Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 2 (2003)

Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable

Stefan Szeider*

Department of Computer Science, University of Toronto,
M5S 3G4 Toronto, Ontario, Canada
szeider@cs.toronto.edu

February 7, 2003

Abstract

The deficiency of a propositional formula F in CNF with n variables and m clauses is defined as $m-n$. It is known that minimal unsatisfiable formulas (unsatisfiable formulas which become satisfiable by removing any clause) have positive deficiency. Recognition of minimal unsatisfiable formulas is NP-hard, and it was shown recently that minimal unsatisfiable formulas with deficiency k can be recognized in time $n^{\mathcal{O}(k)}$. We improve this result and present an algorithm with time complexity $\mathcal{O}(2^k n^4)$. Whence the problem is fixed-parameter tractable in the sense of R. G. Downey and M. R. Fellows, Parameterized Complexity, Springer, New York, 1999.

Our algorithm gives rise to a fixed-parameter tractable parameterization of the satisfiability problem: If the maximum deficiency over all subsets of a formula F is at most k, then we can decide in time $\mathcal{O}(2^k n^3)$ whether F is satisfiable (and we certify the decision by providing either a satisfying truth assignment or a regular resolution refutation). Known parameters for fixed-parameter tractable satisfiability decision are tree-width or related to tree-width. In contrast to tree-width (which is NP-hard to compute) the maximum deficiency can be calculated efficiently by graph matching algorithms. We exhibit an infinite class of formulas where maximum deficiency outperforms tree-width (and related parameters), as well as an infinite class where the converse prevails.

Keywords: SAT problem, minimal unsatisfiability, fixed-parameter complexity, tree-width, branch-width, clique-width, bipartite matching.

*Supported by the Austrian Science Fund (FWF) Project J2111.

ISSN 1433-8092

1 Introduction

We consider propositional formulas in conjunctive normal form (CNF) represented as sets of clauses. A formula is minimal unsatisfiable if it is unsatisfiable but omitting any of its clauses makes it satisfiable. The recognition of minimal unsatisfiable formulas is a computationally hard problem, shown to be D^P-complete by Papadimitriou and Wolfe [29].

Since for a minimal unsatisfiable formula F the number m of clauses is strictly greater than the number n of variables [1], it is natural to parameterize minimal unsatisfiable formulas with respect to the parameter

$$\delta(F) := m - n,$$

the *deficiency* of F. Following [24] we denote the class of minimal unsatisfiable formulas with deficiency k by $\mathrm{MU}(k)$.

It is known that for fixed k, formulas in $\mathrm{MU}(k)$ have short resolution refutations, and so can be recognized in nondeterministic polynomial time [23]. Moreover, deterministic polynomial time algorithms have been developed for MU(1) and MU(2), based on the very structure of formulas in the respective classes [11, 24]. Finally it was shown that for any fixed k, formulas in $\mathrm{MU}(k)$ can be recognized in polynomial time [26, 14]. The algorithm of [26] relies on the fact that formulas in $\mathrm{MU}(k)$ not only have short resolution refutations, but such refutations can even be found in polynomial time. On the other hand, the algorithm in [14] relies on the fact that the search for satisfying truth assignments can be restricted to certain assignments which correspond to matchings in bipartite graphs (we will describe this approach more detailed in Section 4). Both algorithms have time complexity $n^{\mathcal{O}(k)}$ ([14] provides the more explicit upper bound $\mathcal{O}(n^{k+1/2}l)$ for formulas of length l with n variables).

The degree of the polynomials constituting time bounds of the quoted algorithms [26, 14] strongly depends on k, since a "try all subsets of size k"-strategy is embarked. Consequently, even for small k, the algorithms become impracticable for large inputs. The theory of parameterized complexity, developed by Downey and Fellows [13], focuses on this issue. A problem is called *fixed-parameter tractable* (*FPT*) if it can be solved in time $\mathcal{O}(f(k) \cdot n^{\alpha})$ where n is the size of the instance and $f(k)$ is any function of the parameter k (the constant α is independent from k).

In this paper we show that $\mathrm{MU}(k)$ is fixed-parameter tractable, stating an algorithm with time complexity $\mathcal{O}(2^k n^4)$. The obtained speedup relies on the interaction of two concepts, *maximum deficiency* and *expansion*, both stemming from graph theory (the graph theoretic concepts carry over to formulas by means of *incidence graphs*, see Section 4).

1.1 Maximum deficiency and expansion

The maximum deficiency of a formula F is defined as $\delta^*(F) = \max_{F' \subseteq F} \delta(F)$ (thus always $\delta^*(F) \geq 0$). This parameter was first considered for formulas by Franco and Van Gelder [16]. For minimal unsatisfiable formulas, deficiency and maximum deficiency agree. Formulas with maximum deficiency 0, called "matched formulas" in [16], are always satisfiable (for generalizations, see [35]). The maximum deficiency of a formula can be considered as its distance from being a matched formula and provides a measure of its hardness.

We call a formula F *q-expanding* if for every nonempty set X of variables of F there are at least $|X| + q$ clauses C of F such that some variable of X occurs in C. It is known that minimal unsatisfiable formulas are 1-expanding [1] and that every formula contains some equisatisfiable 1-expanding subset; moreover, such subset is unique and can be found efficiently [25, 14]. Furthermore, if each literal of a formula $F \in \mathrm{MU}(k)$, $k \geq 2$, is contained in at least 2 clauses, then F is 2-expanding [23, 24]. We extend the various quoted results and pinpoint the importance of the notion of q-expansion for satisfiability decision.

Let $F[x = \varepsilon]$ denote the formula obtained from F by instantiating the variable x with a truth value $\varepsilon \in \{0, 1\}$ and applying the usual simplifications (see Section 2.2 for exact definitions). It is known that in general $\delta^*(F[x = \varepsilon]) \leq \delta^*(F) + 1$ holds, and if F is 1-expanding, then even $\delta^*(F[x = \varepsilon]) \leq \delta^*(F)$ (see [25]). Moreover by *simultaneous* instantiation of $\delta^*(F)$ variables one can reduce any satisfiable formula F to a formula with maximum deficiency 0 ([14], see Theorem 1 below). Thus, if k is fixed, then trying all possible instantiations of k variables can be carried out in polynomial time, but the degree of the polynomial strongly depends on k. Hence the known approach does not yield a fixed-parameter tractable algorithm.

Key for our improvement is an efficient algorithm which reduces a given formula to an equisatisfiable formula F such that

> *instantiating any variable of F with any truth value* 0 *or* 1 *decreases the maximum deficiency;*

we call a formula F with this property *δ^*-critical.* We show that if every literal of a 2-expanding formula F occurs in at least two clauses, then F is δ^*-critical.

We present a variant of the Davis-Logemann-Loveland (DLL) algorithm applying splittings (branchings from F to $F[x = 0]$ and $F[x = 1]$) to δ^*-critical formulas only. Consequently, the maximum deficiency decreases at each splitting, and so the height of the resulting search tree is bounded by the maximum deficiency of the input formula. A careful analysis of the reductions applied at the nodes of the search tree gives the following time complexity (the hidden constant does not depend on k).

> *Satisfiability of formulas with n variables and maximum deficiency k can be decided in time $\mathcal{O}(2^k n^3)$.*

The presented algorithm provides *certificates* for its decision; i.e., if the input formula is satisfiable, then it outputs a *satisfying truth assignment*, otherwise a *regular resolution refutation.*

To decide whether a formula F belongs to $\mathrm{MU}(k)$, we first check $\delta(F) = \delta^*(F) = k$; if this holds true, then we check whether F is unsatisfiable, and whether $F \setminus \{C\}$ is satisfiable for all clauses C of F. This can be accomplished by $n + k + 1$ applications of the above result. Hence we get the following.

> *Minimal unsatisfiable formulas with n variables and $n + k$ clauses can be recognized in time $\mathcal{O}(2^k n^4)$.*

1.2 Fixed-parameter tractable parameterizations of SAT

Several recursively defined parameterizations of the satisfiability problem are known which allow satisfiability decision in time $n^{\mathcal{O}(k)}$ if the considered parameter is bounded by k; see [30] for references. Such time complexity does not constitute fixed-parameter tractability (however, it appears that no fixed-parameter *in*tractability results are known; the W-hierarchy of [13] provides a means for intractability results).

Fixed-parameter tractability can be achieved by bounding the *tree-width* of the considered formulas. Tree-width is usually defined for graphs, but can be applied to formulas via incidence or primal graphs; we refer to the former as "incidence tree-width" and to the latter as "primal tree-width," see Section 7 for details. Formulas with bounded incidence tree-width have also bounded primal tree-width, but the converse does not hold; hence incidence tree-width is the more general parameter.

Gottlob, et al. [19] show that satisfiability of formulas with bounded primal tree-width is fixed-parameter tractable, applying general methods developed in [19] for constraint satisfaction problems. By means of tree-decompositions, formulas can be transformed into acyclic constraint satisfaction problems (CSPs) which in turn can be solved efficiently. *Branch-width* is another tree-width related parameter; it agrees with tree-width up to a multiplicative constant. Alekhnovich and Razborov [2] show fixed-parameter tractable satisfiability decision for formulas with bounded branch-width. The algorithm developed in [2] is an extension of the algorithm of Robertson and Seymour [32] for computing branch-decompositions. Alekhnovich and Razborov also discuss the relation of branch-width and the "resolution width" of Ben-Sasson and Wigderson [4]. The algorithms of [19] and [2] are suitable for use in praxis.

A variant of *Courcelle's Theorem* (see, e.g., [13]) allows to achieve fixed-parameter tractability even for larger classes of formulas: In [9] it is shown that if a "k-expression" for the directed incidence graph of a formula F is given (thus its *directed clique-width* is at most k, see Section 7.2 for definitions), then satisfying assignments of F can be counted in time $\mathcal{O}(f(k) \cdot l)$. Although it is not known whether graphs of clique-width k can be recognized in polynomial time for fixed $k \geq 4$, the result of [9] yields fixed-parameter tractable satisfiability for formulas with bounded incidence tree-width (for, graphs of bounded tree-width have also bounded clique-width [10]).

Both tree-with and branch-width are NP-hard to compute [33, 3] (in contrast to maximum deficiency, which can be computed efficiently by matching algorithms); however, for fixed k it can be decided efficiently whether a given graph has tree-width (or branch-width) k.

How is maximum deficiency related to the quoted parameters? We show the following.

> *1. There are formulas with bounded primal tree-width (implying bounded incidence tree-width) but arbitrary high maximum deficiency.*
>
> *2. Conversely, there are formulas with bounded maximum deficiency but arbitrary high incidence clique-width (implying arbitrary high incidence tree-width and primal tree-width).*

Thus tree-width (resp. branch-width) and maximum deficiency are in a certain sense *incomparable*.

Finally, we mention some fixed-parameter results for a certain subclass PIF_2 of so-called "pure implicational formulas" (PIF_2 contains propositional formulas whose only connective is the implication, and where each variable occurs at most twice; negations are not allowed, but a formula may contain the constant $\mathbf{f}$ (falsum)). In [15] it is shown that satisfiability of PIF_2 formulas with k occurrences of the symbol $\mathbf{f}$ can be decided in time $\mathcal{O}(k^k n^2)$; thus satisfiability is fixed-parameter tractable. The time complexity has been recently improved to $\mathcal{O}(3^k n^2)$, $k \geq 4$, by means of dynamic programming techniques [21]. Although any CNF formula F can be translated into an equisatisfiable PIF_2 formula P (see [20]), the number of $\mathbf{f}$ occurrences in P always exceeds the maximum deficiency of F, at least if the translation of [20] is used.

A more in-depth study of the fixed-parameter complexity of parameterizations of the satisfiability problem and their relative strength is carried out in a forthcoming paper [34].

2 Notation and preliminaries

2.1 Formulas

We assume an infinite supply of propositional *variables*. A *literal* is a variable x or a complemented variable $\overline{x}$; if $y = \overline{x}$ is a literal, then we write $\overline{y} = x$; we also use the notation $x^1 = x$ and $x^0 = \overline{x}$. For a set S of literals we write $\overline{S} = \{\, \overline{x} : x \in S \,\}$; S is *tautological* if $S \cap \overline{S} \neq \emptyset$. A *clause* is a finite non-tautological set of literals; the empty clause is denoted by $\square$. A finite set of clauses is a *CNF formula* (or *formula*, for short). The *width* of a clause is its cardinality, and the width $w(F)$ of a formula is the width of a largest clause of F (or 0 if F is empty). The *length* of a formula F is the sum of widths of its clauses. For a literal x we write $\#_x(F)$ for the number of clauses of F which contain x. A literal x is a *pure literal* of F if $\#_x(F) \geq 1$ and $\#_{\overline{x}}(F) = 0$; x is a *singular literal* of F if $\#_x(F) = 1$ and $\#_{\overline{x}}(F) \geq 1$.

A literal x *occurs* in a clause C if $x \in C \cup \overline{C}$; $\mathsf{var}(C)$ denotes the set of variables which occur in C. For a formula F we put $\mathsf{var}(F) = \bigcup_{C \in F} \mathsf{var}(C)$. Let F be a formula and $X \subseteq \mathsf{var}(F)$. We denote by F_X the set of clauses of F in which some variable of X occurs; i.e.,

$$F_X := \{\, C \in F : \mathsf{var}(C) \cap X \neq \emptyset \,\}.$$

$F_{(X)}$ denotes the formula obtained from F_X by restricting all clauses to literals over X, i.e.,

$$F_{(X)} := \{\, C \cap (X \cup \overline{X}) : C \in F_X \,\}.$$

2.2 Truth assignments

A *truth assignment* is a map $\tau : X \to \{0,1\}$ defined on some set X of variables; we write $\mathsf{var}(\tau) = X$. If $\mathsf{var}(\tau)$ is just a singleton $\{x\}$ with $\tau(x) = \varepsilon$, then we denote τ simply by $x = \varepsilon$. We call τ *empty* if $\mathsf{var}(\tau) = \emptyset$. A truth assignment τ is *total* for a formula F if $\mathsf{var}(\tau) = \mathsf{var}(F)$. For $x \in \mathsf{var}(\tau)$ we define $\tau(\overline{x}) = 1 - \tau(x)$. For a truth assignment τ and a formula F, we put

$$F[\tau] = \{\, C \setminus \tau^{-1}(0) : C \in F,\ C \cap \tau^{-1}(1) = \emptyset \,\};$$

i.e., $F[\tau]$ denotes the result of instantiating variables according to τ and applying the usual simplifications. A truth assignment τ *satisfies* a clause if the clause contains some literal x with $\tau(x) = 1$; τ satisfies a formula F if it satisfies all clauses of F (i.e., if $F[\tau] = \emptyset$). A formula is *satisfiable* if it is satisfied by some truth assignment; otherwise it is *unsatisfiable*. A formula is *minimal unsatisfiable* if it is unsatisfiable and every proper subset of F is satisfiable. We say that formulas F and F' are *equisatisfiable* (in symbols $F \equiv_{sat} F'$) if either both are satisfiable or both are unsatisfiable.

A truth assignment α is *autark* for a formula F if $\mathsf{var}(\alpha) \subseteq \mathsf{var}(F)$ and α satisfies $F_{\mathsf{var}(\alpha)}$; that is, α satisfies all affected clauses. Note that the empty assignment is autark for every formula, and that any total satisfying assignment of a formula is autark. The key feature of autark assignments is the following observation of [28].

Lemma 1. *If α is an autark assignment of a formula F, then $F[\alpha]$ is an equisatisfiable subset of F.*

Thus, in particular, minimal unsatisfiable formulas have no autark assignments except the empty assignment. If x^ε is a pure literal of F, $(x, \varepsilon) \in \mathsf{var}(F) \times \{0,1\}$, then clearly $x = \varepsilon$ is an autark assignment (and $F[x = \varepsilon]$ can be obtained from F by the "pure literal rule").

2.3 Resolution and Davis-Putnam resolution.

If C_1, C_2 are clauses and $C_1 \cap \overline{C_2} = \{x\}$ holds for some literal x, then the clause $C = (C_1 \cup C_2) \setminus \{x, \overline{x}\}$ is called the *resolvent* of C_1 and C_2.

Let F be a formula. A sequence $C_1, \dots, C_n$ is a *resolution derivation from F* if for each $i \in \{1, \dots, n\}$ either $C_i \in F$ ("C_i is an axiom"), or C_i is the resolvent of C_j and $C_{j'}$ for some $1 \le j < j' \le i-1$ (" C_j and $C_{j'}$ are the parents of C_i"). In general, a clause in a resolution derivation may have different "histories"; that is, the clause may have different pairs of parents, and it may be both, an axiom and a derived clause. However, we tacitly assume that some arbitrary but fixed history is given. A resolution derivation is a *resolution refutation* if it contains the empty clause.

A *thread* of a resolution derivation R is a subsequence $D_1, \ldots, D_k$ of R such that for each $i = 2, \ldots, k$, D_{i-1} is a parent of D_i in R. A resolution derivation R is *regular* if for each thread $D_1, \ldots, D_k$ of R we have $(D_1 \cap D_k) \subseteq D_i$, $i = 1, \ldots, k$. It is well known that a formula is unsatisfiable if and only if it has a regular resolution refutation.

Consider a formula F and a literal x of F. We obtain a formula F' from F by adding all possible resolvents w.r.t. x, and by removing all clauses in which x occurs. We say that F' is obtained from F by *Davis-Putnam resolution* and we write $\mathrm{DP}_x(F) = F'$. It is well known that $F \equiv_{sat} \mathrm{DP}_x(F)$. In fact, the so called Davis-Putnam procedure successively eliminates variables in this manner until either the empty formula or a formula which contains the empty clause is obtained. The Davis-Putnam procedure can be considered as a special case of regular resolution (cf. [37]).

Usually, $\mathrm{DP}_x(F)$ contains more clauses than F, however, if $\#_x(F) \leq 1$ or $\#_{\overline{x}}(F) \leq 1$, then clearly $|\mathrm{DP}_x(F)| < |F|$. In the sequel we will focus on $\mathrm{DP}_x(F)$ where x is a singular literal of F.

3 Graph theoretic tools

All considered graphs are finite and simple (no multiple edges or self-loops). We denote a bipartite graph G by the triple (V_1, V_2, E) where V_1 and V_2 give the bipartition of the vertex set of G, and E denotes the set of edges of G. An edge between $v_1 \in V_1$ and $v_2 \in V_2$ is denoted as ordered pair (v_1, v_2). $N_G(X)$ denotes the set of all vertices y adjacent to some $x \in X$ in G, i.e., $N_G(X)$ is the (open) neighborhood of X. For graph theoretic terminology not defined here, the reader is referred to [12].

A *matching* M of a graph G is a set of independent edges of G; i.e., distinct edges in M have no vertex in common. A vertex of G is called *matched by* M, or M*-matched*, if it is incident with some edge in M; otherwise it is *exposed by* M, or M*-exposed*. A matching M of G is a *maximum matching* if there is no matching M' of G with $|M'| > |M|$. A maximum matching of a bipartite graph $G = (V_1, V_2, E)$ can be found in time

$$\mathcal{O}(|V_1 \cup V_2|^{1/2} \cdot |E|)$$

by the algorithm of Hopcroft and Karp [22], see also [27].

Let M be a matching of a graph G. A path P in G is called M*-alternating* if edges of P are alternately in and out of M; an M-alternating path is M*-augmenting* if both of its ends are M-exposed. If P is an M-augmenting path, then

$$M' := (M \setminus E(P)) \cup (E(P) \setminus M),$$

the *symmetric difference* of M and the set of edges $E(P)$ which lie on P, is a matching of size $|M| + 1$. In this case we say that M' is obtained from M by *augmentation*. Conversely, by a well-known result of Berge [5], a matching M is a maximum matching if there is no M-augmenting path.

In our considerations we often have to deal with bipartite graphs for which an "almost" maximum matching is given. In such cases it would be inefficient to construct a maximum matching from scratch, since a maximum matching can be obtained by just a few augmentations:

Lemma 2. *Let $G = (V_1, V_2, E)$ be a bipartite graph and M a matching of G which exposes s_1 vertices of V_1 and s_2 vertices of V_2. Then we can obtain a maximum matching M' of G in time $\mathcal{O}(\min(s_1, s_2) \cdot (|E| + |V_1 \cup V_2|))$.*

Proof. Alternating paths are just directed paths in the bipartite digraph obtained from G by orienting the edges in M from V_1 to V_2, and orienting the edges in $E \setminus M$ from V_2 to V_1. Hence we can find an M-augmenting path by breadth first search starting from the set of M-exposed vertices in V_2. Thus, an M-augmenting path can be found in time $\mathcal{O}(|E| + |V_1 \cup V_2|)$. Since each augmentation decreases the number of exposed vertices in V_1 and in V_2, the lemma follows. □

We say that a bipartite graph $G = (V_1, V_2, E)$ is *q-expanding* if $q \geq 0$ is an integer such that $|N_G(X)| \geq |X| + q$ holds for every nonempty set $X \subseteq V_1$. Note that by Hall's Theorem, G is 0-expanding if and only if G has a matching of size $|V_1|$ (see [27]).

Let M be a matching of G. We define $R_{G,M}$ to be the set of vertices of G which can be reached from an M-exposed vertex in V_2 by some M-alternating path (see Figure 1 for an illustration). By means of this concept, we can easily obtain the basic graph theoretic results needed for our considerations:

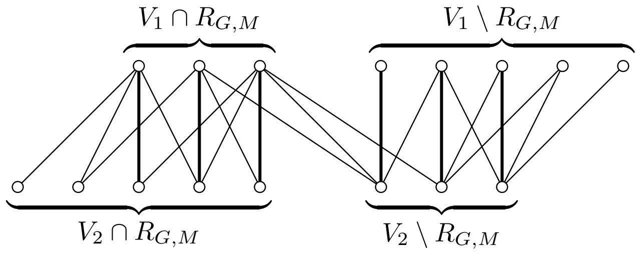

Figure 1: A bipartite graph G with a maximum matching M (indicated by bold lines).

Lemma 3. *Given a maximum matching M of a bipartite graph $G = (V_1, V_2, E)$, $V = V_1 \cup V_2$, then the following statements hold true.*

(i) *$R_{G,M}$ can be obtained in time $\mathcal{O}(|V| + |E|)$.*

(ii) *No edge joins vertices in $V_1 \setminus R_{G,M}$ with vertices in $V_2 \cap R_{G,M}$; no edge in M joins vertices in $V_1 \cap R_{G,M}$ with vertices in $V_2 \setminus R_{G,M}$.*

(iii) *All vertices in $V_1 \cap R_{G,M}$ and $V_2 \setminus R_{G,M}$ are matched vertices.*

(iv) *If G is not 0-expanding, then* $|V_1 \setminus R_{G,M}| > |N_G(V_1 \setminus R_{G,M})|$.

(v) $|V_2 \cap R_{G,M}| - |N_G(V_2 \cap R_{G,M})| = |V_2| - |M|$.

(vi) *If $R_{G,M} \neq \emptyset$, then $R_{G,M}$ induces a 1-expanding subgraph of G.*

Proof. Let S_i denote the set of M-exposed vertices in V_i, $i = 1, 2$.

(i) We consider G as a directed graph as in the proof of Lemma 2; now $R_{G,M}$ contains just the vertices which can be reached from vertices in S_2 by a directed path. Hence $R_{G,M}$ can be obtained by breadth-first-search in time $\mathcal{O}(|V|+|E|)$.

(ii) Suppose there is some edge $(u, w) \in E$ with $u \in V_1 \setminus R_{G,M}$ and $w \in V_2 \cap R_{G,M}$. If $w \in S_2$, then $u \in R_{G,M}$, a contradiction; hence $w \notin S_2$. By definition of $R_{G,M}$, there is an M-alternating path P from some $s \in S_2$ to w; the last edge of P is traversed from V_1 to V_2, hence it belongs to M; consequently $(u, w) \notin M$. Now P, u is an M-alternating path from s to u, and so $u \in R_{G,M}$, again a contradiction. Thus there is no edge between vertices in $V_1 \setminus R_{G,M}$ and $V_2 \cap R_{G,M}$. A similar argument shows that no edge of M joins vertices in $V_1 \cap R_{G,M}$ with vertices in $V_2 \setminus R_{G,M}$.

(iii) Consider any vertex $u \in V_1 \cap R_{G,M}$ and let P be some M-alternating path from some $s \in S_2$ to u (P exists by definition of $R_{G,M}$). It follows that u must be M-matched, since otherwise P would be M-augmenting, contradicting the maximality of M. On the other hand, vertices in $V_2 \setminus R_{G,M}$ are M-matched since $S_2 \subseteq R_{G,M}$ by definition.

(iv) By (ii) and (iii), M matches the vertices in $(V_1 \setminus R_{G,M}) \setminus S_1$ to vertices in $V_2 \setminus R_{G,M}$ and vice versa. Hence $|V_1 \setminus R_{G,M}| - |S_1| = |(V_1 \setminus R_{G,M}) \setminus S_1| = |V_2 \setminus R_{G,M}| \leq |N_G(V_1 \setminus R_{G,M})|$. If G is not 0-expanding, then $S_1 \neq \emptyset$ follows by Hall's Theorem.

(v) By (ii) and (iii), M matches the vertices in $V_1 \cap R_{G,M}$ to vertices in $(V_2 \cap R_{G,M}) \setminus S_2$ and vice versa. Hence $|S_2| = |V_2 \cap R_{G,M}| - |V_1 \cap R_{G,M}| = |V_2 \cap R_{G,M}| - |N_G(V_2 \cap R_{G,M})|$. In turn, $|S_2| = |V_2| - |M|$ by definition of $R_{G,M}$.

(vi) Choose any nonempty set $X = \{u_1, \dots, u_n\} \subseteq V_1 \cap R_{G,M}$. We have to show that $|N_G(X) \cap R_{G,M}| \geq n+1$. Let $w_1, \dots, w_n \in V_2$ such that $(u_i, w_i) \in M$ for $i = 1, \dots, n$. By (ii) above, $\{w_1, \dots, w_n\} \subseteq R_{G,M}$. Choose any $x \in X$. Since $x \in R_{G,M}$, there is some M-alternating path P which starts in some $s \in S_2$ and ends in x. Let (u, w) be the first edge occurring on P with $u \in X$. Since P traverses (u, w) from w to u, $(u, w) \notin M$ and so $w \notin \{w_1, \dots, w_n\}$. However, $w \in N_G(X) \cap R_{G,M}$; hence $|N_G(X) \cap R_{G,M}| \geq |\{w, w_1, \dots, w_n\}| = n + 1$ follows. □

We note in passing that we get the same set $R_{G,M}$ for every maximum matching M of G; this follows from the fact that every maximum matching M' matches the vertices in $V_1 \cap R_{G,M}$ (these vertices belong to every minimum vertex cover [1]).

Let $G = (V_1, V_2, E)$ be a bipartite graph. The *deficiency* of G is defined as $\delta(G) := |V_2| - |N_G(V_2)|$ (if V_1 contains no isolated vertices, then $\delta(G) = |V_2| - |V_1|$). The *maximum deficiency* of G is defined as $\delta^*(G) := \max_{Y \subseteq V_2} |Y| - |N_G(Y)|$. Note that $\delta^*(G) \geq 0$ follows by taking $Y = \emptyset$. The next lemma, a direct consequence of Lemma 3(v), is well-known (see, e.g., [27]). It shows that $\delta^*(G)$ can be calculated efficiently.

Lemma 4. *A maximum matching of a bipartite graph $G = (V_1, V_2, E)$ exposes exactly $\delta^*(G)$ vertices of V_2.*

Lemma 5. *Let $G = (V_1, V_2, E)$ be a 1-expanding bipartite graph and let Y be a proper subset of V_2. Then $|Y| - |N_G(Y)| \leq \delta^*(G) - 1$.*

Proof. Choose a vertex $w \in V_2 \setminus Y$. Since $G - w$ is 0-expanding, there is a maximum matching M of G which exposes w. Let S_2 be the set of M-exposed vertices of V_2. By the preceding lemma, $|S_2| = \delta^*(G)$. Since $w \in S_2 \setminus Y$, $|Y \cap S_2| \leq \delta^*(G) - 1$ follows. However, every vertex in $Y \setminus S_2$ is matched to some vertex in $N_G(Y)$, thus $|N_G(Y)| \geq |Y \setminus S_2|$. Consequently $|Y| - |N_G(Y)| \leq |Y| - |Y \setminus S_2| = |Y \cap S_2| \leq \delta^*(G) - 1$. ☐

4 Matchings and expansion of formulas

To every formula F we associate a bipartite graph $I(F)$ whose vertices are the clauses and variables of F, and where a variable is adjacent to the clauses in which it occurs; that is, $I(F) = (\mathsf{var}(F), F, E(F))$ with $(x, C) \in E(F)$ if and only if $x \in \mathsf{var}(C)$; see Figure 2 for an example[1]. We call $I(F)$ the *incidence graph* of F. Note that $|E(F)|$ equals the length of F.

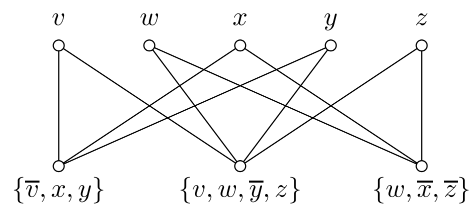

Figure 2: The incidence graph $I(F)$ of the formula $F = \{\{\overline{v}, x, y\}, \{v, w, \overline{y}, z\}, \{w, \overline{x}, \overline{z}\}\}$.

By means of this construction, concepts for bipartite graphs apply directly to formulas. In particular, we will speak of q-expanding formulas, matchings of formulas, and the deficiency and maximum deficiency of formulas. That is, a formula F is *q-expanding* if and only if $|F_X| \geq |X| + q$ for every nonempty set $X \subseteq \mathsf{var}(F)$. The *deficiency* of a formula F is $\delta(F) = |F| - |\mathsf{var}(F)|$; its *maximum deficiency* is $\delta^*(F) = \max_{F' \subseteq F} \delta(F')$. If $\mathsf{var}(F) = \emptyset$, then F is q-expanding for any q, and we have $\delta^*(F) = |F| \leq 1$. Note that 1-expanding formulas are exactly the "matching lean" formulas of [25].

In terms of formulas, the above Lemmas 4 and 5 read as follows (see [25] for an alternate proof of Lemma 7).

[1] If we label edges (x, C) of $I(F)$ with $+$ or $-$ for $x \in C$ or $\overline{x} \in C$, respectively, then we get a "formula graph" as considered in [14]; cf. also the "directed incidence graphs" in Section 7.2.

Lemma 6. *A maximum matching M of a formula F exposes exactly $\delta^*(F)$ clauses.*

Lemma 7. *If F is a 1-expanding formula and F' a proper subset of F, then we have $\delta^*(F') \le \delta^*(F) - 1$.*

A matching M of a formula F gives rise to a partial truth assignment τ_M as follows. For every $(x, C) \in M$ we put $\tau_M(x) = 1$ if $x \in C$, and $\tau_M(x) = 0$ if $\overline{x} \in C$. If $|M| = |F|$, then τ_M evidently satisfies F; thus we have the following (this observation has been made in [36] and [1]).

Lemma 8. *If a formula F has a matching which matches all clauses, i.e., if $\delta^*(F) = 0$, then F is satisfiable.*

Formulas F with maximum deficiency 0 are termed *matched formulas* in [16] (the probabilistic analysis of [16] shows that, in a certain sense, matched formulas are more numerous than formulas belonging to several well-known classes, including extended-, renamable-, and q-Horn formulas, CC-balanced formulas, and single lookahead unit resolution (SLUR) formulas). For example, the formula F of Figure 2 is matched, since all clauses of F are matched by the matching $M = \{(v, \{\overline{v}, x, y\}), (w, \{v, w, \overline{y}, z\}), (x, \{w, \overline{x}, \overline{z}\})\}$. M gives rise to the satisfying truth assignment τ_M with $\tau_M(v) = 0$, $\tau_M(w) = 1$, $\tau_M(x) = 0$.

The next lemma is essentially [14, Lemma 10].

Lemma 9. *Given a formula F of length l and a maximum matching M of F, then we can find in time $\mathcal{O}(l)$ an autark assignment α of F such that $F[\alpha]$ is 1-expanding; $M \cap E(F[\alpha])$ is a maximum matching of $F[\alpha]$.*

Proof. We apply the construction of Lemma 3 to the incidence graph $I(F)$. Thus F splits into formulas $F_1 = F \cap R_{I(G),M}$ and $F_2 = F \setminus F_1$. We consider $M_i = M \cap E(F_i)$, $i = 1, 2$. Consequently, $\alpha := \tau_{M_2}$ is an autark assignment of F with $F[\alpha] = F_1$. Moreover, by Lemma 3, $F[\alpha]$ is 1-expanding and M_1 is a maximum matching of $F[\alpha]$. □

In view of Lemma 1 we get the following corollary (see also [1, 16]).

Corollary 1. *Minimal unsatisfiable formulas are 1-expanding. Hence $\delta^*(F) = \delta(F)$ holds for minimal unsatisfiable formulas.*

The following result of [14] extends Lemma 8 to formulas with positive maximum deficiency.

Theorem 1 (Fleischner, et al. [14]). *A formula F is satisfiable if and only if $F[\tau]$ is a matched formula for some truth assignment τ with $|\mathsf{var}(\tau)| \le \delta^*(F)$.*

Thus, if $\delta^*(F) \le k$ for some fixed constant k, then we can decide satisfiability of F by checking a polynomial number of truth assignments. The time analysis of [14] gives the following estimation.

Theorem 2 (Fleischner, et al. [14]). *Let F be a formula of length l on n variables and let k be any fixed integer. If $\delta^*(F) \leq k$, then we can decide satisfiability of F in time $\mathcal{O}(n^{k+1/2}l)$.*

If the maximum deficiency is at most one, then we get the following.

Lemma 10. *Let F be a formula of length l on n variables. If $\delta^*(F) \leq 1$, then we can find a satisfying truth assignment of F (if it exists) in time $\mathcal{O}(nl)$.*

Proof. First we obtain a maximum matching M of F in time $\mathcal{O}(\sqrt{n}l)$ using the Hopcroft-Karp algorithm (see Section 3). Since $\delta^*(F) \leq 1$, M exposes at most one clause (Lemma 6). We apply Lemma 9 and obtain an autark assignment α of F and a matching M' of $F' = F[\alpha]$ in time $\mathcal{O}(l)$. If $F' = \emptyset$, then α satisfies F and we are done. Now assume $F' \neq \emptyset$. Since F' is 1-expanding, $\delta^*(F') = 1$ follows.

Consider $(x, \varepsilon) \in \mathsf{var}(F') \times \{0, 1\}$ and $F'' := F'[x = \varepsilon]$. There is at most one clause $C_0 \in F''$ with $(x, C_0 \cup \{x^{\varepsilon-1}\}) \in M'$ and at most one clause $C_1 \in F'' \cap F'$ which is M'-exposed (possibly $C_0 = C_1$). For all other clauses C of F'' we can choose $y_C \in \mathsf{var}(F'')$ such that $(y_C, C) \in M'$ or $(y_C, C \cup \{x^{\varepsilon-1}\}) \in M'$. Thus, the edges (y_C, C) form a matching M^* of F'' which exposes at most two clauses (C_0, C_1). Hence we need at most two augmentations to extend M^* to a maximum matching M'' of F''. Thus M'' can be obtained in time $\mathcal{O}(l)$. If M'' matches all clauses of F'' (i.e., if $\delta^*(F'') = 0$), then $\tau_{M''}$ satisfies F'', and consequently $\tau := \alpha \cup \tau_{M''} \cup \{(x, \varepsilon)\}$ satisfies F.

By Theorem 1, F' is satisfiable if and only if $\delta^*(F'[x = \varepsilon]) = 0$ for some $(x, \varepsilon) \in \mathsf{var}(F') \times \{0, 1\}$. Thus the claimed time complexity follows. ☐

5 The main reductions

5.1 δ^*-critical formulas

We call a formula F *δ^*-critical* if $\delta^*(F[x = \varepsilon]) \leq \delta^*(F) - 1$ holds for every $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$. The objective of this section is to reduce a given formula F efficiently to a δ^*-critical formula F' ensuring $\delta^*(F') \leq \delta^*(F)$ and $F \equiv_{sat} F'$. Thus δ^*-critical formulas constitute a "problem kernel" in the sense of [13].

First we pinpoint a sufficient condition for formulas being δ^*-critical.

Lemma 11. *2-expanding formulas without pure or singular literals are δ^*-critical.*

Proof. Let F be a 2-expanding formula without pure or singular literals, $|F| = m$. Choose any $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$ and consider $F' = F[x = \varepsilon]$. We can write $F = \{C_1, \ldots, C_m\}$ such that for integers r, s, t with $1 \leq r \leq s \leq t \leq m$ we have

$$\begin{aligned} x^{\varepsilon} \in C_j \quad &\Leftrightarrow \quad 1 \leq j \leq r; \\ x^{1-\varepsilon} \in C_j \quad &\Leftrightarrow \quad r+1 \leq j \leq t; \\ x^{1-\varepsilon} \in C_j \text{ and } C_j \setminus \{x^{1-\varepsilon}\} \in F \quad &\Leftrightarrow \quad r+1 \leq j \leq s; \end{aligned}$$

we have $r \geq 2$ and $t \geq r+2$. We put $D_j := C_j \setminus \{x^{1-\varepsilon}\}$ and get

$$F' = \{D_{s+1}, \ldots, D_m\} = \{D_{s+1}, \ldots, D_t, C_{t+1}, \ldots, C_m\}.$$

We choose a maximum matching M of F which exposes C_1 and C_2. (Such matching exists: since F is 2-expanding, $F_2 = F \setminus \{C_1, C_2\}$ is 0-expanding; and since F has no pure or singular literals, $\mathsf{var}(F_2) = \mathsf{var}(F)$. Thus F_2 has a maximum matching M with $|M| = |\mathsf{var}(F_2)| = |\mathsf{var}(F)|$; such M is a maximum matching of F.) The matching M gives rise to a (possible non-maximum) matching M' of F' by setting

$$M' = \{\, (y, D_j) : (y, C_j) \in M,\ y \neq x,\ s+1 \leq j \leq m \,\}.$$

We show that the number of M'-exposed vertices of F' is strictly smaller than the number of M-exposed vertices of F. That is, $|I'| < |I|$ for $I = \{\, 1 \leq j \leq m : C_j \text{ is } M\text{-exposed } \,\}$ and $I' = \{\, s+1 \leq j \leq m : D_j \text{ is } M'\text{-exposed } \,\}$.

Let $j_x \in \{1, \ldots, t\}$ be the unique integer such that $(x, C_j) \in M$. If $j_x \leq s$, then $|I \cap \{s+1, \ldots, m\}| = |I'|$; otherwise, if $j_x > s$, then $|I \cap \{s+1, \ldots, m\}| = |I'| - 1$. Thus $|I \cap \{s+1, \ldots, m\}| \geq |I'| - 1$ holds in any case. On the other hand, since $1, 2 \in I$ by the choice of M, we have $|I \cap \{1, \ldots, s\}| \geq 2$. Consequently

$$|I| = |I \cap \{1, \ldots, s\}| + |I \cap \{s+1, \ldots, m\}| \geq 2 + |I'| - 1 \geq |I'| + 1.$$

By means of Lemma 6 we conclude $\delta^*(F) = |I| > |I'| \geq \delta^*(F')$. Thus F is δ^*-critical as claimed. □

5.2 First step: eliminating pure and singular literals

Consider a sequence $S = (F_0, M_0), \ldots, (F_q, M_q)$ where F_i is a formula and M_i is a maximum matching of F_i, $0 \leq i \leq q$. We call S a *reduction sequence* (starting from (F_0, M_0)) if for each $i \in \{1, \ldots, q\}$ one of the following holds:

- $F_i = F_{i-1}[\alpha_i]$ for some nonempty autark assignment α_i of F_{i-1}.
- $F_i = \mathrm{DP}_{x_i}(F_{i-1})$ for a singular literal x_i of F_{i-1}.

Note that $\mathsf{var}(F_i) \subsetneq \mathsf{var}(F_{i-1})$, hence $q \leq |\mathsf{var}(F_0)|$. Evidently, F_0 and F_q are equisatisfiable. Furthermore, we have the following.

Lemma 12. *Let $(F_0, M_0), \ldots, (F_q, M_q)$ be a reduction sequence. Any satisfying truth assignment τ_q of F_q can be extended to a satisfying truth assignment τ_0 of F_0; any regular resolution refutation R_q of F_q can be extended to a regular resolution refutation R_0 of F_0.*

Proof. We put $I = \{\, 1 \leq i \leq q : F_i = F_{i-1}[\alpha_i] \,\}$, and $I' = \{\, 1 \leq i \leq q : F_i = \mathrm{DP}_{x_i}(F_{i-1}) \,\}$; $I \cap I' = \emptyset$ and $I \cup I' = \{1, \ldots, q\}$.

If τ_q is a satisfying assignment of F_q, then we get a satisfying assignment of F_0 by setting $\tau_0 = \tau_q \cup \bigcup_{i \in I} \alpha_i$.

We obtain inductively a regular resolution refutation R_0 of F_0 as follows. Let R_i be a regular resolution refutation of F_i for some $i \in \{1, \dots, q\}$. If $i \in I$, then R_i is trivially a regular resolution refutation of F_{i-1}, since $F_i \subseteq F_{i-1}$. Now assume $i \in I'$. Let $C_1, \dots, C_k$ be the clauses of F_{i-1} which contain x or $\overline{x}$. Every axiom C of R_i which is not contained in F_{i-1} is the resolvent of clauses $C_j, C_{j'}$, $1 \leq j, j' \leq k$. Thus $C_1, \dots, C_k, R_i$ is a regular resolution refutation of F_{i-1}. □

In the proof of the next lemma we have to proceed very carefully, since the time complexities stated in our main results depend directly on it.

Lemma 13. *Let F_0 be a formula on n variables with $\delta^*(F_0) \leq n$, and let M_0 be a maximum matching of F_0. We can construct in time $\mathcal{O}(n^3)$ a reduction sequence $(F_0, M_0), \dots, (F_q, M_q)$, $q \leq n$, such that exactly one of the following holds.*

(i) $\delta^*(F_q) \leq \delta^*(F_0) - 1$*;*

(ii) $\delta^*(F_q) = \delta^*(F_0)$*, F_q is 1-expanding and has no pure or singular literals.*

Proof. We construct the reduction sequence inductively; assume that we have already constructed $(F_0, M_0), \dots, (F_{i-1}, M_{i-1})$ for some $i \geq 1$. We obtain F_i applying the first of the following cases which is appropriate.

Case 1: F_{i-1} is not 1-expanding. We apply Lemma 9 and obtain a nonempty autark assignment α of F_{i-1}. We put $F_i := F_{i-1}[\alpha]$ and $M_i := M_{i-1} \cap E(F_i)$.

Case 2: F_{i-1} has a pure literal x^ε, $(x, \varepsilon) \in \mathsf{var}(F_{i-1}) \times \{0, 1\}$. We remove the clauses which contain x^ε from F_{i-1} and get an equisatisfiable proper subset F_i. (Note that $F_i = F_{i-1}[x = \varepsilon]$ and that $x = \varepsilon$ is an autark assignment of F_{i-1}; cf. the discussion in Section 2.2.) Since F_{i-1} is 1-expanding, $\delta^*(F_i) \leq \delta^*(F_{i-1}) - 1$ follows by Lemma 7. The matching $M'_i = M_{i-1} \cap E(F_i)$ is possibly not a maximum matching of F_i, but it exposes not more clauses of F_i than M_{i-1} exposes clauses of F_{i-1}; thus we need at most $\delta^*(F_{i-1})$ augmentations to get a maximum matching M_i of F_i (cf. Lemma 6). We put $q = i$ and do not extend the reduction sequence any further.

Case 3: F_{i-1} has a singular literal x^ε, $(x, \varepsilon) \in \mathsf{var}(F_{i-1}) \times \{0, 1\}$. We put $F_i = \mathrm{DP}_x(F_{i-1})$. For integers $1 \leq s \leq t \leq m$ we can write

$$\begin{aligned} F_{i-1} &= \{C_1, \dots, C_m\}, \\ F_i &= \{D_{s+1}, \dots, D_m\} = \{D_{s+1}, \dots, D_t, C_{t+1}, \dots, C_m\}, \end{aligned}$$

such that $x^\varepsilon \in C_1$, $x^{1-\varepsilon} \in C_j$ for $2 \leq j \leq t$, and D_j is the resolvent of C_1 and C_j for $j = s+1, \dots, t$ (that is, for $j \in \{2, \dots, s\}$, the resolvent of C_1 and C_j is either tautological, or it is already contained in F_i). We may assume, w.l.o.g., that $(y_1, C_1) \in M_{i-1}$ for some variable $y_1 \in \mathsf{var}(F_{i-1})$ (for, if C_1 is M_{i-1}-exposed, we consider the matching $M_{i-1} \setminus \{(x, C_{j_x})\} \cup \{(x, C_1)\}$) instead; j_x is the unique integer in $\{1, \dots, t\}$ with $(x, C_{j_x}) \in M_{i-1}$).

We define the matching

$$M'_i = \{\, (y, D_i) : (y, C_i) \in M,\ y \neq x,\ s+1 \leq i \leq m \,\}.$$

If there is some $j \in \{s+1, \dots, t\}$ such that C_j is M_{i-1}-matched but D_j is M_i'-exposed, then $(x, C_j) \in M_{i-1}$ follows; and so, since y_1 is M_i'-exposed and since $y_1 \in \mathsf{var}(D_j) = (\mathsf{var}(C_1) \cup \mathsf{var}(C_j)) \setminus \{x\}$, we conclude that $M_i'' = M_i' \cup \{(y_1, D_j)\}$ is a matching of F_i which exposes at most $\delta^*(F_{i-1})$ clauses. Otherwise, if such j does not exist, we simply put $M_i'' = M_i'$. In any case, M_i'' exposes at most $\delta^*(F_{i-1})$ clauses of F_i, and so $\delta^*(F_i) \leq \delta^*(F_{i-1})$ follows by Lemma 6.

Case 3a: $s = 1$; (i.e., $|F_i| = |F_{i-1}| - 1$). We have $\mathsf{var}(F_i) = \mathsf{var}(F_{i-1}) \setminus \{x\}$, and consequently, the matching M_i'' is a maximum matching of F_i; we put $M_i = M_i''$.

Case 3b: $s > 1$; (i.e., $|F_i| < |F_{i-1}| - 1$). Since M_i'' exposes at most $\delta^*(F_{i-1})$ clauses, we need at most $\delta^*(F_{i-1})$ augmentations to obtain a maximum matching M_i of F_i. We put $q = i$, and do not extend the reduction sequence any further.

We show that in Case 3b even $\delta^*(F_i) \leq \delta^*(F_{i-1}) - 1$ holds. Since F_{i-1} is 1-expanding, we can choose for every clause $C \in F_{i-1}$ some maximum matching of F_{i-1} which exposes C. In particular, we can assume that C_2 is M_{i-1}-exposed (and simultaneously, by the same argument as above, that C_1 is M_{i-1}-matched). Then, however, the matching M_i'' constructed above exposes at most $\delta^*(F_{i-1}) - 1$ clauses of F_i. Hence $\delta^*(F_i) \leq \delta^*(F_{i-1}) - 1$ follows by Lemma 6.

In each of the above cases, the construction of F_i can be carried out in time $\mathcal{O}(n^2)$; in Cases 1 and 3a this also suffices to construct M_i. In Cases 2 and 3b we have to perform at most $\delta^*(F_{i-1}) \leq n$ augmentations; thus, by Lemma 2, time $\mathcal{O}(n^3)$ suffices for Cases 2 and 3b. Since $q \leq n$, and since Cases 2 and 3b occur at most once (we stop the construction of the reduction sequence in both cases), the claimed time complexity follows. □

5.3 Second step: reduction to 2-expanding formulas

By the results of the previous section we can efficiently reduce a given formula until we end up with a formula which is 1-expanding and which has no pure or singular literals. In this section we present further reductions which yield to δ^*-critical formulas.

Theorem 3 below is due to Lovász and Plummer [27, Theorem 1.3.6] and provides the basis for an efficient test for q-expansion (see Lemma 14). We state the theorem using the following construction.

From a bipartite graph $G = (V_1, V_2, E)$, $x \in V_1$, and $q \geq 1$, we obtain the bipartite graph G_{qx} by adding new vertices $x_1, \dots, x_q$ to V_1 and adding edges such that the new vertices have exactly the same neighbors as x; that is,

$$G_{qx} = (V_1 \cup \{x_1, \dots, x_q\}, V_2, E \cup \{\, x_i y : xy \in E \,\}).$$

Theorem 3 (Lovász and Plummer [27]). *A 0-expanding bipartite graph $G = (V_1, V_2, E)$ is q-expanding if and only if for every $x \in V_1$ the graph G_{qx} is 0-expanding.*

Lemma 14. *Given a bipartite graph $G = (V_1, V_2, E)$ and a maximum matching M of G. For every fixed integer $q \geq 0$, deciding whether G is q-expanding and, if G is not q-expanding, finding a "witness set" $X \subseteq V_1$ with $|N_G(X)| < |X|+q$, can be performed in time $\mathcal{O}(|V_1| \cdot |E| + |V_2|)$.*

Proof. We may assume that G has no isolated vertices (for, if $x \in V_1$ is isolated, then G is not 0-expanding and $\{x\}$ is a witness set; on the other hand, we can delete any isolated vertex in V_2 without affecting q-expansion). We compute the set of vertices $R_{G,M}$ (recall the definition in Section 3). If G is not 0-expanding, $V_1 \setminus R_{G,M}$ is a witness set by Lemma 3(iv), and we are done. Hence we assume that G is 0-expanding; i.e., $|M| = |V_1|$.

For each vertex $x \in V_1$ we perform the following procedure. We obtain the graph $G_{qx} = (V_1', V_2', E')$ with $V_1' = V_1 \cup \{x_1, \ldots, x_q\}$ and $V_2' = V_2$. Note that the given matching M is also a matching of G_{qx}, and that $x_1, \ldots, x_q$ are exactly the M-exposed vertices of V_1'. We extend M to a maximum matching M' of G_{qx} by at most q augmentations. Now G_{qx} is 0-expanding if and only if $|M'| = |V_1'| = |V_1| + q$.

Assume that G_{qx} is not 0-expanding; i.e., V_1' contains M'-exposed vertices. As above, we obtain the set $R_{G_{qx},M'}$ and put $X' := V_1' \setminus R_{G_{qx},M'}$. Lemma 3(iv) yields $|N_{G_{qx}}(X')| < |X'|$. Since X' contains M'-exposed vertices, and since every M'-exposed vertex of V_1' belongs to $\{x_1, \ldots, x_q\}$ by construction, $\{x_1, \ldots, x_q\} \cap X' \neq \emptyset$ follows. We show that $\{x, x_1, \ldots, x_q\} \subseteq X'$ holds. Suppose to the contrary that for some $x', x'' \in \{x, x_1, \ldots, x_q\}$ we have $x' \in X'$ and $x'' \notin X'$. Since $x'' \in R_{G_{qx},M'}$, G_{qx} contains an M'-alternating path P which starts in some M'-exposed vertex of V_2' and ends in x''. For the last edge (x'', y) of P, $y \in R_{G_{qx},M'} \cap V_1'$ follows. Since $N_{G_{qx}}(x') = N_{G_{qx}}(x'')$ by construction of G_{qx}, we have $(y, x') \in E'$. This, however, is impossible by Lemma 3(ii). Hence indeed $\{x, x_1, \ldots, x_q\} \subseteq X'$. We put $X := X' \setminus \{x_1, \ldots, x_q\}$. Since $N_{G_{qx}}(X') = N_G(X)$, we have $|N_G(X)| < |X'| = |X| - q$; thus X is a witness set.

If we perform the above construction for all $x \in V_1$, we either end up with a witness set $X \subseteq V_1$, $|N_G(X)| < |X| + q$, or we may conclude by means of Theorem 3 that G is q-expanding.

It remains to estimate the required time. The preprocessing (identification of isolated vertices and the construction of $R_{G,M}$) can certainly be carried out in time $\mathcal{O}(|V_1| + |V_2| + |E|)$; see Lemma 3(i). This estimation is dominated by the claimed time complexity. For each $x \in V_1$ we construct G_{qx}, perform at most q augmentations, and construct $R_{G_{qx},M'}$. In view of Lemmas 2 and 3(i), and since q is a fixed constant, each of these three tasks can be carried out in time $\mathcal{O}(|V_1| + |V_2| + |E|)$. Moreover, after the preprocessing, G has no isolated vertices, thus $|V_1| + |V_2| = \mathcal{O}(|E|)$. Hence we need at most time $\mathcal{O}(|V_1| \cdot |E|)$ to process all vertices in V_1; this estimation is dominated by the claimed time complexity as well. $\square$

Lemma 15. *Let F be a 1-expanding formula without pure or singular literals, and let $X \subseteq \mathsf{var}(F)$ with $|F_X| \leq |X|+1$. Then $F \setminus F_X \equiv_{sat} F$ and $\delta^*(F \setminus F_X) \leq \delta^*(F) - 1$.*

Proof. Since F is 1-expanding, $|F_X| = |X| + 1$ follows. We show that $F_{(X)}$ is satisfiable. Because F is 1-expanding, every clause $C \in F$ is exposed by some maximum matching M_C of F. Any maximum matching of F matches the variables in X to clauses in F_X; hence, for every $C \in F_X$, the assignment τ_{M_C} (see Section 4 for the definition) satisfies $F_X \setminus \{C\}$. Every proper subset G of $F_{(X)}$ is a subset of $(F_X \setminus \{C\})_{(X)}$ for some $C \in F_X$; thus τ_{M_C} satisfies G. We conclude that $F_{(X)}$ is either satisfiable or minimal unsatisfiable.

If $F_{(X)}$ is minimal unsatisfiable, then $|F_{(X)}| \geq |X| + 1$ by Corollary 1; on the other hand, $|F_{(X)}| \leq |F_X| = |X| + 1$; hence the deficiency of $F_{(X)}$ is exactly 1. In [11] it is shown that every minimal unsatisfiable formula with deficiency 1 different from $\{\square\}$ has a singular literal; however, every singular literal of $F_{(X)}$ is also a singular of F, but F has no singular literals by assumption. Thus $F_{(X)}$ cannot be minimal unsatisfiable, and must therefore be satisfiable. Since a satisfying total assignment α of $F_{(X)}$ is a nonempty autark assignment of F with $F[\alpha] = F \setminus F_X$, we conclude by Lemma 1 that $F \equiv_{sat} F \setminus F_X$. Using Lemma 7, we get $\delta^*(F \setminus F_X) \leq \delta^*(F) - 1$. $\square$

Lemma 16. *Let F be a 1-expanding formula without pure or singular literals, $m = |F|$, $n = |\mathsf{var}(F)|$, and let M be a maximum matching of F. We need at most $\mathcal{O}(n^2 m)$ time to decide whether F is 2-expanding, and if it is not, to find an autark assignment α of F with $\delta^*(F[\alpha]) \leq \delta^*(F) - 1$ and a maximum matching M' of $F[\alpha]$.*

Proof. We apply Lemma 14 to the incidence graph of F. Thus $\mathcal{O}(n^2 m)$ time suffices to decide whether F is 2-expanding, and if it is not, to find a set $X \subseteq \mathsf{var}(F)$ with $|F_X| = |X| + 1$. Note that $\delta^*(F_{(X)}) \leq 1$, and by the preceding lemma, $F_{(X)}$ is satisfiable. By means of Lemma 10 we can find a satisfying total assignment α of $F_{(X)}$ in time $\mathcal{O}(|X|^2 \cdot (|X| + 1)) \leq \mathcal{O}(n^2 m)$. Since α is a nonempty autark assignment of F, $\delta^*(F[\alpha]) \leq \delta^*(F) - 1$ follows (Lemmas 1 and 7). We consider the matching $M' = M \cap E(F[\alpha])$. Since M matches every variable $x \in X$ to some clause $C \in F_X$, and since $|F_X| - |X| = 1$, it follows that M matches at most one variable $y \in \mathsf{var}(F[\alpha]) \subseteq \mathsf{var}(F) \setminus X$ to a clause $C \in F_X$. Consequently, at most one variable of $F[\alpha]$ is M'-exposed. Therefore, we need at most one augmentation to obtain a maximum matching M' of $F[\alpha]$; this requires $\mathcal{O}(nm)$ time (Lemma 2). Whence the lemma is shown true. $\square$

We summarize the results of this section.

Theorem 4. *Let F_0 be a formula on n variables with $\delta^*(F_0) \leq n$, and let M_0 be a maximum matching of F_0. We can obtain in time $\mathcal{O}(n^3)$ a reduction sequence $(F_0, M_0), \ldots, (F_q, M_q)$, $q \leq n$, such that exactly one of the following holds:*

(i) $\delta^*(F_q) \leq \delta^*(F_0) - 1$;

(ii) $\delta^*(F_q) = \delta^*(F_0)$ *and F_q is δ^*-critical.*

Proof. First we construct a reduction sequence $S = (F_0, M_0), \ldots, (F_p, M_p)$ by means of Lemma 13. If $\delta^*(F_p) \leq \delta^*(F_0) - 1$, then S is the required reduction sequence and we are done; hence assume $\delta^*(F_p) = \delta^*(F_0)$. Now F_p is 1-expanding and has no pure or singular literals (Lemma 13). We apply Lemma 16 to F_p and M_p. If F_p is 2-expanding, then F_p is δ^*-critical by Lemma 11; thus S is the required reduction sequence and we are done as well. If, however, F_p is not 2-expanding, then Lemma 16 provides an autark assignment α of F_p with $\delta^*(F_p[\alpha]) \leq \delta^*(F_p) - 1$ and a maximum matching M' of $F_p[\alpha]$. The concatenation $S, (F_p[\alpha], M')$ is the required reduction sequence. The claimed time complexity follows directly from Lemmas 13 and 16. $\square$

6 Proof of the main results

It remains to combine the results of the preceding sections to gain our main results.

Theorem 5. *Satisfiability of formulas with n variables and maximum deficiency k can be decided in time $\mathcal{O}(2^k n^3)$. The decision is certified by a satisfying truth assignment or a regular resolution refutation of the input formula.*

Proof. Let F be any given formula with $|\mathsf{var}(F)| = n$, $|F| = m$, and $\delta^*(F) = k$. Consequently, $m \leq n + k$, and the length l of F is at most $nm \leq n(n+k)$.

By trivial reasons, we can decide satisfiability of F in time $\mathcal{O}(2^n)$, i.e., by constructing a binary tree T, a "DLL tree": The root is labeled by F, and each vertex which is labeled by a formula F' with $\mathsf{var}(F) \neq \emptyset$ has two children, labeled by $F'[x = 0]$ and $F'[x = 1]$, respectively, for some $x \in \mathsf{var}(F')$. The leaves of F are labeled by $\emptyset$ or $\{\square\}$. F is satisfiable if and only if some leaf w is labeled by $\emptyset$. In this case, the path from the root to w determines a satisfying truth assignment of F. On the other hand, if F is unsatisfiable, then all leaves must be labeled by $\{\square\}$. Now T gives rise to a regular resolution refutation R of F by means of the following (well known) construction:

The formula $\{\square\}$ has the trivial resolution refutation $R = \square$. Let F be a formula and $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$. If R_ε is a regular resolution refutation of $F[x = \varepsilon]$, then adding $x^{1-\varepsilon}$ to some of the clauses in R_ε yields a regular resolution derivation R'_ε of $\{x^{1-\varepsilon}\}$ from F. The concatenation $R'_0, R'_1, \square$ is a regular resolution refutation of F.

Hence the theorem holds trivially if $k \geq n$; next we consider the non-trivial case $k < n$.

We apply the Hopcroft-Karp algorithm to the incidence graph of F and find a maximum matching M of F in time $\mathcal{O}(l\sqrt{n+m}) \leq \mathcal{O}(n^3)$.

We are going to construct a search tree T of height $\leq k$ such that each vertex v of T has at most 2 children and is labeled by a reduction sequence S_v. If $S_v = (F_0, M_0), \ldots, (F_r, M_r)$, then we write $\mathsf{first}(v) = F_0$ and $\mathsf{last}(v) = F_r$.

We construct T inductively as follows. We start with a root vertex v_0, and we label it by a reduction sequence constructed by means of Theorem 4, starting from (F, M). Assume that we have already constructed some search tree T'. If $\mathsf{var}(\mathsf{last}(v)) = \emptyset$ for all leaves v of T', then we halt. Otherwise, we pick a leaf v

of T' with $\mathsf{var}(\mathsf{last}(v)) \neq \emptyset$; let $S_v = (F_0, M_0), \ldots, (F_r, M_r)$. By Theorem 4, one of the following holds:

(i) $\delta^*(F_r) \leq \delta^*(F_0) - 1$;

(ii) $\delta^*(F_r) = \delta^*(F_0)$ and F_r is δ^*-critical.

In the first case we add a single child v' to v, and we label v' by a reduction sequence starting from (F_r, M_r); i.e., $\mathsf{first}(v') = F_r$.

In the second case we pick a variable $x \in \mathsf{var}(F_r)$ and obtain the formulas $F' = F_r[x = 0]$ and $F'' = F_r[x = 1]$. We construct maximum matchings M' and M'' of F' and F'', respectively. As above, M' and M'' can be obtained by the Hopcroft-Karp algorithm in time $\mathcal{O}(n^3)$ (in practice it may be more efficient to construct M' and M'' from M_r as in the proof of Lemma 11). We add two vertices v' and v'' as children of v to T'. We label v' and v'' by a reduction sequence starting from (F', M') and (F'', M''), respectively; i.e., $\mathsf{first}(v') = F'$ and $\mathsf{first}(v'') = F''$.

For any pair of vertices v, v', if v' is a child of v, then $\delta^*(\mathsf{first}(v')) \leq \delta^*(\mathsf{first}(v)) - 1$. Hence the construction terminates and we get a tree T of height at most $\delta^*(F) = k$. Hence T has at most $2^k - 1$ vertices. It follows now from Theorem 4 that time $\mathcal{O}(2^k n^3)$ suffices for constructing T.

If v is a leaf of T, then deciding satisfiability of $\mathsf{last}(v)$ is trivial, since $\mathsf{last}(v) = \emptyset$ or $\mathsf{last}(v) = \{\Box\}$. However, since $\mathsf{first}(v) \equiv_{sat} \mathsf{last}(v)$ holds for all vertices v of T, and since for a non-leaf v, $\mathsf{last}(v)$ is satisfiable if and only if $\mathsf{first}(v')$ is satisfiable for at least on of its children v', we can inductively read off from T whether F is satisfiable. That is, similarly to the DLL tree considered above, F is satisfiable if and only if $\mathsf{last}(v)$ is satisfiable for at least one leaf v of T. Moreover, Lemma 12 allows us to obtain from T a satisfying truth assignment (if F is satisfiable) or a regular resolution refutation (if F is unsatisfiable) similarly as from a DLL tree as described above. Thus the theorem is shown true. $\Box$

Theorem 6. *Minimal unsatisfiable formulas with n variables and $n+k$ clauses can be recognized in time $\mathcal{O}(2^k n^4)$.*

Proof. If $k \geq n$, then the theorem holds by trivial reasons, since we can enumerate all total truth assignments of F in time $\mathcal{O}(2^n)$; hence we assume $k < n$. Let $F = \{C_1, \ldots, C_m\}$, $m = n + k < 2n$. If F is minimal unsatisfiable, then it is 1-expanding and so $\delta^*(F) = \delta(F) = k$ (see Corollary 1). This necessary condition can be checked efficiently (Lemma 9). Furthermore, we have to check whether F is unsatisfiable, and whether $F_i := F \setminus \{C_i\}$ is satisfiable for all $i \in \{1, \ldots, m\}$. This can be accomplished by applying $m+1$ times the algorithm of Theorem 5. We have verified that F is 1-expanding, hence $\delta^*(F_i) \leq k - 1$ by Lemma 7. Thus the over-all time complexity $\mathcal{O}((m + 1)2^k n^3) \leq \mathcal{O}(2^k n^4)$ follows. $\Box$

7 Maximum deficiency vs. tree-width

Tree-width, a popular parameter for graphs, was introduced by Robertson and Seymour in their series of papers on graph minors, see, e.g., [7] for references. Let G be a graph, $T = (V, E)$ a tree, and χ a labeling of the vertices of T by sets vertices of G. Then (T, χ) is a *tree-decomposition* of G if the following conditions hold:

(T1) Every vertex of G belongs to $\chi(t)$ for some $t \in V$;

(T2) for every edge (v, w) of G there is some vertex t of T such that $v, w \in \chi(t)$;

(T3) for any vertices t_1, t_2, t_3 of T, if t_2 lies on a path from t_1 to t_3, then $\chi(t_1) \cap \chi(t_3) \subseteq \chi(t_2)$.

The *width* of a tree-decomposition (T, χ) is the maximum $|\chi(t)| - 1$ over all vertices t of T. The tree-width $tw(G)$ of G is the minimum width over all tree-decompositions of G. Note that trees have tree-width 1 (the only purpose of "-1" in the definition of tree-width is to make this statement true).

For fixed $k \geq 1$, deciding whether a given graph has tree-width at most k (and computing a tree-decomposition of width $\leq k$, if it exists) can be done efficiently (in quadratic time by Robertson and Seymour [31], and even in linear time by Bodlaender [6]). Computing the tree-width of a given graph, however, is an NP-hard problem [3].

The following lemma is well-known (and not difficult to show).

Lemma 17. *Let (T, χ) be a tree-decomposition of a graph G and let $K \subseteq V(G)$ be a set of vertices which induces a complete subgraph in G. Then $K \subseteq \chi(t)$ for some vertex t of T.*

The *primal graph* $P(F)$ of a formula F is the graph whose vertices are the variables of F, two variables are joined by an edge if and only if both variables occur together in a clause. We will consider tree-width of primal graphs as well as tree-width and incidence graphs of formulas; for a formula F we call $tw(I(F))$ the *incidence tree-width* and $tw(P(F))$ the *primal tree-width* of F.

7.1 Tree-width of primal graphs

In [19] the following is shown.

Theorem 7 (Gottlob, et al. [19]). *Satisfiability of formulas with bounded primal tree-width is fixed-parameter tractable.*

The proof of this result relies on the fact that a formula can be considered as a constraint satisfaction problem (CSP) over the universe $\{0, 1\}$; in [19] it is shown that CSPs over a fixed universe and of fixed tree-width can be "fixed-parameter transformed" into an equivalent *acyclic* CSP. Since it is well-known that acyclic CSPs can be solved in linear time, Theorem 7 follows.

The next lemma follows directly from Lemma 17 (recall from Section 2.1 that $w(F) = \max_{C \in F \cup \{\square\}} |C|$).

Lemma 18. *$w(F) \leq tw(P(F)) + 1 \leq |\mathsf{var}(F)|$ holds for every formula F.*

Next we construct formulas with small maximum deficiency and large primal tree-width.

Theorem 8. *For every $k \geq 1$ there are minimal unsatisfiable formulas F such that $\delta^*(F) = 1$ and $tw(P(F)) = k$.*

Proof. We consider formulas used by Cook ([8], see also [37]) for deriving exponential lower bounds for the size of tableaux refutations. Let k be any positive integer and consider the complete binary tree T of height $k+1$, directed from the root to the leaves. Let $v_1, \dots, v_m$, $m = 2^{k+1}$, denote the leaves of T. For each non-leaf v of T we take a new variable x_v, and we label the outgoing edges of v by x_v and $\overline{x_v}$, respectively. For each leaf v_i of T we obtain the clause C_i consisting of all labels occurring on the path from the root to v_i. Consider the formula $F = \{C_1, \dots, C_m\}$. It is not difficult to see that F is minimal unsatisfiable (in fact, it is "strongly minimal unsatisfiable" in the sense of [1]). Moreover, since $|\mathsf{var}(F)| = 2^{k+1} - 1$, we have $\delta^*(F) = \delta(F) = 1$. Since $|C_i| = k + 1$, $tw(P(F)) \geq k$ follows from Lemma 18. On the other hand, $tw(P(F)) \leq k$, since we can define a tree-decomposition (T, χ) of width k for F as follows. For each leaf v_i of T we put $\chi(v) = \mathsf{var}(C_i)$; for each non-leaf w we define $\chi(w)$ as the set of variables x_v such that v lies on the path from the root of T to w (in particular, $x_w \in \chi(w)$). □

Theorem 9. *For every $k \geq 1$ there are minimal unsatisfiable formulas H such that $\delta^*(H) = k$ and $tw(P(H)) \leq 2$.*

Proof. We consider the formula $H := \bigcup_{i=0}^{k} H_i$ where $H_0 = \{\{z_0\}\}$, $H_k = \{\{\overline{z_{k-1}}\}\}$, and for $i = 1, \dots, k-1$,

$$H_i := \{\{\overline{z_{i-1}}, x_i, y_i\}, \{\overline{x_i}, y_i\}, \{x_i, \overline{y_i}\}, \{\overline{x_i}, \overline{y_i}, z_i\}\}.$$

It follows by induction on k that $\delta(H) = k$ and that H is minimal unsatisfiable. Hence $\delta^*(H) = k$. We define a tree-decomposition (T, χ) of H taking the path $v_0, \dots, v_k$ for T and setting $\chi(v_i) = \mathsf{var}(H_i)$. The width of this tree-decomposition is at most 2, hence $tw(H) \leq 2$ follows. □

Results similar to Theorems 8 and 9 can be obtained for *branch-width* as considered for formulas by Alekhnovich and Razborov [2].

7.2 Tree-width of incidence graphs

Since the maximum deficiency is defined in terms of incidence graphs, we will compare it with incidence tree-width.

The next result (which seems to be well-known, [18]) indicates that incidence tree-width is the more general parameter than primal tree-width.

Lemma 19. *For every formula F we have*

$$tw(I(F)) \leq \max(tw(P(F)), w(F)) \leq tw(P(F)) + 1.$$

Proof. Let (T, χ) be a tree-decomposition of $P(F)$ of width k. By Lemma 17 we can choose for every clause $C \in F$ some vertex t_C of T such that $\mathsf{var}(C) \subseteq \chi(t_C)$. We obtain a tree T' from T by adding for very clauses $C \in F$ a new vertex t'_C and the edge (t_C, t'_C). Finally, we extend the labeling χ to T' defining $\chi(t'_C) = \mathsf{var}(C) \cup \{C\}$. We can verify that (T', χ) is a tree-decomposition of $I(F)$ by checking the conditions (T1)–(T3). Since $|\chi(t'_C)| = |C| + 1$, the width of (T', χ) is the maximum of k and $w(F)$. However, Lemma 17 also implies that $tw(P(F)) \geq w(F) - 1$, hence the result is shown true. □

On the other hand, there are formulas whose primal graphs have arbitrary high tree-width and whose incidence graphs are trees (i.e., have tree-width 1); take, for example, the minimal unsatisfiable formula $\{\{x_1, \ldots, x_n\}, \{\overline{x_1}\}, \ldots, \{\overline{x_n}\}\}$.

The question rises whether Theorem 7 can be generalized to incidence tree-width. Below we answer this question positively, deploying a variant of Courcelle's Theorem.

First we give some definitions taken from [10]. Let k be a positive integer. A *k-graph* G is a graph whose vertices are labeled by integers from $\{1, \ldots, k\}$. We consider any graph as k-graph with all vertices labeled by 1. We call the k-graph consisting of exactly one vertex v labeled by $i \in \{1, \ldots, k\}$ an *initial k-graph* and denote it by $i(v)$. Let $\mathcal{C}(k)$ denote the class of k-graphs which can be constructed from initial k-graphs by means of the following three operations.

(C1) If $G, H \in \mathcal{C}(k)$ and $V(G) \cap V(H) = \emptyset$, then the union of G and H, denoted by $G \oplus H$, belongs to $\mathcal{C}(k)$.

(C2) If $G \in \mathcal{C}(k)$ and $i, j \in \{1, \ldots, k\}$, then the k-graph $\rho_{i \to j}(G)$ obtained from G by changing the labels of all vertices which are labeled by i to j belongs to $\mathcal{C}(k)$.

(C3) If $G \in \mathcal{C}(k)$, $i, j \in \{1, \ldots, k\}$, and $i \neq j$, then the k-graph $\eta_{i,j}(G)$ obtained from G by connecting all vertices labeled by i with all vertices labeled by j belongs to $\mathcal{C}(k)$.

The *clique-width* $cw(G)$ of a graph G is the smallest integer k such that $G \in \mathcal{C}(k)$. Constructions of a k-graph using the above steps (C1)–(C3) can be represented by *k-expressions*, terms composed of $i(v)$, $G \oplus H$, $\eta_{i,j}(G)$ and $\rho_{i \to j}(G)$. Thus, a k-expression certifies that a graph has clique-width $\leq k$. For example, the 4-expression

$$\rho_{2 \to 1}(\eta_{1,2}(2(y) \oplus \rho_{2 \to 1}(\eta_{1,2}(2(x) \oplus \rho_{2 \to 1}(\eta_{1,2}(1(v) \oplus 2(w))))))$$

represents a construction of the complete graph K_4 on $\{v, w, x, y\}$, hence $cw(K_4) \leq 2$). In view of this example it is easy to see that any complete graph has has clique-width ≤ 2, hence a result similar to Lemma 17 does not hold for clique-width.

The above definitions apply also to *directed graphs* except that in construction (C3) the added edges are directed from label i to label j. Thus, we can consider k-expressions for a directed graph D and we can define the *directed clique-width* $dcw(D)$ of D as the smallest k such that D has a k-expression. Let D be a directed graph and G_D its *underlying undirected graph* (i.e., G is obtained from D by "forgetting" the direction of edges and by identifying possible parallel edges); since every k-expression for D is also a k-expression for G_D, $cw(G_D) \leq dcw(D)$ follows.

The next result is due to Courcelle and Olariu [10] (see also [9]).

Theorem 10 (Courcelle and Olariu [10]). *Let D be a directed graph and (T, χ) a width k' tree-decomposition of G_D. Then we can obtain in polynomial time a k-expression for D with $k \leq 2^{2k'+1} + 1$. Thus $dcw(D) \leq 2^{2tw(G_D)+1} + 1$.*

From the incidence graph $I(F)$ of a formula F we obtain a directed graph $I_d(F)$ by orienting edges (x, C) from C to x if $x \in C$, and from x to C if $\overline{x} \in C$; we call $I_d(F)$ the *directed incidence graph* of F.

Courcelle, et al. [9] show the following.

Theorem 11 (Courcelle, et al. [9]). *Given a formula F of length l and a k-expression for $I_d(F)$ (thus $dcw(I_d(G)) \leq k$). Then the number of satisfying total truth assignments of F can be counted in time $\mathcal{O}(f(k) \cdot l)$ where f is some function which does not depend on F.*

The proof of this result is based on a variant of Courcelle's Theorem: If k is a constant and a k-expression for a directed graph D is given, then statements formulated in a certain fragment of monadic second-order logic (MS_1) can be evaluated on D in linear time. Satisfiability of F can be formulated as an MS_1 statement on $I_d(F)$: F is satisfiable if and only if there exists a set of variables V_0 such that for every clause $C \in F$, $I_d(F)$ contains either an edge directed from C to some variable in V_0, or it contains an edge directed from some variable in $\mathsf{var}(F) \setminus V_0$ to C.

Before we can apply Theorem 11 to a given formula we have to find a k-expression for its directed incidence graph; though, it is not known whether k-expressions can be found in polynomial time for constants $k \geq 4$ (see, e.g., [9]). Anyway, in view of Theorem 10, we can use the previous result to improve on Theorem 7 by considering incidence graphs instead of primal graphs.

Corollary 2. *Satisfiability of formulas with bounded incidence tree-width is fixed-parameter tractable.*

Note, however, that a practical use of Theorem 11 is very limited because of large hidden constants and high space requirements; cf. the discussion in [9]. Nevertheless, it seems to be feasible to develop algorithms which decide satisfiability directly by examining a given tree-decomposition of the incidence graph, without calling on Courcelle's Theorem.

Next we show a result similar to Theorem 8.

Theorem 12. *For every $k \geq 1$ there are formulas F such that $\delta^*(F) = 1$ and $dcw(I_d(F)) \geq cw(I(F)) \geq k$.*

Proof. Let k be a positive integer and let q be the smallest odd integer with $q \geq \max(3, k-1)$. We consider the $q \times q$ grid G_q (see Figure 3 for an example). We denote by $v_{i,j}$ the vertex of row i and column j. Evidently, G_q is bipartite;

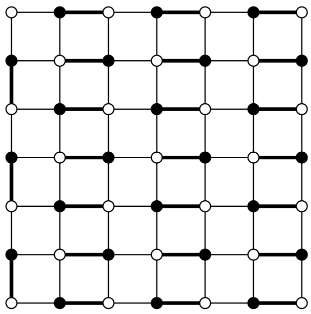

Figure 3: The grid G_7; bold edges indicate the maximum matching M_7.

let V_1, V_2 be the bipartition with $v_{1,1} \in V_2$ (in Figure 3, vertices in V_1 are drawn black, vertices in V_2 are drawn white). Since q is odd, we have $|V_1| = (q^2+1)/2 - 1$ and $|V_2| = (q^2+1)/2$. Next we obtain a formula F_q with $I(F_q) = G_q$: We consider vertices in V_1 as variables, and we associate to every vertex $v_{i,j} \in V_2$ the clause $\{v_{i,j-1}, \overline{v_{i,j+1}}, v_{i-1,j}, \overline{v_{i+1,j}}\} \cap (V_1 \cup \overline{V_1})$. As shown in [17], any $q \times q$ grid, $q \geq 3$, has exactly clique-width $q+1$; hence $dcw(I_d(F_q)) \geq cw(I(F_q)) = cw(G_q) \geq k$.

Consider the matching M_q of G_q consisting of all the edges $(v_{i,2j}, v_{i,2j+1})$ for $i = 1, \dots, q$ and $j = 1, \dots, (q-1)/2$, and the edges $(v_{2i,1}, v_{2i+1,1})$ for $i = 1, \dots, (q-1)/2$ (in Figure 3, edges of M_q are indicated by bold lines). Since $|M_q| = |V_1|$, M_q is a maximum matching and F_q is 0-expanding. By Lemma 6 $\delta^*(F_q) = \delta(F_q) = 1$ follows. (Moreover, since every vertex of G_q can be reached by an M_q-augmenting path from the only unmatched vertex $v_{1,1}$, it follows from Lemma 3(vi) that F_q is 1-expanding.) ☐

It can be shown that every formula whose incidence graph is a square grid is satisfiable (i.e., such formulas are "graph-satisfiable" [35]); hence the formulas F_q constructed in the preceding proof are satisfiable. Since for a directed graph D the directed clique-width of any induced subgraph of D does not exceed the directed clique-width of D, it is not difficult to obtain from F_q *unsatisfiable* formulas of high directed clique-width and constant maximum deficiency. However, it would be interesting to find *minimal unsatisfiable* formulas with such property.

8 Final remarks

We have shown fixed-parameter tractability of the following problems:

(i) Recognition of minimal unsatisfiable formulas with bounded deficiency.

(ii) Satisfiability of formulas with bounded maximum deficiency.

Furthermore, we have shown that tree-width and related parameters which allow fixed-parameter tractability of SAT are incomparable with maximum deficiency. In contrast to tree-width, maximum deficiency can be computed efficiently.

It is remarkable that maximum deficiency as well as tree-width (and the above mentioned variants) ignore the polarities of literal occurrences: we do not distinguish between $x \in C$ and $\overline{x} \in C$ for a variable x and a clause C when we form primal or incidence graphs. Hence some important information gets lost. We think that other translations of formulas into graphs could benefit from this information.

References

[1] R. Aharoni and N. Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *J. Combin. Theory Ser. A*, 43:196–204, 1986.

[2] M. Alekhnovich and A. A. Razborov. Satisfiability, branch-width and Tseitin tautologies. To appear in Proc. of the 43rd IEEE FOCS, 2002.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[4] E. Ben-Sasson and A. Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[5] C. Berge. Two theorems in graph theory. *Proc. Nat. Acad. Sci. U.S.A.*, 43:842–844, 1957.

[6] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[7] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[8] S. A. Cook. An exponential example for analytic tableaux. Manuscript, 1972.

[9] B. Courcelle, J. A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discr. Appl. Math.*, 108(1-2):23–52, 2001.

[10] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discr. Appl. Math.*, 101(1-3):77–114, 2000.

[11] G. Davydov, I. Davydova, and H. Kleine Büning. An efficient algorithm for the minimal unsatisfiability problem for a subclass of CNF. *Annals of Mathematics and Artificial Intelligence*, 23:229–245, 1998.

[12] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2nd edition, 2000.

[13] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Verlag, 1999.

[14] H. Fleischner, O. Kullmann, and S. Szeider. Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference. *Theoretical Computer Science*, 289(1):503–516, 2002.

[15] J. Franco, J. Goldsmith, J. Schlipf, E. Speckenmeyer, and R. P. Swaminathan. An algorithm for the class of pure implicational formulas. *Discr. Appl. Math.*, 96/97:89–106, 1999.

[16] J. Franco and A. Van Gelder. A perspective on certain polynomial time solvable classes of satisfiability. *Discr. Appl. Math.*, 125:177–214, 2003.

[17] M. C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *Internat. J. Found. Comput. Sci.*, 11(3):423–443, 2000. Selected papers from the Workshop on Graph-Theoretical Aspects of Computer Science (WG 99), Part 1 (Ascona).

[18] G. Gottlob and R. Pichler. Hypergraphs in model checking: Acyclicity and hypertree-width versus clique-width. In F. Orejas, P. G. Spirakis, and J. van Leeuwen, editors, *28th International Colloquium on Automata, Languages and Programming (ICALP01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 708–719, 2001.

[19] G. Gottlob, F. Scarcello, and M. Sideri. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artificial Intelligence*, 138(1-2):55–86, 2002.

[20] P. Heusch. The complexity of the falsifiability problem for pure implicational formulas. *Discr. Appl. Math.*, 96/97:127–138, 1999.

[21] P. Heusch, S. Porschen, and E. Speckenmeyer. Improving a fixed parameter tractability time bound for the shadow problem. Technical Report 2001-425, Universität zu Köln, 2001.

[22] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *J. Comput.*, 2:225–231, 1973.

[23] H. Kleine Büning. An upper bound for minimal resolution refutations. In G. Gottlob, E. Grandjean, and K. Seyr, editors, *CSL'98*, volume 1584 of *Lecture Notes in Computer Science*, pages 171–178. Springer Verlag, 1999.

[24] H. Kleine Büning. On subclasses of minimal unsatisfiable formulas. *Discr. Appl. Math.*, 107(1–3):83–98, 2000.

[25] O. Kullmann. Lean clause-sets: Generalizations of minimally unsatisfiable clause-sets. To appear in *Discr. Appl. Math.*

[26] O. Kullmann. An application of matroid theory to the SAT problem. In *Fifteenth Annual IEEE Conference on Computational Complexity*, pages 116–124, 2000.

[27] L. Lovász and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, 1986.

[28] B. Monien and E. Speckenmeyer. Solving satisfiability in less than 2^n steps. *Discr. Appl. Math.*, 10:287–295, 1985.

[29] C. H. Papadimitriou and D. Wolfe. The complexity of facets resolved. *J. of Computer and System Sciences*, 37(1):2–13, 1988.

[30] D. Pretolani. Hierarchies of polynomially solvable satisfiability problems. *Annals of Mathematics and Artificial Intelligence*, 17(3-4):339–357, 1996.

[31] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[32] N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.

[33] P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[34] S. Szeider. On fixed-parameter tractable parameterizations of SAT. In preparation.

[35] S. Szeider. Generalizations of matched CNF formulas. Submitted, 2002.

[36] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discr. Appl. Math.*, 8(1):85–89, 1984.

[37] A. Urquhart. The complexity of propositional proofs. *The Bulletin of Symbolic Logic*, 1(4):425–467, Dec. 1995.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'