ \leq \ n^{k-2} \prod_{\ell=\lambda-r}^{\lambda} |(\phi_{S_\ell})^*| \ \leq n^{k-2} |\phi^*|^{r+1}.$$

□

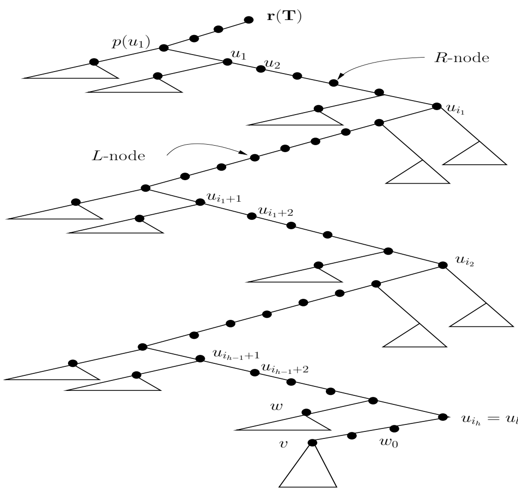

Fig. 4. The path from $\mathbf{r}(\mathbf{T})$ to v.

Proof of Theorem 5. For an arbitrary node $v \in \mathcal{N}(\mathbf{T})$, let $\mathcal{L}$ and $\mathcal{R}$ be respectively the sets of L-nodes and R-nodes in the path from the root $\mathbf{r}(\mathbf{T})$ to v. We can assume, without loss of generality, that v is an L-node, since otherwise, we have $\phi^v = \phi^{p(v)}$ and we can repeatedly replace v by $p(v)$. We shall prove by induction on right(v) that

$$|(\phi^v)^*| \leq |\phi^*|^{\mathrm{d}(v)}. \tag{7}$$

Since $\mathbf{T}$ is proper, for every L-node u, there is a set $S_u \subseteq V(\phi)$ such that $\phi(p(u))_{S_u} = \phi(u)$. In particular, if $\mathrm{d}(v) = 1$ then there exists a set S such that $\phi^v = \phi(v) = \phi_S$, and hence (7) holds by Proposition 2. If $\mathrm{d}(v) > 1$, then R-nodes u in the path do not satisfy $\phi(p(u))_{S_u} = \phi(u)$ in general, and we have to argue slightly differently. Let w be the left child of the parent of the last R-node in the path from $\mathbf{r}(\mathbf{T})$ to v (see Figure 4). Then, $\mathrm{d}(w) = \mathrm{d}(v) - 1$. We assume by induction that

$$|(\phi^w)^*| \leq |\phi^*|^{\mathrm{d}(w)} = |\phi^*|^{\mathrm{d}(v)-1}. \tag{8}$$

Let $(\phi^w)^* = \bigvee_{j=1}^k t_j$, where $k = |(\phi^w)^*|$. Then

$$(\phi^v)^* \equiv (\phi^w)^* \wedge \phi(v) \;\equiv\; \bigvee_{j=1}^{k} (t_j \wedge \phi(v)) \;\equiv\; \bigvee_{j=1}^{k} (t_j \wedge \phi(v)_{V\setminus t_j}).$$

where $\phi(v)_{V\setminus t_j} = \wedge\{C \in \phi(v) :\ C \cap t_j = \emptyset\}$. We claim that $\phi(v)_{V\setminus t_j} = \phi_S$, where

$$S = \left(\bigcap_{u\in\mathcal{L}} S_u\right) \setminus t_j.$$

Clearly, $\phi(v)_{V\setminus t_j} \subseteq \phi_S$ by definitions of $\phi(v)_{V\setminus t_j}$ and S. Conversely, let C be a clause of ϕ_S. Then $C \subseteq S_u$ for all $u \in \mathcal{L}$ and $C \cap t_j = \emptyset$. For every $u \in \mathcal{L}$, $C \in \phi(p(u))$ implies $C \in \phi(u)$, since $\mathbf{T}$ is proper. Note that $\phi^w = \bigwedge_{u\in\mathcal{R}} \phi(\mathrm{left}(p(u)))$, where left$(p(u))$ is the left sibling of node u in $\mathbf{T}$ (see Figure 4). This implies in particular that $C \notin \phi^w$, since C is disjoint from a term t_j of $(\phi^w)^*$. For every $u \in \mathcal{R}$, $C \in \phi(p(u))$ implies $C \in \phi(u)$. Therefore, starting from the root, C will end up in $\phi(v)$. Since $C \cap t_j = \emptyset$, we have $C \in \phi(v)_{V\setminus t_j}$, establishing our claim.

It follows from this claim and Proposition 2 that $|(\phi(v)_{V\setminus t_j})^*| \leq |\phi^*|$, and hence by (8),

$$|(\phi^v)^*| \;\leq\; \sum_{j=1}^{k} \max\{|(\phi(v)_{V\setminus t_j})^*|, 1\} \;\leq\; k|\phi^*| \;=\; |(\phi^w)^*||\phi^*| \;\leq\; |\phi^*|^{\mathrm{d}(v)}.$$

This shows (7) and proves the lemma. $\square$

Proof of Claim (C1). If every subset $X \subseteq V(\phi')$ of size at most k is contained in some clause of ϕ', then $V(\phi')$ is contained in some clause C of ϕ by the k-conformality of ϕ. This implies $C = V(\phi')$. Since ϕ' is prime, we have $|\phi'| = 1$, which is a contradiction. Thus there exists a set $X \subseteq V(\phi')$ of size at most k such that X is not contained in any clause of ϕ'. This gives $\phi' = \bigwedge_{i\in X} \phi'_{V(\phi')\setminus\{i\}}$, implying that there is an $i \in X$ such that $|\phi'_{V(\phi')\setminus\{i\}}| \geq |\phi'|/k$. $\square$

Proof of Claim (C2). If every pair of elements of $V(\phi')$ is contained in some clause of ϕ', then, by (5), $|t \cap V(\phi')| = 1$ for every $t \in \phi^*$. On the other hand, $\phi_{V(\phi')}$ contains at least two distinct clauses and hence $(\phi_{V(\phi')})^*$ has a term of size at least 2, which can be extended to a term of ϕ^*. This contradiction shows that there must exist a pair of elements not contained in any clause of ϕ', and hence at leat one of the elements i in the pair satisfies $|\phi'_{(i)}| \leq \frac{1}{2}|\phi'|$. □

Proof of Lemma 10. Let ϕ' be a subformula of ϕ and t be a term of $(\phi')^*$. If $|t| > k+1$, then any subterm $t' \subset t$ of size $|t'| = k+1$ must intersect every clause of ϕ'. This contradicts the primality of t. Thus every term t of $(\phi')^*$ has size at most $k+1$, and hence $|(\phi')^*| \leq n^{k+1}$. □

Proof of Lemma 11. Let $i \in V(\phi)$ be a variable of minimum degree in ϕ'. Then

$$(n-k)|\phi'| \geq \sum_{C \in \phi'} |C| = \sum_{j \in V(\phi)} \deg_{\phi'}(j) \geq n \cdot \deg_{\phi'}(i),$$

and thus $\deg_{\phi'}(i) \leq (1 - k/n)|\phi'|$. □