A trade-off between length and width in resolution

Neil Thapen[*]
Czech Academy of Sciences
thapen@math.cas.cz

October 15, 2014

Abstract

We describe a family of CNF formulas in n variables, with small initial width, which have polynomial length resolution refutations. By a result of Ben-Sasson and Wigderson it follows that they must also have narrow resolution refutations, of width $O(\sqrt{n \log n})$. We show that, for our formulas, this decrease in width comes at the expense of an increase in size, and any such narrow refutations must have exponential length.

1 Introduction and results

Resolution is a well-known proof system for refuting propositional CNF formulas. A *literal* is a propositional variable or its negation. A *clause* is a disjunction of literals. We define a *conjunctive normal form formula* or CNF to be a set of clauses, which we treat semantically as though it were a conjunction of clauses. The *resolution rule* allows us to derive the clause $C \vee D$ from the two clauses $C \vee q$ and $D \vee \neg q$, where q is any propositional variable. The *weakening rule* allows us to derive a clause C from any subclause D of C. A resolution refutation of a CNF F is a sequence of clauses, ending with the empty clause, where each clause either comes from F or follows from earlier clauses by resolution or weakening.

Every unsatisfiable CNF has a resolution refutation. However, interesting questions remain about the complexity of refutations. We consider two measures of complexity, *length* and *width*, and will also mention a third, *space*. The *length* (or *size*) of a resolution refutation Π is the number of clauses it contains. The *width* of Π is the maximum width of any clause in Π, where the width of a clause is just the number of literals it contains. Similarly the width of a CNF F is the maximum width of any clause in F. The *space* or *clause space* of Π is the number of clauses that need to be kept in memory while verifying Π [8].

A result of [6] showed an interesting and useful connection between the minimal length and minimal width of refutations:

Theorem 1 *Let F be a CNF in n variables with width k. Suppose that F has a resolution refutation Π of length S. Then F also has a resolution refutation Π' of width at most $k + \sqrt{n \log S}$.* □

[*] Partially supported by grant P202/12/G061 of GAČR and RVO:67985840.

In other words, every short refutation can be transformed into a narrow refutation. However, the transformation of Π into Π' used in the proof of Theorem 1 may increase the length of the refutation exponentially. In this paper we address the natural question, posed for example in [4, 15, 14], of whether the theorem can be strengthened to guarantee that the narrow refutation Π' is not substantially longer than the initial short refutation Π.[1] We show that the expected answer ("no") is correct. Our main result is:

Theorem 2 *Fix a small constant $\varepsilon > 0$. Take any sufficiently large m such that both m and m^ε are powers of two. There is a CNF Φ_m with $\Theta(m^{1+2\varepsilon})$ variables and $\Theta(m^{1+3\varepsilon})$ clauses, of width $O(\log m)$, such that*

1. *Φ_m has a refutation of length $O(m^{1+3\varepsilon})$ and width $m + O(\log m)$*
2. *Φ_m has a refutation of width $O(m^\varepsilon)$*
3. *Φ_m has no subexponential length refutation of width strictly less than m.*

By Theorem 1, it follows from item 1 of Theorem 2 (even without item 2) that Φ_m has a refutation of width $O(m^{\frac{1}{2}+\varepsilon}\sqrt{\log m})$. But, by item 3, as long as $\varepsilon < \frac{1}{2}$ every such refutation requires exponential length.

This kind of result is known as a *trade-off* between length and width. The reason for the name is that if we need a refutation of small length, we can find one; and if we need a refutation of small width, we can find one; but we must choose between small length and small width, since there is no way to minimize both in the same refutation. We briefly describe some known trade-offs between complexity measures for resolution — see [14] for a detailed survey.

They were first studied by Ben-Sasson [4], who showed trade-offs between space and width for resolution and between space and length for treelike resolution (in which the underlying graph of every refutation must be a tree). In particular he gave formulas of size n which have linear length treelike refutations with constant space, and which also have constant width treelike refutations, but for which for any refutation Π, the product of the width and the space of Π must be at least $\Omega(n/\log n)$, and for any treelike refutation Π the product of the width and the logarithm of the length of Π must be at least $\Omega(n/\log n)$.

A trade-off between space and length for unrestricted resolution was shown by Nordström in [13], and a robust system for showing such trade-offs was developed in [5]. For example, there are formulas of size n and constant width which have linear length refutations, and which also have refutations with space $O(n/\log n)$, but for which any refutation with this minimal space must have exponential length.

A trade-off between length and width was also shown in [13], giving formulas of size n and constant width which have linear length refutations, and which also have refutations of width $O(\sqrt[3]{n})$, but for which any refutation with this minimal width must have exponential length. However these parameters are not enough to answer the question about Theorem 1 discussed above, and the formula and method of proof in this paper are completely different.

[1] A question about the relation between length and width in the opposite direction also arises from [6]. Any refutation with width w must have length at most $n^{O(w)}$, since there only exist $n^{O(w)}$ many clauses of suitable width. Is there a family of formulas for which this bound is tight, that is, the formulas are refutable in width w, but require length $n^{\Omega(w)}$? This was answered recently in [2]: such families do exist, for $w = n^c$ for any constant $c < 1/2$.

The CNF Φ_m in Theorem 2 is a propositional version of the *coloured polynomial local search principle*, or CPLS, which was introduced in [12] as a combinatorial principle as strong as reflection for resolution. It thus in some sense captures the strength of resolution, and also of first-order theories built around bounded Π_2 induction (such as Buss's theory T^2_2 [7]), as these are closely connected with resolution. We say more about this in Section 2 below. In Section 3 we formally define the CNF Φ_m and prove the length upper bound, and in Section 4 we prove the width upper bound, describing two different refutations of small width. Finally in Section 5 we prove the length lower bound on refutations of small width.

The idea of the lower bound proof is, roughly, that we consider four senses in which a clause can be "narrow" – mostly these differ in which variables we are counting (see Lemma 11). Given a refutation Π, if Π has small width it follows immediately that every clause in Π is narrow in our first sense. If furthermore Π has subexponential length, then we can hit Π with a random restriction such that with high probability every clause in the resulting refutation is also narrow in the remaining three senses. We then use what is essentially an adversary argument to show that no such narrow refutation of the restricted CNF can exist. The restriction and the adversary argument are simpler versions of those used in the resolution length lower bound for the related formula $\overline{\mathrm{GI}}_3$ in [17].

2 Coloured polynomial local search

Definition 3 *The coloured polynomial local search principle is the universal closure of the following first-order formula with parameters a, b, c. Suppose that $G_i(x, y)$ is a three-place relation on $[a] \times [b] \times [c]$, that $u(x)$ is a single-argument function from $[b]$ to $[c]$, and that $f_i(x)$ is a two-argument function, with arguments i and x, from $[a] \times [b]$ to $[b]$. Then the following three formulas cannot all be true:*

1. $\forall y < c,\ \neg G_0(0, y)$
2. $\forall i < a-1\ \forall x < b\ \forall y < c,\ G_{i+1}(f_i(x), y) \to G_i(x, y)$
3. $\forall x < b,\ G_{a-1}(x, u(x))$.

We think of the pairs (i, x) as nodes in a levelled directed graph. We call (i, x) node x on level i. For $i < a-1$, this node has a single neighbour, node $f_i(x)$ on level $i+1$. We take $G_i(x, y)$ as asserting that colour y is present on node x on level i, where a node may potentially have any set of colours from $[0, c)$. The three formulas in Definition 3 become:

1. Node 0 on level 0 has no colours.
2. For every node x on every level $i < a-1$, if the neighbour $f_i(x)$ of x on level $i+1$ has any colour y, then x also has colour y.
3. Every node x on the bottom level has at least one colour, $u(x)$.

Clearly these cannot be true simultaneously in any finite structure.

We also use CPLS as the name of the NP search problem in which we are given the size parameters, together with either oracles or polynomial time machines computing G, u and f, and have to find a witness that one of the three formulas above is false. If we fix $c = 1$ this is equivalent to the well-known *polynomial local search* problem PLS [10]. The CPLS principle asserts that the CPLS search problem is total.

Without going into details about first-order proof systems, the CPLS principle can be proved by bounded Π_2 induction on i, starting at $i = a - 1$ and working towards $i = 0$, using the inductive hypothesis $\forall x{<}b\, \exists y{<}c\, G_i(x, y)$, that every node at level i has a colour. The short resolution refutation in the next section will have essentially this form, deriving a set of clauses expressing $\forall x{<}b\, \exists y{<}c\, G_i(x, y)$ for each i in turn, and in particular using space and width closely related to the bounds b and c on the universal and existential quantifiers. In fact, any first-order proof using a suitable form of bounded Π_2 induction can be made into a resolution refutation in a similar way [11] (see [3] for a recent, self-contained presentation of this translation).

On the other hand, CPLS is the hardest NP search problem that is provably total using this amount of induction, in the sense that any other such search problem is reducible to CPLS. This is the main result of [12], and follows from the translation of bounded Π_2 induction into resolution mentioned above, plus the fact that 1-reflection for resolution is reducible to CPLS. Here 1-reflection for resolution is the NP search problem in which we are given (as oracles or polynomial time machines) a resolution refutation of a narrow CNF together with an assignment to its variables, and have to find a clause of the CNF that is falsified by the assignment. For more on connections of this form between proof systems, search problems and induction, see [17].

3 The CNF and a short refutation

Let a be any natural number and let b and c be powers of two. We will define a CNF formula $\overline{\mathrm{CPLS}}_{a,b,c}$. The formula Φ_m in Theorem 2 is $\overline{\mathrm{CPLS}}_{a,b,c}$ with parameters $a = b = m^\varepsilon$ and $c = m$. The bounds on formula size and proof size in Theorem 2 are shown in this section.

We first list the propositional variables that we will use.

1. For each $i < a$, $x < b$ and $y < c$, there is a variable $G_i(x, y)$.
2. For each $i < a$, $x < b$ and $j < \log b$, there is a variable $(f_i(x))_j$, standing for the jth bit of the value of $f_i(x)$.
3. For each $x < b$ and $j < \log a$, there is a variable $(u(x))_j$, standing for the jth bit of the value of $u(x)$.

The total number of variables is $abc + ab \log b + b \log a$.

If a number $x' < b$ has binary expansion $(x')_0 \dots (x')_{\log b - 1}$ we write $f_i(x) = x'$ to stand for the conjunction expressing that, for each $j < \log b$, the variable $(f_i(x))_j$ has the same value as the corresponding bit $(x')_j$. That is, $f_i(x) = x'$ is the conjunction

$$q_0 \wedge \dots \wedge q_{\log b - 1} \qquad \text{where } q_j = \begin{cases} (f_i(x))_j & \text{if } (x')_j = 1 \\ \neg(f_i(x))_j & \text{if } (x')_j = 0. \end{cases}$$

Similarly if $y < c$ has binary expansion $(y)_0 \dots (y)_{\log c - 1}$ we write $u(x) = y$ to stand for the conjunction expressing that, for each $j < \log c$, the variable $(u(x))_j$ has the same value as the corresponding bit $(y)_j$.

We will frequently write $v_1 \wedge \cdots \wedge v_k \to w_1 \vee \cdots \vee w_\ell$ to stand for the clause $\neg v_1 \vee \cdots \vee \neg v_k \vee w_1 \vee \cdots \vee w_\ell$. With this notation the resolution rule may take the form: from $A \wedge q \to C$ and $A \wedge \neg q \to D$ derive $A \to C \vee D$ (where A is a conjunction).

Definition 4 *The formula $\overline{\mathrm{CPLS}}_{a,b,c}$ consists of the following three sets of clauses, which we will call axioms 1, 2 and 3:*

1. *For each $y < c$, the clause*

$$\neg G_0(0, y)$$

2. *For each $i < a - 1$, each pair $x, x' < b$ and each $y < c$, the clause*

$$f_i(x) = x' \wedge G_{i+1}(x', y) \to G_i(x, y)$$

3. *For each $x < b$ and each $y < c$, the clause*

$$u(x) = y \to G_{a-1}(x, y).$$

Axiom 2 has width $\log b + 2$ and axiom 3 has width $\log c + 1$. The total number of clauses is $c + (a - 1)b^2c + bc$.

Theorem 5 *The formula $\overline{\mathrm{CPLS}}_{a,b,c}$ has a refutation simultaneously of length $O(ab^2c)$, space $2b + \log b + 3$ (assuming $\log c \leq b$) and width $c + \log b + 1$.*

Proof For each i, define a set of clauses

$$M_i := \{ \bigvee_{y<c} G_i(x, y) : x < b \}$$

expressing that every node at level i has a colour. Notice that M_i has space b and width c. We construct the refutation by deriving M_i for $i = a - 1, \ldots, 0$ in turn, and then deriving the empty clause from M_0. The details are in the following three claims.

Claim 1 From axiom 3 we can derive M_{a-1} in length $O(bc)$, space $b + \log c + 1$ and width c.

Claim 2 For each $i < a - 1$, from axiom 2 and M_{i+1} we can derive M_i in length $O(b^2c)$, space $b + \log b + 3$ and width $c + \log b + 1$.

Claim 3 From axiom 1 and M_0 we can derive the empty clause in length $O(c)$, space 3 and width c.

We use Claim 1 to derive M_{a-1}. We then keep M_{a-1} in memory, taking up b memory locations, while using Claim 2 to derive M_{a-2}. We store M_{a-2}, forget M_{a-1}, and continue. Once we have derived M_0 we use Claim 3 to reach a contradiction. The maximum space used is either $b + \log c + 1$ while deriving M_{a-1}, or $2b + \log b + 3$ while deriving each M_i from M_{i+1}.

Proof of claim 1 Fix $x < b$. For a binary string σ of length $\log c$ or less, and a number $y < c$, say that y *extends* σ if the sequence of the first $|\sigma|$ bits in the binary expansion of y equals σ, that is, if $(y)_j = \sigma_j$ for all $j < |\sigma|$.

Let $\phi_\sigma(x)$ be the conjunction $q_0 \wedge \cdots \wedge q_{|\sigma|-1}$ where q_j is $(u(x))_j$ if $\sigma_j = 1$ or $\neg(u(x))_j$ if $\sigma_j = 0$, so that $\phi_\sigma(x)$ is true exactly in assignments where $u(x)$ extends σ. Let $\theta_\sigma(x)$ be the clause

$$\phi_\sigma(x) \to \bigvee_{y \text{ extends } \sigma} G_{a-1}(x, y).$$

Notice that if $|\sigma| < \log c$, then $\theta_{\sigma 0}(x)$ and $\theta_{\sigma 1}(x)$ have the forms

$$\theta_{\sigma 0}(x) : \ \phi_\sigma(x) \wedge \neg(u(x))_{|\sigma|} \to \bigvee_{y \text{ extends } \sigma 0} G_{a-1}(x,y)$$

$$\theta_{\sigma 1}(x) : \ \phi_\sigma(x) \wedge (u(x))_{|\sigma|} \ \to \bigvee_{y \text{ extends } \sigma 1} G_{a-1}(x,y).$$

We can derive $\theta_\sigma(x)$ from these by resolving on the variable $(u(x))_{|\sigma|}$.

Axiom 3 consists of $\theta_\sigma(x)$ for every σ of length exactly $\log c$. So by the observation above, we can derive $\theta_\emptyset(x)$ from axiom 3 using a derivation in the form of a complete binary tree of height $\log c$. This uses length $O(c)$, space $\log c + 2$ and width c, the maximum width of the clauses $\theta_\sigma(x)$. Finally, the clause $\theta_\emptyset(x)$ is exactly $\bigvee_{y<c} G_{a-1}(x,y)$, so to show the claim we derive $\theta_\emptyset(x)$ for each $x < b$ in turn.

Proof of claim 2 Fix $x < b$. For $x' < b$, let $\phi(x')$ be the clause

$$f_i(x) = x' \to \bigvee_{y<c} G_i(x,y).$$

The clause $\phi(x')$ can be obtained from $\bigvee_{y<c} G_{i+1}(x',y)$, which is in M_{i+1}, by resolving with instances of axiom 2 for each $y < c$ in turn. This takes length $O(c)$, space 3 and width $c + \log b + 1$.

We now use a similar argument to the proof of claim 1 to derive the clause $\bigvee_{y<c} G_i(x,y)$ from all the clauses $\phi(x')$, using a derivation in the form of a complete binary tree of height $\log b$. To save space we do not derive all of the clauses $\phi(x')$ together at the beginning, but only as we need them. Hence the derivation of $\bigvee_{y<c} G_i(x,y)$ takes length $O(bc)$, space $\log b + 4$ and width $c + \log b + 1$. As before, to show the claim derive this for each $x < b$ in turn.

Proof of claim 3 Resolve $\bigvee_{y<c} G_0(0,y)$ with all instances of axiom 1. □

4 Two narrow refutations

The main purpose of this section is to motivate the definition of the random restriction ρ in Section 5 below. We describe, in Theorems 6 and 7, two narrow strategies for the Prover in a certain Prover-Adversary game based on $\overline{\mathrm{CPLS}}_{a,b,c}$ (this is equivalent to, but intuitively simpler than, describing narrow resolution refutations). In Section 5 we want to show that no narrow refutation can be small, which in particular means that we should be able to show that no small strategy similar to the two outlined here can work.

Part 1 of the definition of ρ (Definition 8) can be seen as blocking any small strategy similar to the one outlined in Theorem 6, where the Prover tries to learn long paths in f, because it generates lots of cases that the Prover must be able to remember, forcing his strategy to have many nodes. Part 3 of the definition does the same for strategies similar to the one in Theorem 7, where the Prover tries to remember a colour on many different nodes. (See [16] for more on this kind of approach to length lower bounds.)

The Prover-Adversary game works as follows. At each turn, the Prover can ask the Adversary the value of a variable, and record the corresponding literal in his memory; alternatively, the Prover can forget a literal from memory to allow the space to be re-used. The Adversary can give any answer which does not directly contradict the current contents of the Prover's memory, and the

Prover wins when his memory falsifies some axiom of $\overline{\mathrm{CPLS}}_{a,b,c}$. It is easy to see that a winning strategy for the Prover that requires no more than w units of memory (where a unit is enough space to record one literal) can be turned into a resolution refutation of $\overline{\mathrm{CPLS}}_{a,b,c}$ of width w.

Theorem 6 $\overline{\mathrm{CPLS}}_{a,b,c}$ *has a refutation of width* $a \log b + \log c$.

Proof By querying all bits of each $f_i(x_i)$ in turn, the Prover first learns a sequence $x_0, \ldots, x_{a-1}$ such that $x_0 = 0$ and $f_i(x_i) = x_{i+1}$ for each $i < a - 1$. This requires $(a-1)\log b$ units of memory.

The Prover then uses $\log c$ more units of memory to learn that $u(x_{a-1}) = y$ for some colour y. The Prover then queries $G_{a-1}(x_{a-1}, y)$ and must get the answer 1, since otherwise the Adversary would violate axiom 3. At this point the Prover can forget $u(x_{a-1}) = y$.

For $i = a-2, \ldots, 0$ the Prover then queries $G_i(x_i, y)$ and must get the answer 1 each time, or the Adversary would violate axiom 2. Each time the Prover may then forget the previous value $G_{i+1}(x_{i+1}, y)$. For $i = 0$ this forces the Adversary to violate axiom 1. □

Theorem 7 $\overline{\mathrm{CPLS}}_{a,b,c}$ *has a refutation of width* $2b + \log b + \log c$.

Proof By a result of [1] bounding the minimal width of refuting a CNF in terms of the minimal space, the existence of a refutation of roughly this width follows already from the space upper bound on $\overline{\mathrm{CPLS}}_{a,b,c}$ shown by the refutation in Theorem 5. In some sense the refutation we describe here is dual to that one (see also [9]).

For each $x < b$ in turn, the Prover learns $u(x) = y$ for some colour y, queries $G_{a-1}(x, y)$ and must get the answer 1 (by axiom 3), and then forgets $u(x) = y$. This can be done in $b + \log c$ units of memory in total.

The Prover then repeats the following process for each $i = a-2, \ldots, 0$. For each $x < b$ the Prover learns $f_i(x) = x'$ for some x', then queries $G_i(x, y)$, where y is the colour for which he knows $G_{i+1}(x', y)$. This must get the answer 1 (by axiom 2). The Prover then forgets $f_i(x) = x'$ and goes on to the next x. Having done this for every x at level i, he forgets all the values $G_{i+1}(x, y)$ from the previous level. The maximum memory used during this process is $2b + \log b$.

When this has reached level 0, the Prover knows that $G_0(0, y) = 1$ for some colour y, contradicting axiom 1. □

The refutation in Theorem 6 has length at least b^a, since it contains a distinct clause for every possible sequence $x_0, \ldots, x_{a-1}$. The refutation in Theorem 7 has length at least c^b, since it contains a distinct clause corresponding to the conjunction $G_{a-1}(0, y_0) \wedge \cdots \wedge G_{a-1}(b-1, y_{b-1})$ for every possible choice of colours $y_0, \ldots, y_{b-1}$.

5 A length lower bound for narrow refutations

We now prove the last part of Theorem 2, that there is no refutation of Φ_m with simultaneously small width and subexponential length. Recall that Φ_m is the formula $\overline{\mathrm{CPLS}}_{a,b,c}$ with parameters $a = b = m^\varepsilon$ and $c = m$, where $\varepsilon > 0$ is a small constant and m and m^ε are both powers of two.

By *subexponential* we mean smaller than 2^{m^δ} for every $\delta > 0$. By *exponentially high probability* we mean probability greater than $1 - 2^{-m^\delta}$ for some $\delta > 0$. By *polynomially high probability* we mean probability greater than $1 - m^{-\delta}$ for some $\delta > 0$. The main parameter appearing in the proof will be a rather than m, but since $a = m^\varepsilon$ this does not change these definitions.

Suppose for a contradiction that there is a refutation Π of Φ_m with subexponential length and with width strictly less than m. Let $p = a^{-3/4}$ and $w = a^{7/8}$.

Definition 8 *A* random restriction *ρ is a partial assignment chosen in three stages, as follows.*

1. *Independently for each pair (i, x), with probability p put (i, x) into a set Γ. Then for each $(i, x) \in \Gamma$, for each y set the variable $G_i(x, y)$ independently to 0 or 1 with probability $1/2$. For such (i, x) we say "$G_i(x, \cdot)$ is set in ρ."*
2. *For each node x on level $a - 1$ with $(i, a-1) \in \Gamma$, choose a random y such that $G_{a-1}(x, y) = 1$ and set all bits of $u(x)$ to satisfy $u(x) = y$. For such x we say "$u(x)$ is set in ρ." (With exponentially small probability there is no such y – in this case do nothing.)*
3. *Independently for each pair (i, x) with $i < a-1$, with probability p put (i, x) into a set Δ. For each $i < a - 1$ let S_i be the set of nodes x on level i with $(i, x) \in \Delta$. Randomly choose an injection h_i from S_i onto a random set of nodes of size $|S_i|$ on level $i + 1$, and for each $x \in S_i$ set all bits of $f_i(x)$ according to h_i. For such (i, x) we say "$f_i(x)$ is set in ρ."*

Definition 9 *The CNF $\Phi_m \restriction \rho$ is formed from Φ_m by removing every clause containing a literal satisfied by ρ, and removing every literal falsified by ρ from the remaining clauses. $\Pi \restriction \rho$ is formed from Π by the same operations.*

After the restriction, $\Pi \restriction \rho$ is a resolution refutation of $\Phi_m \restriction \rho$ (some instances of the resolution rule in Π may have become instances of weakening in $\Pi \restriction \rho$).

We now have two goals. The first is to use the assumption about the length of Π to show that with exponentially high probability ρ simplifies Π, in that every clause in $\Pi \restriction \rho$ is narrow in a certain sense. This is Lemma 11. The second is to show that, with polynomially high probability, not only does ρ not immediately falsify Φ_m, but $\Phi_m \restriction \rho$ does not even have any refutation that is narrow in the above sense.

For this we define *safe configurations*, which informally are certain partial assignment α such that $\Phi_m \restriction \alpha$ looks difficult to refute. In Lemma 12 we show that with polynomially high probability ρ does not contain certain local patterns that would make refuting $\Phi_m \restriction \rho$ easy. In Lemmas 14 and 15 we show that this implies that ρ is a safe configuration, and that no safe configuration falsifies Φ_m. Finally we use a sequence of safe configurations to show that $\Pi \restriction \rho$ cannot be a narrow refutation of $\Phi_m \restriction \rho$, completing the proof.

Lemma 10 *By the Chernoff bound, with exponentially high probability, for each pair (i, x) such that $G_i(x, \cdot)$ is set in ρ, $G_i(x, y) = 1$ in ρ for at least a third of the colours $y < c$. Furthermore for each $i < a$, $G_i(x, \cdot)$ is set in ρ for at most $2pa$ values $x < b$, and for each $i < a - 1$, $f_i(x)$ is set in ρ for at most $2pa$ values $x < b$.* □

Lemma 11 *With exponentially high probability, for every clause C in $\Pi \restriction \rho$,*

1. *C contains a variable $G_i(x,y)$ for at most $c-1$ many triples (i,x,y)*
2. *C contains any variable $G_i(x,y)$ for at most w many pairs (i,x)*
3. *C contains any variable from $f_i(x)$ for at most w many pairs (i,x)*
4. *C contains any variable from $u(x)$ for at most w many values x.*

Proof Item 1 follows directly from the assumption that the width of Π is strictly less than m. This is the only place where we use this assumption.

For the remaining three items, since Π has subexponential length it is enough to show that, independently for each clause C in Π, if C is not narrow in this sense then with exponentially high probability C is satisfied by ρ, and hence does not appear in $\Pi \restriction \rho$.

For item 2, suppose that a clause C in Π contains a literal $G_i(x,y)$ or $\neg G_i(x,y)$ for more than w many pairs (i,x). For each such (i,x), the probability that such a literal is satisfied in ρ is at least $p/2$. Hence the probability that none of these literals in C is satisfied is at most $(1-p/2)^w < e^{-\frac{1}{2}pw} = e^{-\frac{1}{2}a^{1/8}}$.

For item 3, there is a complication that, if $f_i(x_1), \dots, f_i(x_t)$ are all the values of f set in ρ on level i, then the bits $(f_i(x_k))_j$ are not all independent, since the values assigned to $f_i(x)$ on level i are constrained to be distinct for distinct nodes x. However, we may assume that $f_i(x_1), \dots, f_i(x_t)$ were chosen in the order shown, and that when each $f_i(x_k)$ was chosen the only constraint was that the $k-1$ values already chosen on that level were excluded. By Lemma 10 we may assume $k \leq 2pa$. Hence if a literal ℓ has the form $(f_i(x))_j$ or $\neg(f_i(x))_j$, if $f_i(x)$ is set in ρ then there are $a/2$ possible values it may take which satisfy ℓ, of which at most $2pa$ were excluded. Hence the probability that ℓ is satisfied is at least $p(a/2 - 2pa)/a$, regardless of how earlier values were set, which is at least $p/3$ for large a. We then argue as for item 2.

For item 4, suppose that a literal ℓ has the form $(u(x))_j$ or $\neg(u(x))_j$. Then if $G_{a-1}(x,\cdot)$ is set in ρ, by the Chernoff bound we may assume that, of the $c/2$ possible values y of $u(x)$ that would satisfy ℓ, for at least one third we have $G_{a-1}(x,y) = 1$. Hence for any x the probability that $u(x)$ is set in ρ in a way that satisfies ℓ is at least $p/6$. We then argue as for item 2. □

We define a *path* of length $k \geq 0$ in a partial assignment α as a sequence of pairs $(i, x_0), \dots, (i+k, x_k)$ such that $f_{i+j}(x_j) = x_{j+1}$ in α for each $j < k$.

Lemma 12 *With polynomially high probability, the following are all true.*

1. *$G_0(0,\cdot)$ and $f_0(0)$ are not set in ρ.*
2. *There is no triple (i,x,x') such that $G_i(x,\cdot)$, $G_{i+1}(x',\cdot)$ and $f_i(x)$ are all set in ρ, with $f_i(x) = x'$. In other words, there is no path in ρ of length 1 with G set at both ends.*
3. *There is no 4-tuple (i,x,x',x'') such that $G_i(x,\cdot)$, $G_{i+2}(x'',\cdot)$, $f_i(x)$ and $f_{i+1}(x')$ are all set in ρ, with $f_i(x) = x'$ and $f_{i+1}(x') = x''$. That is, there is no path in ρ of length 2 with G set at both ends.*
4. *There is no 4-tuple (i,x,x',x'') such that $f_i(x)$, $f_{i+1}(x')$ and $f_{i+2}(x'')$ are all set in ρ, with $f_i(x) = x'$ and $f_{i+1}(x') = x''$. That is, there is no path in ρ of length 3 or more.*

Proof Item 1 is true with probability $(1-p)^2$.

For item 2, for any triple (i, x, x') the probability that $G_i(x, \cdot)$, $G_{i+1}(x', \cdot)$ and $f_i(x)$ are all set is p^3, and the probability that $f_i(x) = x'$ is $1/a$. There are no more than a^3 such triples, so by the union bound the probability that there is any triple violating the condition is less than $a^2p^3 = a^{-1/4}$. The calculation for item 3 is similar.

For item 4, for any 4-tuple (i, x, x', x'') the probability that $f_i(x)$, $f_{i+1}(x')$ and $f_{i+2}(x'')$ are set is p^3, and the probability that $f_i(x) = x'$ and $f_{i+1}(x') = x''$ is $1/a^2$. There are no more than a^4 such tuples, so by the union bound the probability that there is any tuple violating the condition is less than $a^2p^3 = a^{-1/4}$. □

Fix a restriction ρ which satisfies the conditions of Lemmas 10, 11 and 12.

Definition 13 *A* safe configuration *is a partial assignment α which extends ρ and satisfies the conditions listed below. We say that a colour y is* present *or* forbidden *at (i, x) if respectively $G_i(x, y) = 0$ or $G_i(x, y) = 1$ in α.*

1. *For each pair (i, x) either all variables belonging to $f_i(x)$ are set, or none are. Similarly for each x either all variables belonging to $u(x)$ are set, or none are.*
2. *For each level $i < a$, the partial assignment to the variables f_i defines a partial injection.*
3. *If $(0, 0)$ and (i, x) are on the same path in α, then no colour y is present at (i, x).*
4. *If (i, x) and (i', x') are on the same path in α, then no colour y is simultaneously present at (i, x) and forbidden at (i', x').*
5. *If (i, x) and $(a-1, x')$ are on the same path in α and $u(x')$ is set to a colour y, then the colour y is not forbidden at (i, x).*

Lemma 14 *The restriction ρ is a safe configuration.*

Proof It satisfies conditions 1 and 2 by construction. It satisfies condition 3 by item 1 of Lemma 12, which guarantees that $(0, 0)$ is not on any non-trivial path. It satisfies condition 4 by items 2, 3 and 4 of Lemma 12. Condition 5 follows from condition 4 and the fact that, if $u(x)$ is set in ρ, then we must have $G_{a-1}(x, u(x)) = 1$ in ρ. □

Lemma 15 *No clause in Φ_m, and hence no clause in $\Phi_m \restriction \rho$, is falsified by any safe configuration.*

Proof Conditions 3, 4 and 5 of the definition of safe configuration respectively guarantee that no clause from axiom 1, 2 or 3 of $\overline{\mathrm{CPLS}}_{a,b,c}$ is falsified. □

The next lemma will allow us to derive a contradiction from the existence of the refutation $\Pi \restriction \rho$. The empty clause at the end of the refutation is falsified by a safe configuration, namely ρ. Now suppose that a clause E in $\Pi \restriction \rho$ is falsified by some safe configuration. Either E is derived from an earlier clause by weakening, or E is derived from two earlier clauses by resolution, or E is an initial clause of $\Phi_n \restriction \rho$. In both of the first two cases we can find an earlier clause in the proof which is falsified by some safe configuration – in the case of weakening this is trivial, and in the case of resolution we use Lemma 16. Hence we must eventually find an initial clause of $\Phi_n \restriction \rho$ which is falsified by some safe configuration, contradicting Lemma 15.

Lemma 16 *Suppose that a clause E in $\Pi \restriction \rho$ is derived from clauses C and D by a single use of the resolution rule, and that there is a safe configuration α which falsifies E. Then there is a safe configuration β which falsifies either C or D.*

Proof We write $\alpha \setminus \rho$ for the assignment γ disjoint from ρ such that $\alpha = \rho \cup \gamma$. By Lemma 11, by shrinking α as necessary we may assume without loss of generality that $\alpha \setminus \rho$ is *narrow* in the following sense: it sets a variable $G_i(x, y)$ for at most $c - 1$ many triples (i, x, y); it sets any variable $G_i(x, y)$ for at most w many pairs (i, x); it sets $f_i(x)$ for at most w many pairs (i, x); and it sets $u(x)$ for at most w many values x.

Let q be the variable resolved on to derive E. If α already assigns a value to q, then α already falsifies either C or D, by the structure of the resolution rule. Otherwise, it is enough to show how to extend α to a safe configuration which assigns a value to q. We consider three cases.

First suppose that q has the form $G_i(x, y)$. If (i, x) is on the same path as some node at which colour y is present, or some node $(a - 1, x')$ such that $u(x')$ is set to y, we put $G_i(x, y) = 1$. Otherwise we put $G_i(x, y) = 0$. This does not affect conditions 1 and 2 of the definition of a safe configuration and preserves conditions 4 and 5 by construction. The only way it can falsify condition 3 is if (i, x) is on a path which contains both $(0, 0)$ and some node $(a - 1, x')$ such that $u(x')$ is set to y. But any such path must have length $a - 1$, the full height of the graph. By Lemma 12 all paths in ρ have length 2 or less, hence by our assumption about the narrowness of $\alpha \setminus \rho$, the longest possible path in α would consist of $w + 1$ many paths of length 2 from ρ linked together by w many paths of length 1 from $\alpha \setminus \rho$, with total length $3w + 2$.

Now suppose that q has the form $(f_i(x))_j$. Say that a node $(i + 1, x')$ is *marked* if any variable $G_{i+1}(x', y)$ is assigned a value, or $f_{i+1}(x')$ is set, or $f_i(x'') = x'$ for some x'', or $i + 1 = a - 1$ and $u(x')$ is set. In each of the four cases there are at most $2pa + w$ such nodes, by Lemma 10 and our assumption about the narrowness of $\alpha \setminus \rho$. Hence for large a there are many unmarked nodes. Choose any unmarked node $(i + 1, x')$ and set $f_i(x)$ to be x'. By construction, this preserves conditions 1 and 2. It preserves conditions 3 and 4 because it does not add or forbid a colour on any existing path, or join any paths together. It preserves condition 5 because we avoid nodes $(a - 1, x')$ for which $u(x')$ is set.

Finally suppose that q has the form $(u(x))_j$. Let π be the path containing $(a - 1, x)$. If π contains a node (i, x) for which $G_i(x, \cdot)$ is set in ρ, then every colour y is either forbidden or present on π, and by Lemma 10 at most $2/3$ of colours are forbidden. If π contains no such node, then by the assumption about the narrowness of $\alpha \setminus \rho$, at most $c - 1$ colours are forbidden on π. In either case, at least one colour y is not forbidden on π. Set $u(x) = y$. This does not affect conditions 1 to 4, and preserves condition 5 by construction. □

Acknowledgements I am grateful to Jakob Nordström for making me aware of this problem, and to Jakob Nordström and Nicola Galesi for helpful comments on an early version of this paper.

References

[1] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008.

[2] A. Atserias, M. Lauria, and J. Nordström. Narrow proofs may be maximally long. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC '14)*, pages 286–297, 2014.

[3] A. Beckmann, P. Pudlák, and N. Thapen. Parity games and propositional proofs. *ACM Transactions on Computational Logic*, 15(2):17:1–17:30, 2014.

[4] E. Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, 2009.

[5] E. Ben-Sasson and J. Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions (extended abstract). In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, 2011. The full version is available as ECCC Technical Report TR10-125.

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

[7] S. Buss. *Bounded Arithmetic.* Bibliopolis, 1986.

[8] J. L. Esteban and J. Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001.

[9] Y. Filmus, M. Lauria, M. Mikša, J. Nordström, and M. Vinyals. From small space to small width in resolution. In *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14)*, pages 300–311, 2014.

[10] D. Johnson, C. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[11] J. Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[12] J. Krajíček, A. Skelley, and N. Thapen. NP search problems in low fragments of bounded arithmetic. *Journal of Symbolic Logic*, 72(2):649–672, 2007.

[13] J. Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, 2009.

[14] J. Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, 2013.

[15] J. Nordström and J. Håstad. Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, 2013.

[16] P. Pudlák. Proofs as games. *American Mathematical Monthly*, pages 541–550, 2000.

[17] A. Skelley and N. Thapen. The provably total search problems of bounded arithmetic. *Proceedings of the London Mathematical Society*, 103(1):106–138, 2011.