On the expressive power of CNF formulas of bounded Tree- and Clique-Width

Pascal Koiran[1] and Klaus Meer[2] *

[1] Laboratoire de l'Informatique du Parallélisme
ENS Lyon, France, e–mail: koiran@ens-lyon.fr
[2] Lehrstuhl Theoretische Informatik
BTU Cottbus, Germany, e–mail: meer@informatik.tu-cottbus.de

Abstract. Starting point of our work is a previous paper by Flarup, Koiran, and Lyaudet [5]. There the expressive power of certain families of polynomials is investigated. Among other things it is shown that polynomials arising as permanents of bounded tree-width matrices have the same expressiveness as polynomials given via arithmetic formulas. A natural question is how expressive such restricted permanent polynomials are with respect to other graph-theoretic concepts for representing polynomials over a field $\mathbb{K}$. One such is representing polynomials by formulas in conjunctive normal form. Here, a monomial occurs according to whether the exponent vector satisfies a given CNF formula or not. We can in a canonical way assign a graph to such a CNF formula and speak about the tree-width of the related CNF polynomial.
In this paper we show that the expressiveness of CNF polynomials of bounded tree-width again gives precisely arithmetic formulas. We then study how far the approach of evaluating subclasses of permanents efficiently using a reduction to CNF formulas of bounded tree-width leads. We show that there does not exist a family of CNF polynomials of bounded tree-width which can express general permanent polynomials. The statement is unconditional. An analoguous result for CNF polynomials of bounded clique-width is given, this time under the assumption that $\#P \not\subseteq FP/poly$.
The paper contributes to the comparison between classical Boolean complexity and algebraic approaches like Valiant's one.

1 Introduction

An active field of research in computational complexity is devoted to the design of efficient algorithms for subclasses of problems which in full generality likely are hard to solve. It is common in this area to define such subclasses via bounding some significant problem parameters. Typical such parameters are the tree- and clique-width if a graph structure is involved in the problem's description.

* Support of the following institutions is gratefully acknowledged: ENS Lyon; the French embassy in Denmark, Service de Coopération et d'Action Culturelle, Ref.:39/2007-CSU 8.2.1; the Danish Agency for Science, Technology and Innovation FNU.

In the present paper we consider a graph-theoretic approach in order to deal with problems that are related to families of polynomials. These families are given in a particular manner through certain Boolean formulas in conjunctive normal form, shortly CNF formulas. More precisely, we consider functions of the form

$$f(x) = \sum_{e\in\{0,1\}^n} \varphi(e)x^e, x \in \{0,1\}^n \text{ , for some } n \in \mathbb{N}, \qquad (*)$$

where φ is a CNF formula in n Boolean variables. We are interested in the question how expressive such a representation of polynomials is and under which additional conditions $f(x)$ can be evaluated efficiently. Fischer, Makowsky, and Ravve [4], extending earlier results from [2], have shown that the counting SAT problem, i.e. computing $\sum\limits_{e\in\{0,1\}^n} \varphi(e)$ for a CNF formula φ can be solved in time $O(n \cdot 4^k)$ if a certain bipartite graph G_φ canonically attached to φ is of bounded tree-width k.

Our first main result precisely characterizes the expressive power of functions of form $(*)$ when G_φ is of bounded tree-width. It is shown that the class of these polynomials equals both the class of polynomials representable by arithmetic formulas of polynomial size and the class of functions obtained as permanents of matrices of bounded tree-width and polynomially bounded dimension. Here, equality of the latter two concepts was known before due to a result of Flarup, Koiran, and Lyaudet [5].

Recall that in Valiant's algebraic model of computation for families of polynomials the permanent is VNP complete and thus likely not efficiently computable. Though an unconditional proof of this conjecture seems extremely difficult we next show that at least trying to obtain an efficient algorithm for computing permanents through formulas of type $(*)$ with G_φ of bounded tree-width must fail. This result is unconditioned in that it does not rely on any open conjecture in complexity theory.

Next, we pose the corresponding question for CNF formulas of bounded cliquewidth. Using another result from [4] we show that expressing the permanent of an arbitrary matrix by formulas of type $(*)$, this time with G_φ of bounded clique-width would imply $\#P \subseteq FP/poly$ and thus is unlikely.

The paper is organized as follows. In Section 2 we recall basic definitions as well as the needed results from [4] and [5]. Section 3 first shows how permanents of matrices of bounded tree-width can be expressed via polynomials of form $\sum\limits_{e\in\{0,1\}^n} \varphi(e)x^e$ with G_φ of bounded tree-width. Then, we extend a result from [4] to link such polynomials to arithmetic formulas. The results in [5] now imply equivalence of all three notions. In Section 4 the above mentioned negative results concerning expressiveness of (general) permanents by CNF formulas of bounded tree- or clique-width are proven.

Our results contribute to the comparison of Boolean and algebraic complexity. In particular, we consider it to be interesting to find more results like Theorem 8 below which states that certain properties **cannot** be expressed via (certain) graphs of bounded tree-width.

2 Basic definitions

We start briefly collecting basic definitions and results that are needed below.

2.1 Arithmetic circuits

Definition 1. *a) An arithmetic circuit is a finite, acyclic, directed graph. Vertices have indegree 0 or 2, where those with indegree 0 are referred to as* inputs*. A single vertex must have outdegree 0, and is referred to as* output*. Each vertex of indegree 2 must be labeled by either + or ×, thus representing computation. Vertices are commonly referred to as* gates*. By choosing as input nodes either some variables x or constants from a field $\mathbb{K}$ a circuit in a natural way represents a multivariate polynomial over $\mathbb{K}$.*

b) An arithmetic formula *is a circuit for which all gates except the output have outdegree 1 (therefore, reuse of partial results is not allowed in arithmetic formulas).*

c) The size *of a circuit is the total number of* gates *in the circuit.*

2.2 Tree- and clique-width

Treewidth for undirected graphs is defined as follows:

Definition 2. *Let $G = \langle V, E\rangle$ be a graph. A k-tree-decomposition of G is a tree $T = \langle V_T, E_T\rangle$ such that:*

(i) Each $t \in V_T$ is labelled by a subset $X_t \subseteq V$ of size at most $k + 1$.
(ii) For each edge $(u, v) \in E$ there is a $t \in V_T$ such that $\{u, v\} \subseteq X_t$.
(iii) For each vertex $v \in V$ the set $\{t \in V_T | v \in X_T\}$ forms a (connected) subtree of T.

The tree-width of G is then the smallest k such that there exists a k-tree-decomposition for G.

Next we recall the clique-width notion.

Definition 3. *A graph G has* clique-width *at most k iff there exists a set of k labels $\mathcal{S}$ such that G can be constructed using a finite number of the following operations:*

i) $vert_a, a \in \mathcal{S}$ (create a single vertex with label a);
ii) $\phi_{a\to b}(H), a, b \in \mathcal{S}$ (rename all vertices having label a to have label b);
iii) $\eta_{a,b}(H), a, b \in \mathcal{S}, a \neq b$ (add edges between all vertices having label a and all vertices having label b);
iv) $H_1 \oplus H_2$ (disjoint union of graphs).

To each graph of clique-width k we can attach a (rooted) parse-tree whose leaves correspond to singleton graphs and whose vertices represent one of the operations above. The graph G then is represented at the root.

2.3 Permanent polynomials

Definition 4. *The permanent of an* (n,n)*-matrix* $M = (m_{i,j})$ *is defined as* $perm(M) := \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$ *, where* S_n *is the symmetric group.*

We are interested in representing polynomials via permanents. If M above has as entries either variables or constants from some field $\mathbb{K}$, then $f = perm(M)$ is a polynomial with coefficients in $\mathbb{K}$ (in Valiant's terms f is a projection of the permanent polynomial). One main result in [5] characterizes arithmetic formulas of polynomial size by certain such polynomials. The tree-width of a matrix $M = [m_{ij}]$ is defined to be the tree-width of the graph we get by including an edge (i,j) iff $m_{ij} \neq 0$.

Theorem 1. *([5]) Let* $(f_n)_{n\in\mathbb{N}}$ *be a family of polynomials with coefficients in a field* $\mathbb{K}$*. The following properties are equivalent:*

(i) $(f_n)_{n\in\mathbb{N}}$ *can be represented by a family of polynomial size arithmetic formulas.*
(ii) There exists a family $(M_n)_{n\in\mathbb{N}}$ *of polynomial size, bounded tree-width matrices such that the entries of* M_n *are constants from* $\mathbb{K}$ *or variables of* f_n*, and* $f_n = perm(M_n)$*.*

2.4 Clause graphs

One of our goals is to relate Theorem 1 to yet another concept, namely CNF formulas of bounded tree-width. The latter will be defined in this subsection. Our presentation follows closely [4].

Definition 5. *Let* φ *be a Boolean formula in conjunctive normal form with clauses* $C_1, \ldots, C_m$ *and Boolean variables* $x_1, \ldots, x_n$*.*

a) The signed clause graph $SI(\varphi)$ *is a bipartite graph with the* x_i *and the* C_j *as nodes. Edges connect a variable* x_i *and a clause* C_j *iff* x_i *occurs in* C_j*. An edge is signed* $+$ *or* $-$ *if* x_i *occurs positively or negated in* C_j*.*
b) The incidence graph $I(\varphi)$ *of* φ *arises from* $SI(\varphi)$ *by omiting the signs* $+,-$*.*
c) The primal graph $P(\varphi)$ *of* φ *has only the* x_i*'s as its nodes. An edge connects* x_i *and* x_j *iff both occur commonly in one of the clauses.*
d) The tree- or clique-width of a CNF formula φ *is defined to be the tree- or clique-width of* $I(\varphi)$*, respectively.*
If below we want to speak about the tree-width of $P(\varphi)$ *we mention this explicitly.*

Theorem 2. *([4]) a) Given* φ *and a tree-decomposition of* $I(\varphi)$ *of width* k *one can compute the number of satisfying assignments* $\sum_{x\in\{0,1\}^n} \varphi(x)$ *of* φ *in* $4^k n$ *arithmetic operations.*
b) Given a CNF formula φ *and a parse-tree for the signed clause graph* $SI(\varphi)$ *of clique-width* $\leq k$ *the number* $\sum_x \varphi(x)$ *of satisfying assignments of* φ *can be computed in* $O(n2^{ck})$ *many arithmetic operations.*

Below, we extend the algorithm proving Theorem 2 a) in order to relate CNF formulas to arithmetic formulas and Theorem 1. Note that similar results to those of part a) of the Theorem 2 have independently been obtained in [9].

2.5 Non-deterministic OBDDs

The final notion we need to introduce is that of deterministic and non-deterministic Ordered Binary Decision Diagrams OBDDs. For a more extensive presentation of OBDDs see [10].

Definition 6. *a) A* binary decision diagram *or BDD is a rooted directed acyclic graph having two kinds of nodes. Output nodes are nodes with no outgoing edge and are labeled with a Boolean constant from* $\{0, 1\}$. *Inner nodes are labeled with an element from some variable set* $\{x_1, \ldots, x_n\}$. *They have two outgoing edges one of which is labeled by* 0 *and the other by* 1. *The* size *of a BDD is the number of nodes of the underlying graph.*

b) A BDD is ordered (denoted OBDD) if there is an ordering of the variables such that they occur along each path from the root to an output node according to the ordering.

c) Each OBDD computes a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ *in the following way. Given an assignment for the Boolean variables* $x_1, \ldots, x_n$ *one follows starting from the root at each node labeled* x_i *that edge which is labeled with the value of* x_i. *The result obtained is the label of the output node reached.*

d) An OBDD is non-deterministic *if it contains an additional type of nodes labeled as guess nodes. Such a node has two outgoing edges that are unlabeled. Each such edge can be followed by an input. The non-deterministic OBDD computes the result 1 for input* x *iff there is at least one path leading to a leaf labeled by* 1 *that can be followed for input* x.

A non-deterministic OBDD $\mathcal{O}$ with n variables can be seen as a deterministic OBDD $\tilde{\mathcal{O}}$ working on m additional inputs Θ. Then $\mathcal{O}(x) = 1 \quad \Leftrightarrow \quad \exists\ \Theta \in \{0, 1\}^m$ s.t. $\tilde{\mathcal{O}}(x, \Theta) = 1$.

3 Expressiveness of CNF polynomials of bounded tree-width

In this section we prove our first main result. We study how expressive polynomials p_n are which are given via CNF formulas φ_n of bounded tree-width. It turns out that permanents of bounded tree-width matrices are captured by such CNF polynomials, whereas the latter in turn are captured by arithmetic formulas. Given the equivalence stated in Theorem 1 all three concepts have the same expressive power.

3.1 From permanents to clause graphs

Theorem 3. *Let $M = [m_{ij}]$ be an $n \times n$ matrix such that the corresponding directed weighted graph $G_M = (V_M, E_M)$ is of tree-width k. Then there is a CNF formula φ of tree-width $O(k^2)$ and of size polynomially bounded in n such that*

$$perm(M) \ = \ \sum_{e,\theta} \varphi(e,\theta) \cdot m^e.$$

Here, $e = \{e_{i,j}\}$ denotes variables representing the edges of $G_M, m = \{m_{i,j}\}$ denotes the entries of M and $m^e := \prod_{i,j} m_{i,j}^{e_{i,j}}$, where $m_{i,j}^{e_{i,j}} = \begin{cases} m_{i,j} \text{ if } e_{i,j} = 1 \\ 1 \quad \text{if } e_{i,j} = 0 \end{cases}$.

For every e there exists θ such that $\varphi(e,\theta) = 1$ if and only if e is a cycle cover of G_M; in this case, the corresponding θ is unique.

Moreover, the number of additional variables θ is of order $O(n)$. Finally, a tree decomposition of $I(\varphi)$ of width $O(k^2)$ can be obtained from a decomposition of G_M in time $O(n)$.

Remark 1. In the above CNF polynomial $\sum_{e,\theta} \varphi(e,\theta) \cdot m^e$ there are no monomials corresponding to θ. Formally one could introduce another block y of variables and add to each monomial m^e another factor y^θ. Then $perm(M)$ is obtained as a projection (in Valiant's sense) of a CNF-polynomial $\sum_{e,\theta} \varphi(e,\theta) \cdot m^e \cdot y^\theta$ by plugging in for each y-variable the value 1.

Proof. Let $(T, \{X_t\}_t)$ be a tree decomposition of width k for G_M. Without loss of generality T is a binary tree. The CNF formula φ to be constructed contains two blocks of variable vertices, one being the edge-variables $e_{i,j}$ of G_M and another block θ of auxiliary variables to be explained below. The tree decomposition $(T, \{X'_t\}_t)$ that we shall construct for φ uses the same underlying tree T as the tree decomposition of G_M, but the boxes X'_t will be different from the boxes X_t in the initial decomposition.

A straightforward set of clauses to describe cycle covers in G_M is the following collection:

(i) for each vertex $i \in V_M$ clauses Out_i and In_i containing as its literals all outgoing edges from and all incoming edges into i, respectively;
(ii) for each $i \in V_M$ and each pair of outgoing edges $e_{i,j}, e_{i,l}$ a clause $\neg e_{i,j} \vee \neg e_{i,l}$; similarly for incoming edges to i.

A tree decomposition of the resulting formula then is obtained from T by taking the same tree and joining in a box X'_t for every $i \in X_t$ all vertices resulting from (i) and (ii). However, due to the conditions under (ii) this might not result in a decomposition of bounded width.

To resolve this problem for each box $t \in T$ and each $i \in V_M$ we add additional variables $check^t_{i+}, check^t_{i-}$. Fix t and the subtree T_t of T that has t as its root. For any assignment of the $e_{i,j}$ indicating which edges in G_M have been chosen for a potential cycle cover a condition $check^t_{i+} = 1$ indicates that an edge starting in i has already been chosen with respect to those vertices of G_M occuring in the subtree T_t.

Further clauses are introduced to guarantee that each i finally is covered exactly once for a satisfying assignment of $\varphi(e, \theta)$, where θ is the collection of all check variables. More precisely, we proceed bottom up. Let t be a leaf of T. For every $i \in X_t$ in addition to the variable vertices $check^t_{i+}, check^t_{i-}$ introduce clause variables representing the following clauses:

(1) $\bigvee_{j \in X_t} e_{i,j} \vee \neg check^t_{i+}$;
Interpretation: if none of the $e_{i,j}$'s where chosen yet, then $check^t_{i+} = 0$.
(2) $\neg e_{i,j} \vee \neg e_{i,l}$ for all $j, l \in X_t$;
Interpretation: at most one outgoing edge covers i.
(3) $\neg e_{i,j} \vee \neg check^t_{i+}$ for all $j \in X_t$;
Interpretation: if an $e_{i,j}$ was chosen (i.e. $e_{i,j} = 1$), then $check^t_{i+} = 1$.

Analogue clause variables are added for $check^t_{i-}$.

For the box X'_t in the decomposition of $I(\varphi)$ that corresponds to box X_t of T all variable vertices $e_{i,j}, check^t_{i+}, check^t_{i-}, i, j, \in X_t$ as well as the clause variables resulting from (1)-(3) above are included. These are $O(k^2)$ many elements in X'_t. Now T' is constructed bottom up. The check variables propagate bottom up the information whether a partial assignment for those $e_{i,j}$ that already occured in a subtree can still be extended to a cycle cover of G_M. At the same time, the width of the new boxes of T' constructed will not increase too much. Suppose in T there are boxes t, t_1, t_2 such that t_1 is the left and t_2 the right child of t. Let $i \in X_t \cap X_{t_1} \cap X_{t_2}$. The case where i only occurs in two or one of the boxes is treated similarly. Assuming X'_{t_1}, X'_{t_2} already been constructed the following clauses are included in X_t :

(1′) $\bigvee_{j \in X_t \setminus \{X_{t_1} \cup X_{t_2}\}} e_{i,j} \vee check^{t_1}_{i+} \vee check^{t_2}_{i+} \vee \neg check^t_{i+}$;
Interpretation: if all new $e_{i,j}$'s and the previous check variables are 0, then the new check variable $check^t_{i+}$ is 0 as well;
(2′) $\neg x \vee \neg y$ for all $x, y \in \{e_{i,j} : j \in X_t \setminus \{X_{t_1} \cup X_{t_2}\}\} \cup \{check^{t_1}_{i+}, check^{t_2}_{i+}\}; x \neq y$
Interpretation: at most one among the old check variables and the new edge variables gets the value 1;
(3′) $\neg x \vee \neg check^t_{i+}$ for all $x \in X_t \setminus \{X_{t_1} \cup X_{t_2}\} \cup \{check^{t_1}_{i+}, check^{t_2}_{i+}\}$;
Interpretation: if one among the values $e_{i,j}$ or $check^{t_1}_{i+}, check^{t_2}_{i+}$ is 1, then $check^t_{i+} = 1$.

Again, analogue clauses are added for the ingoing edges to i. Box X'_t contains all related edge vertices $e_{i,j}$ for the new $j \in X_t \setminus \{X_{t_1} \cup X_{t_2}\}$, the six check vertices and the $O(k^2)$ many clause vertices resulting from (1')-(3').

This way $(T, \{X'_t\}_t)$ is obtained. Finally, for each $i \in T$ two new clauses containing the single literals $check^r_{i+}$ and $check^r_{i-}$, respectively, are included in that box X_r which represents the root r of the subtree of T generated by all boxes that contain i. This is to guarantee that i is covered in both directions.

Clearly, $(T, \{X'_t\}_t)$ is a binary tree with each X'_t containing at most $O(k^2)$ many vertices. Let θ denote the vector of all check variables. It is obvious from the construction that

$$\exists \theta \ \varphi(e, \theta) \Leftrightarrow e \text{ represents a cycle cover}$$

(via those $e_{i,j}$ that have value 1). Moreover, for each assignment of e^* giving a cycle cover there is precisely one assignment θ^* such that $\varphi(e^*, \theta^*)$ because e^* uniquely determines which check variables have to be assigned the value 1. Therefore

$$perm(M) \ = \ \sum_{e,\theta} \varphi(e, \theta) \cdot m^e.$$

Finally, it remains to show that $(T', \{X'_t\}_t)$ actually is a tree decomposition of the graph $I(\varphi)$. Vertices resulting from check variables at most occur in two consecutive boxes of T' and thus trivially satisfy the connectivity condition. Clause vertices related to one of the construction rules (1), (3), $(1') - (3')$ for a fixed $t \in T$ only occur in the single box X'_t. Finally, an edge variable $e_{i,j}$ occurs in a box X'_t iff both i and j occur in X_t. Thus, the fact that $(T, \{X_t\}_t)$ is a tree decomposition implies that the connectivity condition also holds for these vertices and $(T', \{X'_t\}_t)$. □

3.2 From clause graphs to arithmetic formulas

In the next step we link CNF polynomials to arithmetic formulas. More precisely, the next theorem shows the latter concept to be strong enough to capture the former.

Theorem 4. *Let $\mathbb{K}$ be a field. Let $\{\varphi_n\}_n$ be a family of CNF formulas of bounded tree-width k and with n variables, $SI(\varphi_n)$ the related signed clause graphs and $(T_n, \{X_t\}_t)$ a tree decomposition of $I(\varphi_n)$. Then there is a family $\{f_n\}_n$ of polynomials with coefficients in $\mathbb{K}$ such that $\{f_n\}_n$ can be represented by a family of polynomial size arithmetic formulas and for all $x \in \mathbb{K}^n$,*

$$f_n(x) = \sum_{z \in \{0,1\}^n} \varphi_n(z) \cdot x^z$$

The proof is based on an extension of results in [4] and shall be given in the full paper. Theorems 1, 3 and 4 imply

Theorem 5. *Let $(f_n)_{n \in \mathbb{N}}$ be a family of polynomials with coefficients in a field $\mathbb{K}$. The following properties are equivalent:*

(i) $(f_n)_{n \in \mathbb{N}}$ can be represented by a family of polynomial size arithmetic formulas.

(ii) *There exists a family $(M_n)_{n\in\mathbb{N}}$ of polynomial size, bounded tree-width matrices such that the entries of M_n are constants from $\mathbb{K}$ or variables of f_n, and $f_n = perm(M_n)$.*

(iii) *There exists a family $(\varphi_n)_{n\in\mathbb{N}}$ of CNF formulas having polynomial size in n and of bounded tree-width such that $f_n(x)$ can be expressed as the projection: $f_n(x) = \sum_{\tilde{e}} \varphi_n(\tilde{e}) \cdot z^{\tilde{e}}$. Here, projection means that the z_i's can be taken either as constants from $\mathbb{K}$ or as variables among the x_j's.* □

4 Lower bounds

Given Theorem 3 together with the efficient algorithm resulting from Theorem 4 the following question arises: How far does the approach of reducing permanent computations to computations of the form $\sum_{e,\theta} \varphi(e,\theta) \cdot m^e$ lead when φ comes from a clause graph of bounded tree-width?

More precisely, we ask whether there exist polynomial size CNF formulas $\varphi_n(e,\theta)$ of bounded tree-width such that $\varphi_n(e,\theta) = 1$ iff $e \in \{0,1\}^{n\times n}$ is a permutation matrix and for each permutation matrix e there is exactly one θ such that $\varphi(e,\theta) = 1$.

In this section we prove that such formulas do not exist. A negative answer could be expected since Theorem 4 would otherwise imply that the permanent can be represented by polynomial size arithmetic formulas. The point is that Theorem 8 below is unconditional (and does not even need the uniqueness assumption on θ). A second (conditional) result shows that when replacing tree- by clique-width a formula with the above properties does not exist unless $\#P \subseteq FP/poly$.

The unconditional statement concerning tree-width relies on two results by Ferrara et al. [3] on the one side and by Krause et al. [8] on the other. The former relates clause graphs to OBDDs (ordered binary decision diagrams), whereas the latter gives lower bounds on the size of non-deterministic OBDDs deciding the property of being a permutation matrix.

Let us first recall these results.

Theorem 6. *([3]) Let φ be a CNF formula with n variables and $P(\varphi)$ the corresponding primal graph of φ. If $P(\varphi)$ has tree-width k, then there is an OBDD representation of φ which has size polynomial in n and exponential in k.*

Due to the additional block θ of variables introduced in the transformation constructed in the proof of Theorem 3, an application of the above theorem leads to non-deterministic OBDDs.

Theorem 7. *([8, 6]) For $n \in \mathbb{N}$ define $PERM_n : \{0,1\}^{n^2} \to \{0,1\}$ to be the characteristic function for permutation matrices, i.e. for $e \in \{0,1\}^{n\times n}$ we have $PERM_n(e) = 1$ iff e is a permutation matrix. Then $2^{\Omega(n)}$ is a lower size bound for any non-deterministic OBDD computing $PERM_n$.*

The technical result below allows to apply Theorem 6 the way we need it. Note that the theorem assumes the primal graph $P(\varphi)$ to be of bounded tree-width, whereas we want to deal with the incidence graph $I(\varphi)$.

Proposition 1. *Let $\varphi = C_1 \wedge \ldots \wedge C_m$ be a CNF formula with n variables $x_1, \ldots, x_n$ such that its incidence clause graph $I(\varphi)$ has tree-width k. Then there is a CNF formula $\tilde{\varphi}(x, y)$ such that the following conditions are satisfied:*

- *each clause of $\tilde{\varphi}$ has at most $O(k)$ many literals;*
- *the primal graph $P(\tilde{\varphi})$ has tree-width $O(k)$. A tree-decomposition can be constructed in linear time from one of $I(\varphi)$;*
- *the number of variables and clauses in $\tilde{\varphi}$ is of order $O(n \cdot m)$;*
- *for all $x^* \in \{0,1\}^n$ $\varphi(x^*)$ holds true iff there exists y^* such that $\tilde{\varphi}(x^*, y^*)$. Moreover, such a y^* is unique.*

Proof. Let $(T, \{X_t\}_t)$ be a (binary) tree-decomposition of $I(\varphi)$. The construction below combines the use of check variables in the proof of Theorem 3 with the usual way of reducing a general CNF formula instance to one with bounded number of literals in each clause. Let C be a clause of φ and T_C the subtree of T induced by C. We replace C bottom up in T_C by introducing $O(n)$ many new variables and clauses. More precisely, start with a leaf box X_t of T_C. Suppose it contains k variables that occur in literals of C, without loss of generality say $x_1 \vee \ldots \vee x_k$. The case where additional clause variables occur in X_t is treated similarly. Introduce a new variable y_t together with $O(k)$ many clauses expressing the equivalence $y_t \Leftrightarrow x_1 \vee \ldots \vee x_k$. Each of the new clauses has at most $k+1$ many literals. Next, consider an inner node t of T_C having two childs t_1, t_2. Suppose $x'_1, \ldots, x'_k$ to be those variables in X_t that occur as literals in C, again without loss of generality in the form $x'_1 \vee \ldots \vee x'_k$. If y_{t_1}, y_{t_2} denote the new variables related to C that have been introduced for X_{t_1}, X_{t_2}, for X_t define a new variable y_t together with clauses expressing $y_t \Leftrightarrow y_{t_1} \vee y_{t_2} \vee x'_1 \vee \ldots \vee x'_k$. Again, there are at most $O(k)$ new clauses containing $O(k)$ literals each. Finally, if t is the root of T_C we define y_t as before and add a clause saying $y_t = 1$.

Do the same for all clauses of φ. This results in a CNF formula $\tilde{\varphi}$ which depends on $O(m \cdot n)$ additional variables y and contains $O(m \cdot n \cdot k)$ many clauses. The construction guarantees that $\varphi(x)$ iff there exists a y such that $\tilde{\varphi}(x, y)$ and in that case y is unique.

A tree-decomposition of the primal graph $P(\tilde{\varphi})$ is obtained as follows. For each occurence of a clause C in X_t of T replace the clause vertex by the newly introduced y variables related to the clause and the box X_t. In addition, for boxes X_t, X_{t_1}, X_{t_2} such that t_1, t_2 are sons of t include the variables y_{t_1}, y_{t_2} also in the upper box X_t. The x_i variables that previously occured are maintained. Since for a single box X_t at most three y_j are included for each clause, and since there are at most $k+1$ clause vertices in an original box, the tree-width of $P(\tilde{\varphi})$ is $\leq 4(k+1)$. The decomposition satisfies the requirements of a tree-decomposition since we did not change occurences of the x_i's and the only y_t-variables that occur in several boxes occur in two consecutive ones. □

As consequence we get

Theorem 8. *There does not exist a family $\{\varphi_n\}_n$ of CNF formulas $\varphi_n(e,\theta)$ such that $I(\varphi_n)$ is of bounded tree-width, the size of φ_n is polynomially bounded in n, and $\exists\theta\ \varphi_n(e,\theta)$ iff $e \in \{0,1\}^{n\times n}$ is a permutation matrix.*

Proof. Suppose to the contrary that such a family exists. Then by Proposition 1 we can assume without loss of generality that the primal graphs $P(\varphi_n)$ are of bounded tree-width as well. Moreover, the number of new variables and clauses introduced by applying the proposition remians polynomially bounded; they can formally be added to the θ variables of the statement.

Now Theorem 6 implies the existence of an OBDD representation of $\varphi_n(e,\theta)$ of polynomial size in n. Taking into account the role the θ variables are playing this OBDD is a non-deterministic polynomial-size OBDD for computing the function $PERM_n$. However, the existence of such an OBDD contradicts Theorem 7. □

The question answered negatively by Theorem 8 for tree-width can be posed as well in relation to the clique-width parameter. That is: Can the permanent function be described via CNF formulas of bounded clique-width and polynomial size? Next we relate this question to Theorem 2 b) and show that such a representation is only possible if the conjecture $\#P \not\subseteq FP/poly$ fails to be true.

Theorem 9. *Suppose there is a family $\{\varphi_n\}_n$ of CNF formulas of polynomial size such that all $I(\varphi_n)$ are of clique-width at most k for some fixed k and for each $Y \in \{0,1\}^{n^2}$ we have that $\varphi_n(Y)$ holds iff Y is a permutation matrix. Then $\#P \subseteq FP/poly$.*

Proof. Suppose $\{\varphi_n\}$ is given as in the assumption. For a matrix $X \in \{0,1\}^{n^2}$ we shall construct from φ_n and a parse-tree of it (given as non-uniform advice) another CNF formula $\psi_n^X(Y)$ of bounded clique-width together with a parse-tree for ψ_n^X such that

$$Perm(X) = \sum_{Y\in\{0,1\}^{n^2}} \psi_n^X(Y).$$

Theorem 2 b) then implies that the latter can be computed in polynomial time. Given $\#P$-completeness of the permanent function on 0-1-matrices the claim follows. The construction of ψ_n^X works as follows. We have $Perm(X) = \sum_{Y\in\{0,1\}^{n^2}} \varphi_n(Y)\cdot X^Y$, where $X^Y = \prod_{i,j} x_{i,j}^{y_{i,j}}$ and $x_{i,j}^{y_{i,j}} = \begin{cases} x_{i,j} & \text{if } y_{i,j}=1 \\ 1 & \text{if } y_{i,j}=0 \end{cases}$. We replace the monomial X^Y by the conjunctions $\bigwedge_{i,j}(x_{i,j}\vee\neg y_{i,j})$. The clause graph $I(\psi_n)$ of the CNF formula

$$\psi_n(X,Y) \equiv \varphi_n(Y) \wedge \bigwedge_{i,j}(x_{i,j}\vee\neg y_{i,j})$$

can easily be seen to have clique-width $\leq k+2$. Each time when in the clique-width construction of $I(\varphi_n)$ along the parse-tree a node $y_{i,j}$ is created, in the corresponding construction for $I(\psi_n)$ two new nodes for $x_{i,j}$ and the clause

$D_{i,j} := x_{i,j} \vee \neg y_{i,j}$ are created with an own label each. Then $D_{i,j}$ is connected to both $x_{i,j}$ and $y_{i,j}$ (respecting the necessary signs of the edges). Finally the labels for $D_{i,j}$ and $x_{i,j}$ are removed again.

Now for a fixed given matrix X we plug the values of the $x_{i,j}$ into the CNF formula $\psi_n(X,Y)$. Clauses that are satisfied by the assignment are removed. In clauses that are not satisfied by the assignment all occurences of $x_{i,j}$ literals are removed. That way a new CNF formula ψ_n^X is obtained. The clause graph $I(\psi_n^X)$ results from $I(\psi_n)$ by removing certain nodes, namely the $x_{i,j}$ as well as some clause nodes. This operation clearly does not increase the clique-width. □

The result holds as well if we allow additional variables in $\varphi_n(Y)$ as in the statement of Theorem 8. It remains an open question whether Theorem 9 can be strengthened to hold unconditionally:

Conjecture: There is no family $\{\varphi_n\}_n$ of CNF formulas of polynomial size with all $I(\varphi_n)$ of bounded clique-width such that $\varphi_n(Y) \Leftrightarrow Y$ is a permutation matrix.

We will show in the full version of this paper that Theorem 8 holds even without the polynomial size hypothesis on φ_n.

References

1. H.L. Bodlaender: *NC-algorithms for graphs with small tree-width*, In: Proc. Graph-theoretic concepts in Computer Science, LNCS, vol. 344, Springer, pp. 1–10 (1989).
2. B. Courcelle, J.A. Makowsky, U. Rotics: *Linear Time Solvable Optimization Problems on Graphs of Bounded Clique Width*, Theory of Computing Systems, vol. 33(2),pp. 125–150 (2000).
3. A. Ferrara, G. Pan, M.Y. Vardi: *Treewidth in Verification: Local vs. Global*, Proc. LPAR 2005, LNAI, vol. 3835 , Springer, pp. 489–503 (2005).
4. E. Fischer, J. Makowsky, E.V. Ravve: *Counting Truth Assignments of Formulas of Bounded Tree-Width or Clique-Width*, Disc. Appl. Mathematics 154(4), pp. 511–529 (2008).
5. U. Flarup, P. Koiran, L. Lyaudet: *On the expressive power of planar perfect matching and permanents of bounded tree-width matrices*, Proc. ISAAC 2007, LNCS, vol. 4835, Springer, pp. 124–136 (2007).
6. S. Jukna: *The effect of null-chains on the complexity of contact schemes*, In: Proc. Fundamentals of Computation Theory, LNCS, vol. 380, Springer, pp. 246–256 (1989).
7. M. Karpinski: *On the computation power of randomized branching programs*, Electronic Colloquium on Computational Complexity, Report Nr. 38 (1998).
8. M. Krause, C. Meinel, S. Waack: *Separating the eraser Turing machine classes* $L_e, NL_e, co-NL_e$ *and* P_e, Theoretical Computer Science 86, pp. 267–275 (1991).
9. M. Samer, S. Szeider: *Algorithms for Propositional Model Counting*, Proc. LPAR 2007,LNCS, vol. 4790, Springer, pp. 484–498 (2007).
10. I. Wegener: *Branching Programs and Binary Decision Diagrams: Theory and Applications*, SIAM Monographs on Discrete Mathematics and Applications (2000).