# A NEAR-OPTIMAL SEPARATION OF REGULAR AND GENERAL RESOLUTION *

ALASDAIR URQUHART†

**Abstract.** This paper gives a near-optimal separation between regular and unrestricted resolution. The main result is that there is a sequence of sets of clauses $\Pi_1, \Pi_2, \ldots, \Pi_i, \ldots$ for which the minimum regular resolution refutation of $\Pi_i$ has size $2^{\Omega(R_i/(\log R_i)^7 \log\log R_i)}$, where $R_i$ is the minimum size of an unrestricted resolution refutation of $\Pi_i$. This improves earlier lower bounds for which the separations proved were of the form $2^{\Omega(\sqrt[3]{R})}$ and $2^{\Omega(\sqrt[4]{R}/(\log R)^3)}$.

## 1. Introduction.

**1.1. The regularity restriction.** This paper proves a separation between the size of regular and unrestricted resolution refutations of sets of clauses that provides a much closer approach to an optimal separation between these two propositional proof systems than earlier results.

The regularity restriction was first introduced by Grigory Tseitin in a groundbreaking article [16] that is the published version of a talk given in 1966 at a Leningrad seminar. This restriction is very natural, in the sense that algorithms such as that of Davis, Logemann and Loveland [7] (the prototype of almost all satisfiability algorithms used in practice today) can be understood as a search for a regular refutation of a set of clauses. If refutations are represented as trees, rather than directed acyclic graphs, then minimal-size refutations are regular, a result first stated without proof by Tseitin [16]. It can be proved easily by a simple pruning argument [17, p. 436].

The main result of Tseitin's paper [16] is an exponential lower bound for regular resolution refutations of contradictory CNF formulas based on graphs. Tseitin [14, p. 472] makes the following remarks about the heuristic interpretation of the regularity restriction:

> The regularity condition can be interpreted as a requirement for not proving intermediate results in a form stronger than that in which they are later used (if $A$ and $B$ are disjunctions such that $A \subseteq B$, then $A$ may be considered to be the stronger assertion of the two); if the derivation of a disjunction containing a variable $\xi$ involves the annihilation of the latter, then we can avoid this annihilation, some of the disjunctions in the derivation being replaced by "weaker" disjunctions containing $\xi$.

These remarks of Tseitin suggest that there is always a regular resolution refutation of minimal size, as in the case of tree resolution. Consequently, some researchers tried to extend Tseitin's results to general resolution by showing that regular resolution can simulate general resolution efficiently; there is at least one published claim of such a simulation. However, these attempts were doomed to failure.

The first example of a contradictory CNF formula whose shortest resolution refutation is irregular was given by Wenqi Huang and Xiangdong Yu [9]. Subsequently, Andreas Goerdt [8] gave the first super-polynomial separation between regular res-

*The author gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada.

† Department of Computer Science, University of Toronto, Toronto, Ontario, Canada. A preliminary version of this paper [18] was published in the proceedings of the 11th International Conference on Theory and Applications of Satisfiability Testing (SAT 2008), Guangzhou, China.

olution and unrestricted resolution by constructing a family of formulas that have polynomial-size resolution refutations, but require super-polynomial-size regular resolution refutations.

Goerdt's results were improved to an exponential separation by Alekhnovich, Johannsen, Pitassi and Urquhart [1]. Their paper in fact contains two separate proofs of an exponential separation. The first presents a sequence $GT'_{n,\rho}$ of sets of clauses that have general resolution refutations with size $O(n^3)$, but require regular resolution refutations of size $2^{\Omega(n)}$. The second gives an infinite sequence of sets of clauses $Stone(G, S)$ based on a pebbling problem that have general resolution refutations with size $O(n^4)$, but require regular resolution refutations of size $2^{\Omega(n/(\log n)^3)}$.

Hence, the best separations so far between regular and general resolution are of the form $2^{\Omega(\sqrt[3]{R})}$, and $2^{\Omega(\sqrt[4]{R}/(\log R)^3)}$, where $R$ is the size of the smallest general resolution refutation of the set of clauses in question. It is natural to ask whether we can improve these separations. In fact, we know that we cannot do better than a $2^{\Omega(R \log\log R/\log R)}$ separation. This is because Ben-Sasson, Impagliazzo and Wigderson [4] showed that if $R$ is the size of a general resolution refutation of a set of clauses, then there is a tree resolution refutation with size $2^{O(R \log\log R/\log R)}$. Since a tree resolution refutation of minimal size is regular, it follows that the same upper bound holds for regular resolution.

The present paper makes a much closer approach to a matching lower bound. The main result is that there is a sequence of contradictory sets of clause $\Pi_i$, with an associated unbounded parameter $n = n(i)$, so that $\Pi_i$ has a general resolution refutation with size $O(n(\log n)^5)$, but any regular resolution refutation has size $2^{\Omega(n/[(\log n)^2 \log\log n])}$ The proof of this result is an amalgamation and extension of ideas underlying the two previous separation results.

**1.2. Preliminaries.** A *literal* is a propositional variable $x$ or its negation $\neg x$; we say that the literals $x$ and $\neg x$ both *involve* the variable $x$. A *clause* is a set of literals, interpreted as the disjunction of the set. The *language* of a set of clauses is the set of variables in it. For clauses containing exactly one positive literal, we use the implication $p_1, \ldots, p_k \to q$ as alternative notation for the clause $\neg p_1 \vee \cdots \vee \neg p_k \vee q$. For notational convenience, we shall also allow the case where the positive literal $q$ is replaced by the propositional constant $\perp$. For any assignment $\sigma$, $\sigma(\perp) = 0$, so that the expression "$p_1, \ldots, p_k \to \perp$" is an alternative notation for the purely negative clause $\neg p_1 \vee \cdots \vee \neg p_k$

If $p$ is a variable, and $C$ a clause, then we say that $p$ has a *positive occurrence* in $C$ if $p$ is in $C$, and a *negative occurrence* in $C$ if $\neg p$ is in $C$. In addition, we shall say that $\perp$ has a positive occurrence in the purely negative Horn clause $p_1, \ldots, p_k \to \perp$. If $\Sigma$ is a set of clauses, and $x, y$ are variables in $\Sigma$, or the propositional constant $\perp$, then we say that there is an *implicational chain from $x$ to $y$ in* $\Sigma$ if there is a sequence $x = x_0, \ldots, x_k = y$ of variables (or constants) and a sequence $C_1, \ldots, C_k$ of clauses so that for all $i$, $0 < i \leq k$, $x_{i-1}$ occurs negatively and $x_i$ positively in $C_i$.

The *resolution rule* allows us to derive the *resolvent* $C \vee D$ from the clauses $C \vee x$ and $D \vee \neg x$ by *resolving on* the variable $x$; a clause $C \vee D$ can also be derived from a clause $C$ by *weakening*. A *resolution derivation* of a clause $C$ from a set of clauses $\Sigma$ consists of a sequence of clauses in which each clause is either a clause of $\Sigma$, or derived from earlier clauses by resolution or weakening, and $C$ is the last clause in the sequence; it is a *refutation* of $\Sigma$ if $C$ is the empty clause $\Lambda$. The *size* $|\mathcal{R}|$ of a refutation $\mathcal{R}$ is the number of resolvents in it. We can represent it as a directed acyclic graph (dag) where the nodes are the clauses in the refutation, each clause of

$\Sigma$ has out-degree 0, and any other clause has one or two arcs pointing to the clause or clauses from which it is derived. Resolution is a *sound* and *complete* propositional proof system, that is to say, a set of clauses $\Sigma$ is unsatisfiable if and only if there is a resolution refutation for $\Sigma$.

A resolution refutation is *regular* if on any path from $\Lambda$ to an initial clause in $\Sigma$ (in the directed acyclic graph associated with the refutation), each variable is resolved on at most once along the path.

It is sometimes helpful to view a regular resolution refutation as a branching program. Representing the refutation as a dag, let us say that a variable $x$ is *queried at a node* in the dag if the node is labelled with a clause $C \vee D$, derived from parent clauses $C \vee x$ and $D \vee \neg x$ by resolving on $x$. Starting from the empty clause $\Lambda$ at the root of the dag, we can construct a path in the refutation by answering the queries occurring in the path; the answers determine an assignment to the variables queried along the path. The path is chosen so that the assignment falsifies all the clauses in the path. Thus, if the variable $x$ is queried at a node, and the answer is "false," then the next node in the path is labelled with the parent clause containing the literal $x$; similarly for the answer "true." If $C \vee D$ is derived by weakening from $C$, then the path continues to $C$. The path constructed in this way must end with an initial clause falsified by this assignment.

An *assignment* (*restriction*) for a set of clauses is a Boolean assignment to some of the variables in the set; the assignment is *total* if all the variables in the set are assigned values. If $C$ is a clause, and $\sigma$ an assignment, then we write $C \restriction \sigma$ for the result of applying the assignment to $C$, that is, $C \restriction \sigma = 1$ if $\sigma(l) = 1$ for some literal $l$ in $C$, otherwise, $C \restriction \sigma$ is the result of removing all literals set to 0 by $\sigma$ from $C$. If $\Sigma$ is a set of clauses, then $\Sigma \restriction \sigma$ is the set of clauses $C \restriction \sigma$, $C$ a clause in $\Sigma$, with clauses set to 1 by $\sigma$ removed.

If $\mathcal{R}$ is a resolution refutation of $\Sigma$, and $\sigma$ a restriction for $\Sigma$, then we define the *restriction* $\mathcal{R} \restriction \sigma$ of $\mathcal{R}$ to be the sequence of clauses resulting from $\mathcal{R}$ by replacing all of the clauses $C$ in $\mathcal{R}$ by $C \restriction \sigma$, and then removing all of the clauses set to 1. It is easy to verify that $\mathcal{R} \restriction \sigma$ is a resolution refutation of $\Sigma \restriction \sigma$, and that $\mathcal{R} \restriction \sigma$ is regular, if $\mathcal{R}$ is regular.

The notation $\log x$ stands for the base two logarithm of $x$, and $\ln x$ the natural logarithm of $x$.

## 2. Pebbling games and pebbling formulas.

**2.1. The pebbling game.** A *pointed graph* $G$ is a directed acyclic graph where all nodes have indegree at most two, having a unique sink, or target node, to which there is a directed path from all the nodes in $G$. It is *binary* if all nodes except for the source nodes have indegree two. If $v$ is a node in a pointed graph $G$, then $G \restriction v$ is the subgraph of $G$ restricted to the nodes from which there is a directed path to $v$.

The *pebbling game* played on a pointed graph $G$ is a one-player game in the course of which pebbles are placed on or removed from nodes in $G$. The rules of the game are as follows;

1. A pebble may be placed on a source node at any time.
2. If all predecessors of a node $u$ are marked with pebbles, then a pebble may be placed on node $u$.
3. A pebble may be removed from a node at any time.

A *move* in the game consists of placing or removing one of the pebbles in accordance with one of the three rules. The *configuration* at a given stage in the game is the set of nodes in $G$ that are marked with a pebble. A play of the game begins with no

pebbles on $G$. The goal of the game is to place a pebble on the sink node $t$, while minimizing the number of pebbles used (that is, minimizing the number of pebbles on the graph at any stage of the game). Thus a successful play of the game can be presented as a sequence of configurations $C_0, \ldots, C_k$, where $C_0 = \emptyset$ and $t \in C_k$, where $C_{j+1}$ is obtained from $C_j$ by one of the three rules.

A *strategy* for the game is a sequence of moves following the rules of the game that ends in pebbling the target node. The *cost* of such a strategy is the minimum number of pebbles required in order to execute it, that is to say, the size of the largest configuration in the sequence of configurations produced by following the strategy. The *pebbling number* of $G$, written as $\sharp G$, is the minimum cost of a strategy for the pebbling game played on $G$.

**2.2. Pebbling formulas.** We associate a contradictory set of clauses $\mathrm{Peb}(G)$ with every pointed graph $G$. Each node in $G$ except the target $t$ is assigned a distinct variable; to simplify notation, we identify a node with the variable associated with it, and use the notation $\mathrm{Var}(G)$ for the set of these variables. We associate the constant $\perp$ (falsum) with the target node $t$, and make the identification $t = \perp$.

Definition 2.1. *If $G$ is a pointed graph, $Peb(G)$ is a set of clauses expressed in terms of the variables $Var(G)$, so that $Peb(G) = \{Clause(v) : v \in G\}$.*

1. *If $v$ is a source node of $G$, then $Clause(v) = v$.*
2. *If $v$ is a node in $G$, with predecessor $u$, then $Clause(v) = u \to v$.*
3. *If $v$ is a node in $G$, with predecessors $u, w$, then $Clause(v) = u, w \to v$.*

If we set some variables in $\mathrm{Peb}(G)$, then the resulting set of clauses is not necessarily of the form $\mathrm{Peb}(G')$, where $G'$ is a subgraph of $G$. We shall focus on a family of special assignments, called *pebbling assignments*, that preserve this property. If $v \in G$, $v \neq t$, then we define the assignment $[\![v := 1]\!]$ to be the assignment defined by first setting the variable $v$ to 1, and then setting to 1 any variable $u$ for which there is no implicational chain from $u$ to $\perp$ in the resulting clause set. The assignment $[\![v := 0]\!]$ is defined as follows: first, choose a directed path $\pi = (v, \ldots, t)$ from $v$ to the target $t$, set all the nodes in the path to 0, and in addition set any node from which $v$ is not reachable, but not in the path $\pi$, to 1. The assignment $[\![v := 0]\!]$ is not uniquely determined by this construction, since it depends on the path chosen – however, this is not important, since the set of clauses $\mathrm{Peb}(G) \restriction [\![v := 0]\!]$ resulting from the restriction is independent of the path. A *pebbling assignment* results from a sequence of restrictions of the form $[\![v := 0]\!]$ and $[\![w := 1]\!]$.

The effect of the restrictions just defined can be described directly as an operation on the underlying graph. If $G$ is a pointed graph, and $v \in G$, $v \neq t$, $G[v := 1]$ is the graph resulting from $G$ by first removing $v$, together with all edges entering or leaving $v$, and then restricting the resulting graph to the nodes from which the target node $t$ is accessible. $G[v := 0]$ is the pointed graph $G \restriction v$.

Lemma 2.2.

1. *For $b = 0, 1$, $Peb(G) \restriction [\![v := b]\!] = Peb(G[v := b])$.*
2. *If $G$ is a pointed graph, and $v \in G$, then*

$$\sharp G \leq \max\{\sharp G[v := 0], \sharp G[v := 1] + 1\}.$$

**Proof.** The first part of the lemma follows straightforwardly from the definitions. For the second part, we employ the following strategy in the pebble game on $G$; the strategy is the same as the one used in Lemma 15 of [4].

First, follow a minimum cost strategy to pebble $v$ in $G[v := 0]$. Second, leaving a pebble on $v$, but removing all other pebbles, follow a minimum cost strategy in the

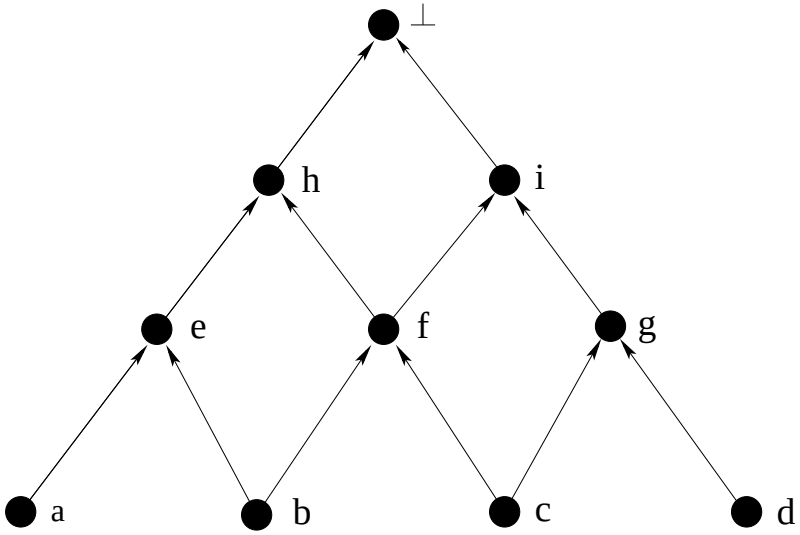


FIG. 2.1. *A pyramid graph*

pebbling game on $G[v := 1]$ to pebble the target node in $G$, using the extra pebble for any moves where a pebble is needed on $v$ to justify a placement. The cost is at most $\max\{\sharp G[v := 0], \sharp G[v := 1] + 1\}$. □

EXAMPLE 2.3. *If $G$ is the pyramid graph shown in Figure 1, then Peb($G$) is the set of clauses*

$$\{a, b, c, d, (a, b \to e), (b, c \to f), (c, d \to g), (e, f \to h), (f, g \to i), (h, i \to \bot)\}.$$

*The assignment $[\![e := 1]\!]$ sets the variables $a$ and $e$ to 1; thus the set of clauses Peb($G$)$\restriction [\![e := 1]\!]$ is Peb($G[e := 1]$) where $G[e := 1]$ results from $G$ by removing the nodes $a$ and $e$. If we choose the path $f \to h \to \bot$, then the assignment $[\![f := 0]\!]$ sets the variables $f$ and $h$ to 0, while the variables $a, d, e, g, i$ are all set to 1. The set of clauses Peb($G$)$\restriction [\![f := 0]\!]$ is Peb($G[f := 0]$), where $G[f := 0]$ is the subgraph of $G$ containing only the nodes $b, c$ and $f$.*

If $\Sigma$ is a set of clauses, then a *$C$-critical assignment* is a total assignment to the variables in $\Sigma$ that makes all the clauses true, except $C$. In the case of Peb($G$), we are interested in a particular family of critical assignments. Let $v$ be a vertex in $G$, and $\pi = (v, \ldots, t)$ a directed path in $G$ from $v$ to the target node $t$. Set all the nodes in the path $\pi$ to 0, and all other nodes in $G$ to 1. This assignment makes all of the clauses in Peb($G$) true, except for Clause($v$). An assignment determined by the path $\pi$ we shall call a *$v$-critical assignment*, since the clause that it falsifies is associated with the node $v$. Since we have assumed that $G$ is a pointed graph, such $v$-critical assignments exist for all the nodes $v$ in $G$, so that Peb($G$) is minimally inconsistent.

LEMMA 2.4. *If $G$ is a pointed graph with $\sharp G = p$, then there are at least $p$ vertices $v$ in $G$ for which there is a $v$-critical assignment for Peb($G$).*

**Proof.** Every pebbling strategy for $G$ must contain a configuration with $p$ pebbles, so there must be at least $p$ vertices in $G$. For every vertex $v$ in $G$, we can construct a $v$-critical assignment for Peb($G$) by choosing a path from $v$ to the target node. □

If $G$ is a binary pointed graph, then the clause set $\mathrm{Peb}(G)$ contains, in addition to three-literal clauses, unit clauses (associated with source nodes) and a two-literal clause (associated with the target node). It is convenient, in view of the construction in Lemma 3.4, to convert it into a set of 3-literal clauses.

DEFINITION 2.5. *Let $\Sigma$ be a set of clauses, in which each clause contains at most three literals Then $\Sigma^*$ is the set of clauses obtained from $\Sigma$ by the following construction. First, introduce for each unit clause $l$ in $\Sigma$, a pair of new auxiliary variables $x_l$ and $y_l$. Second, replace the unit clause $l$ by the set of four 3-literal clauses $\{l \vee x_l \vee y_l, l \vee \overline{x_l} \vee y_l, l \vee x_l \vee \overline{y_l}, l \vee \overline{x_l} \vee \overline{y_l}\}$. Third, introduce for each two-literal clause $C$ a new auxiliary variable $x_C$, and replace $C$ by the set of two 3-literal clauses $\{C \vee x_C, C \vee \overline{x_C}\}$.*

We write $\mathrm{Peb}^*(G)$ for $\mathrm{Peb}(G)^*$. Let $G$ be a binary pointed graph. If $v$ is an interior node in $G$ (neither a source node, nor the target node), then we set $\mathrm{Clauses}(v) = \{\mathrm{Clause}(v)\}$. If $v$ is a source node, then $\mathrm{Clauses}(v)$ is defined to be the set of four clauses $\{v \vee x_v \vee y_v, v \vee \overline{x_v} \vee y_v, v \vee x_v \vee \overline{y_v}, v \vee \overline{x_v} \vee \overline{y_v}\}$. If $\perp$ is the target node in $G$, and $C = \mathrm{Clause}(\perp)$, then $\mathrm{Clauses}(\perp)$ is defined to be the set $\{C \vee x_C, C \vee \overline{x_C}\}$. If $X$ is a subset of the nodes in $G$, then $\mathrm{Clauses}(X)$ is defined to be $\bigcup\{\mathrm{Clauses}(v) : v \in X\}$. In this notation, $\mathrm{Peb}^*(G) = \bigcup\{\mathrm{Clauses}(v) : v \in G\}$, in analogy with the definition $\mathrm{Peb}(G) = \{\mathrm{Clause}(v) : v \in G\}$.

## 3. Constructing hard problems.

**3.1. Earlier constructions.** In this section, we construct the problems that produce our improved separation between general and regular resolution. The overall approach is derived from the first separation result described above in §1; the proof of this result is based on the following idea. The construction begins with a sequence of problems $GT_n$ that are hard for tree resolution though not for regular resolution. The set of clauses $GT_n$ asserts that there is a directed acyclic ordering on $n$ nodes that has no sink; these problems were introduced in the proof complexity literature by Krishnamurthy [11], who conjectured that they require superpolynomial-size resolution refutations. That conjecture was refuted by Stålmarck, who showed that they in fact have linear size resolution refutations [15]. However, Bonet and Galesi [6] showed that they require exponentially large tree resolution refutations, thus showing an exponential separation between general and tree resolution. In fact, the refutations constructed by Stålmarck are regular, so that the result of Bonet and Galesi shows an exponential separation between regular and tree resolution.

The exponential lower bound for tree resolution shows that any tree refutation for $GT_n$ must contain exponentially many paths starting from the root of the tree. Although this fact does not force regular resolution refutations to be large, as Stålmarck showed, nevertheless we can convert the $GT_n$ examples into hard problems for regular resolution by making a small modification. The idea (due to the late Misha Alekhnovich) is to add "spoiler" literals to the clauses in such a way as to force the exponentially many paths in a tree refutation not to overlap, at least in their initial segments. The new sets of clauses $GT'_{n,\rho}$ require exponentially large regular resolution refutations, though the general resolution size remains linear, as in the case of the original $GT_n$ problems.

The construction in the present paper follows the outline above, but this time starting from the pebbling formulas. The second lower bound proof in [1] also began from the pebbling formulas, but used a somewhat different construction to convert them into hard examples for regular resolution. The present result combines features of both proofs; the construction proceeds in two stages.

**3.2. Xorification of clause sets.** The construction starts from $\mathrm{Peb}^*(G)$, for $G$ a pointed graph. The first stage applies to $\mathrm{Peb}^*(G)$ a construction of Alekhnovich and Razborov. This stage has two purposes. If we apply the construction to a graph $G$ with large pebbling number, then the resulting set of clauses requires large tree refutations, though not large general resolution refutations (see the remarks following Theorem 3.2), thus achieving the first purpose. The second purpose is to "blow up" the set of clauses $\mathrm{Peb}^*(G)$ sufficiently so that the random construction in §3.3 succeeds with high probability.

Definition 3.1. *Let $\Sigma$ be a set of clauses, and $k > 1$ a positive integer. For each variable $x$ in $\Sigma$, introduce a set of $k$ distinct variables $\{x_1, \cdots, x_k\}$. Then the set of clauses $\Sigma^{k\oplus}$, the $k$-*xorification *of $\Sigma$, is defined as follows: first, substitute the formula $x_1 \oplus \cdots \oplus x_k$ for all of the variables $x$ occurring in $\Sigma$, second, convert the resulting formula into conjunctive normal form.*

In Definition 3.1, by "conversion to conjunctive normal form" is meant a simple expansion without introducing extra variables. For example, if $C$ is the clause $(b \vee \neg c)$, then $\{C\}^{2\oplus}$ is the set of four clauses

$$\{(\neg b_1 \vee \neg b_2 \vee \neg c_1 \vee c_2), (\neg b_1 \vee \neg b_2 \vee c_1 \vee \neg c_2), (b_1 \vee b_2 \vee \neg c_1 \vee c_2), (b_1 \vee b_2 \vee c_1 \vee \neg c_2)\}.$$

If $C$ is a clause containing $m$ literals, then $\{C\}^{k\oplus}$ contains $2^{m(k-1)}$ clauses, each of length $mk$. Consequently, if $v$ is an interior node of a binary pointed graph, then $|\mathrm{Clauses}(v)^{k\oplus}| = 2^{3(k-1)}$, while if $v$ is a source node, then

$$|\mathrm{Clauses}(v)^{k\oplus}| = 4 \cdot 2^{3(k-1)} = 2^{3k-1},$$

and finally, if $v$ is the target node, then

$$|\mathrm{Clauses}(v)^{k\oplus}| = 2 \cdot 2^{3(k-1)} = 2^{3k}.$$

Hence, if $G$ has $n$ nodes,

$$n \cdot 2^{3(k-1)} \leq |\mathrm{Peb}^*(G)^{k\oplus}| \leq n \cdot 2^{3k-1}.$$

If $V$ is the set of variables in $\mathrm{Peb}^*(G)^{k\oplus}$, then the size of $V$ can be estimated by the inequalities $nk \leq |V| \leq 3nk$.

The special case of Definition 3.1 where $k = 2$ is the original construction of Alekhnovich and Razborov, as reported in a paper by Eli Ben-Sasson [3]. They observed that it could be used to produce hard problems for resolution from clause sets requiring refutations of large width. Let $\mathrm{Width}(\Sigma)$ be the size of the largest clause in $\Sigma$, and $\mathrm{Width}(\Sigma \vdash 0)$ the minimum width of a resolution refutation of $\Sigma$.

Theorem 3.2. **(Alekhnovich and Razborov)** *If $\Sigma$ is contradictory, then any resolution refutation of $\Sigma^{2\oplus}$ has size* $\exp[\Omega(\mathit{Width}(\Sigma \vdash 0) - \mathit{Width}(\Sigma))]$.

More important in the present context is the fact that xorification can be used to produce examples that separate tree resolution from general resolution; this forms part of the proof strategy outlined in §3.1. More precisely, the following result holds: If $G$ is a pointed graph with $n$ nodes and pebbling number $p$, then the set of clauses $\mathrm{Peb}(G)^{2\oplus}$ has general resolution refutations of size $O(n)$, but every tree resolution refutation of $\mathrm{Peb}(G)^{2\oplus}$ has size $2^{\Omega(p)}$. This theorem can be proved by imitating the proof of Ben-Sasson, Impagliazzo and Wigderson [4]. Their result involves clause sets that are the "orification" $\mathrm{Peb}(G)^{\vee}$ of $\mathrm{Peb}(G)$ rather than the xorification $\mathrm{Peb}(G)^{2\oplus}$; however, the steps in their proof can be imitated almost word for word in the case of $\mathrm{Peb}(G)^{2\oplus}$ to produce essentially the same result as their main theorem.

An assignment $\mu$ for $\mathrm{Peb}^*(G)^{k\oplus}$ is defined to be *full* if whenever $v$ is a vertex in $G$, and $\mu$ assigns a value to some variable $v_j$ associated with $v$, then all the variables attached to $v$ are assigned values by $\mu$. If $\mu$ is such a full assignment, then we can construct an assignment for $\mathrm{Peb}(G)$ from $\mu$ by setting $\sigma(v) = \mu(v_1) \oplus \cdots \oplus \mu(v_k)$. In this case, we say that the constructed assignment is the *projection* of $\mu$, written $\pi(\mu)$. We shall say that a full assignment $\mu$ for $\mathrm{Peb}^*(G)^{k\oplus}$ is a *pebbling assignment* if its projection $\pi(\mu)$ is a pebbling assignment for $\mathrm{Peb}(G)$.

LEMMA 3.3. *Let $G$ be a binary pointed graph. If $\sigma$ is a $v$-critical assignment for $Peb(G)$, for $v \in G$, and $C$ is in $Clauses(v)^{k\oplus}$, then there is a $C$-critical full assignment $\mu$ for $Peb^*(G)^{k\oplus}$ so that $\pi(\mu) = \sigma$.*

**Proof.** If $v$ is a source node or the target node in $G$, and $D$ is a clause in $\mathrm{Clauses}(v)$, then we can construct a $D$-critical assignment for $\mathrm{Peb}^*(G)$ from $\sigma$ by giving the appropriate values to the auxiliary variables (if any) occurring in the clause $D$. If $v$ is not a source node or the target node, and $\mathrm{Clauses}(v) = \{D\}$, then $\sigma$ can be extended to a $D$-critical assignment for $\mathrm{Peb}^*(G)$ by setting the auxiliary variables arbitrarily.

Starting from a $v$-critical assignment for $\mathrm{Peb}^*(G)$, we can construct an assignment $\mu$ that assigns values to $u_1, \ldots, u_k$, for all nodes $u \in G$, so as to make $C$ false, but all other clauses in $\mathrm{Peb}^*(G)^{k\oplus}$ true, and in addition, this assignment $\mu$ satisfies $\pi(\mu) = \sigma$. $\square$

**3.3. Adding random literals.** The set of clauses $\mathrm{Peb}^*(G)^{k\oplus}$, for a binary pointed graph $G$ and suitable $k$, is not in fact hard to refute using regular resolution. This can be seen by examining the short irregular refutations described in the proof of Lemma 4.3 below. If the step in the fourth paragraph of that proof is omitted, then the resulting refutations of $\mathrm{Peb}^*(G)^{k\oplus}$ are regular refutations with size linear in the size of the initial clauses. To construct hard instances for regular resolution, we have to add "spoiler literals," following the method of this section.

The second stage of the construction starts from $\mathrm{Peb}^*(G)^{k\oplus}$, for a binary pointed graph $G$ and suitable $k$, and replaces each clause $C$ in the set with a pair of clauses, $C \vee \rho(C)$ and $C \vee \neg\rho(C)$, where $\rho(C)$ is a variable associated with $C$ by a function $\rho$. For the second stage to work (that is to say, for the resulting sets of clauses to require exponentially large regular resolution refutations), it is essential that $\rho$ have a special property, namely that the image of a large set of clauses has a large intersection with any large set of variables (where the notion of "large" is assigned an appropriate meaning in each case). The easiest way to construct such a function is by a probabilistic argument, given in the following lemma.

LEMMA 3.4. *Let $G$ be a binary pointed graph with $n$ nodes, $\delta = 5/3$, $k = \lceil \delta \log\log n \rceil + 1$; define $\Sigma = Peb^*(G)^{k\oplus}$, and $V = Var(\Sigma)$.*

*For any $d > 0$, and sufficiently large $n$, there exists a map $\rho$ from $\Sigma$ to $V$ satisfying the condition: For all $A \subseteq G$ and $B \subseteq V$, with $|A| = |B| = \lfloor dn/\log n \rfloor$, $|\rho(Clauses(A)^{k\oplus}) \cap B| \geq dn/2\log n$.*

**Proof.** Using the abbreviation $m = \lfloor dn/\log n \rfloor$, let $A$ be a subset of $G$ with $|A| = m$. By the estimates following Definition 3.1, $\mathrm{Clauses}(A)^{k\oplus}$ contains $\Theta(n(\log n)^4)$ clauses, while $|\Sigma| = \Theta(n(\log n)^5)$. In addition, for the set of variables $V$, we have the estimate

$$2n \log\log n \leq |V| \leq 6n \log\log n,$$

for sufficiently large $n$.

Consider the space $\mathcal{R}$ of all random maps from $\Sigma$ to $V$; that is to say, for each $C \in \Sigma$, a variable $\rho(C) \in V$ is chosen uniformly at random. For $A \subseteq G$ and $B \subseteq V$, with $|A| = |B| = m$, we say that $\rho$ is *bad for $A$ and $B$* if $|\rho(\text{Clauses}(A)^{k\oplus}) \cap B| < dn/2 \log n$.

We establish the existence of the map $\rho$ by a probabilistic argument; to accomplish this, we need to prove exponentially small upper bounds on the probability that a random map is bad for given sets $A$ and $B$. For the first stage of the proof, we think of these sets $A$ and $B$ as being fixed. In proving the upper bound on the probability of a random map being bad for these sets, it helps to view the construction of a random map as resulting from a series of independent experiments, each of them consisting in the construction of a random map from a subset of $\Sigma$.

We partition $\text{Clauses}(x)^{k\oplus}$ as $\Xi_1(x), \ldots, \Xi_q(x)$, where $q = \lfloor (\log n)^3 \rfloor$, so that each set $\Xi_j(x)$ in the sequence contains at least $(\log n)^2$ clauses. For fixed $j$, $1 \leq j \leq q$, let $\Sigma_j$ be the union of all the $\Xi_j(x)$, for $x \in G$, and for $A \subseteq G$, $|A| = m$, let $\text{Clauses}_j(A)$ be the union of all the $\Xi_j(x)$, for $x \in A$. Then $\text{Clauses}_j(A)$ contains $\Theta(n \log n)$ clauses. Let $\rho_j$ be a random map from $\Sigma_j$ to $V$; we take $\rho$ to be the union of the sequence $\rho_1, \ldots, \rho_q$ of independently constructed random maps. We use the notation $\mathcal{R}_j$ to denote the space of all random maps from $\text{Clauses}_j(A)$ to $V$.

For a given $j$, where $1 \leq j \leq q$, let $Z$ be the random variable representing the number of variables in $B$ not in the image of $\text{Clauses}_j(A)$ under $\rho_j$:

$$Z(\rho_j) = |\{x \in B |\, x \notin \rho_j(\text{Clauses}_j(A))\ \}|.$$

The first stage in the proof of the lemma involves giving an upper bound on the expected value $E(Z)$. For $B = \{b_1, b_2, \ldots, b_i, \ldots, b_m\}$, define an indicator random variable $\Theta_i$ by:

$$\Theta_i(\rho_j) = \begin{cases} 1, & \text{if } b_i \notin \rho_j(\text{Clauses}_j(A)) \\ 0, & \text{if } b_i \in \rho_j(\text{Clauses}_j(A)), \end{cases}$$

so that $Z = \Theta_1 + \cdots + \Theta_m$. We estimate the expected value of $\Theta_i$ by

$$\begin{aligned} E(\Theta_i) &= \left(1 - \frac{1}{|V|}\right)^{|\text{Clauses}_j(A)|} \\ &\leq \left(1 - \frac{1}{6n \log\log n}\right)^{\Theta(n \log n)} \\ &\leq \exp\left(-\Omega\left(\frac{\log n}{\log\log n}\right)\right), \end{aligned}$$

showing that

$$E(Z) \leq m \cdot \exp\left(-\Omega\left(\frac{\log n}{\log\log n}\right)\right) = m \cdot o(1).$$

It follows that for any given positive $\gamma$, $E(Z) < \gamma m$, for sufficiently large $n$. For the remainder of the proof, we assume that $n$ is chosen sufficiently large so that $E(Z) < m/8$.

In the second stage of the proof, we need to show that the random variable $Z$ is tightly concentrated around its mean. To do this, we employ a large deviation bound for martingales, following [10].

Order $\text{Clauses}_j(A)$ as $\{C_1, \ldots, C_p\}$. For $\rho \in \mathcal{R}_j$, and $1 \le t \le p$, define $\rho\restriction t$ to be the restriction of $\rho$ to the set $\{C_1, \ldots, C_t\}$. Define an equivalence relation $\equiv_t$ on $\mathcal{R}_j$ by setting

$$\rho \equiv_t \sigma \iff \rho\restriction t = \sigma\restriction t,$$

for $1 \le t \le p$, and let $\equiv_0$ be the universal relation on $\mathcal{R}_j$. Let $\mathcal{F}_t$ be the finite Boolean algebra whose atoms are the blocks of the partition of $\mathcal{R}_j$ induced by $\equiv_t$, for $0 \le t \le p$; the sequence $\mathcal{F}_0, \ldots, \mathcal{F}_p$ of Boolean algebras forms a *filtration* over the set $\text{Clauses}_j(A)$. Now define a sequence of random variables $Z_0, \ldots, Z_p$ by setting $Z_t = E(Z|\mathcal{F}_t)$. Then $Z_0 = E(Z)$, $Z_p = Z$, and the sequence $Z_0, \ldots, Z_p$ forms a martingale [12, p. 221], the Doob martingale associated with the filtration $\mathcal{F}_0, \ldots, \mathcal{F}_p$. The intuitive picture here is that at time 0, we begin with no specific information about a given map $\rho$; we learn its values on $\text{Clauses}_j(A)$ one by one at each successive time step $t$, until we have full information about $\rho\restriction \text{Clauses}_j(A)$ at time $p$. The values that $\rho$ takes on $\Sigma_j \setminus \text{Clauses}_j(A)$ have no influence on $Z$; that is why $Z_p = Z$.

If $\rho$ and $\rho'$ are two maps from $\Sigma_j$ to $V$ that differ at most at a single point in $\Sigma_j$, then $|Z(\rho) - Z(\rho')| \le 1$. In the terminology of Alon and Spencer [2, p. 89], the random variable $Z$ satisfies the Lipschitz condition relative to the filtration $\mathcal{F}_0, \ldots, \mathcal{F}_p$. It follows by Theorem 4.1 of Chapter 7 of the monograph by Alon and Spencer [2, p. 90] that $|Z_{t+1} - Z_t| \le 1$. Consequently, by the martingale tail inequality of Hoeffding and Azuma [12, p. 221] [2, p. 85],

$$\begin{aligned} P(Z \ge m/2) &\le P(Z - E(Z) > 3m/8) \\ &< \exp(-(3m/8)^2/2p) \\ &\le \exp(-\Omega(n/(\log n)^3)). \end{aligned}$$

Let $W$ be the random variable representing the number of variables in $B$ not in the image of $\text{Clauses}(A)^{k\oplus}$ under $\rho$:

$$W(\rho) = |\{x \in B |\, x \notin \rho(\text{Clauses}(A))\ \}|.$$

Since the maps $\rho_1, \ldots, \rho_q$ are constructed independently, it follows that

$$P(W \ge m/2) \le [\exp(-\Omega(n/(\log n)^3))]^q = \exp(-\Omega(n)).$$

We can now complete the proof of the existence of a map $\rho$ satisfying the condition of the lemma. The probability that a random map $\rho \in \mathcal{R}$ is bad for some $A$ and $B$ is bounded by

$$\binom{n}{m}\binom{6n\log\log n}{m}\exp(-\Omega(n)).$$

Using the simple inequality

$$\binom{n}{k} \le \left(\frac{en}{k}\right)^k,$$

found in Bollobás's textbook on graph theory [5, p. 216], the first binomial coefficient above can be bounded by

$$\begin{aligned} \binom{n}{m} &\le \left(\frac{en}{m}\right)^m \\ &= \left(e^{O(\log\log n)}\right)^{O(n/\log n)} \\ &= e^{O(n\log\log n/\log n)}. \end{aligned}$$

A similar computation shows that the second binomial coefficient has the same upper bound. Hence, the probability can be bounded above by

$$\exp(O(n \log\log n/\log n))\exp(-\Omega(n)) = \exp(-\Omega(n)).$$

Consequently, the probability that a random map $\rho$ is bad for some $A$ and $B$ is exponentially small for sufficiently large $n$, showing that a map satisfying the condition of the lemma must exist. □

**3.4. Construction of the hard problems.** Let's say that for $\Sigma = \mathrm{Peb}^*(G)^{k\oplus}$, a map $\rho$ is *good for* $\Sigma$ if it satisfies the condition of Lemma 3.4. This lemma states that for $\delta = 5/3$, $k = \lceil \delta \log\log n \rceil + 1$, and sufficiently large $n$, there is a map that is good for $\Sigma = \mathrm{Peb}^*(G)^{k\oplus}$. This enables us to construct our set of hard problems for regular resolution. The construction is based on the following result of Paul, Tarjan and Celoni.

THEOREM 3.5. *[13] There is a sequence of binary pointed graphs $G_1, \ldots, G_i, \ldots$ with pebbling number at least $cn(i)/\log n(i)$, for sufficiently large $i$, where $n(i) = |G_i| = O(i2^i)$, and $c > 1/20$.*

It should be mentioned that the graphs $H_i$ constructed by Paul, Tarjan and Celoni, though binary, are not pointed, since they are constructed to have multiple sink nodes. However, in their main theorem, they show that for sufficiently large $i$, their graph $H_i$ contains a sink node that requires $cn(i)/\log n(i)$ pebbles to pebble it, starting from the empty configuration. Hence, we can construct a pointed graph from $H_i$ by choosing such a sink node, and considering the subgraph $G_i$ containing all the nodes from which this sink is accessible. This subgraph $G_i$ still satisfies the condition $n(i) = |G_i| = O(i2^i)$, so the main theorem of Paul, Tarjan and Celoni continues to hold, if we add the qualifier "pointed." We construct our sequence $\Pi_1, \Pi_2, \ldots, \Pi_i, \ldots$ by applying the earlier constructions to $G_1, G_2, \ldots, G_i, \ldots$.

DEFINITION 3.6. *Let $G_1, G_2, \ldots, G_i, \ldots$ be the sequence of graphs of Theorem 3.5, $k(i) = \lceil \delta \log\log n(i) \rceil + 1$, and $\rho_i$ a map that is good for $\Sigma_i = Peb^*(G_i)^{k(i)\oplus}$. Then $\Pi_i$ is defined to be the set of clauses*

$$\{C \vee \rho_i(C) : C \in \Sigma_i\} \cup \{C \vee \neg\rho_i(C) : C \in \Sigma_i\}.$$

The set of clauses $\Pi_i$ contains $\Theta(n(\log n)^5)$ clauses, and $\Theta(n \log\log n)$ variables, where $n = n(i)$ is the size of the pointed graph $G_i$. By a "pebbling assignment for $\Pi_i$" we mean a pebbling assignment for $\Sigma_i = \mathrm{Peb}^*(G_i)^{k(i)\oplus}$.

**4. Lower bound for regular resolution.** In this section, to avoid notational clutter, we adopt the following conventions. We assume that we are dealing with the set of clauses $\Pi = \Pi_i$, for sufficiently large $i$, write $G$ for $G_i$, $n$ for $n(i) = |G_i|$, and $\Pi$ for $\Pi_i$. In addition, we take $c > 1/20$ to be the constant in Theorem 3.5.

Before entering in to the details of the next two lemmas it may be helpful to outline the basic steps in the proofs. The central quantity that controls the strategy of both proofs is the pebbling number of the underlying graph $G$; this basic quantity we shall write as $N = cn/\log n$. In Lemma 4.1, we show that if a set of clauses in the language of $\Pi$ is not near-exponentially large, then we can apply a restriction to it in such a way that $G$, when acted on by the projection of this restriction, still has pebbling number at least $N/2$, but the restricted set contains no large clauses, where "large" means "containing at least $N/8$ literals." In Lemma 4.2, we show that if $\mu$ is a pebbling restriction such that $G\restriction\pi(\mu)$ has pebbling number at least $N/2$, then any regular resolution refutation of $G\restriction\pi(\mu)$ must contain a large clause. Taken

together, the two lemmas imply that any regular resolution refutation of $\Pi$ has to be near-exponentially large.

**4.1. Destroying large clauses by restrictions.** Define a clause to be *large* if it contains at least $N/8$ literals.

LEMMA 4.1. *If $\Sigma$ is a set of clauses in the language of $\Pi$, containing fewer than $2^{N/[3840(\log n)\log\log n]}$ clauses, then there is a pebbling assignment $\mu$ so that:*

1. *$\Sigma\restriction\mu$ contains no large clauses.*
2. *$G\restriction\pi(\mu)$ has pebbling number at least $N/2$.*

**Proof.** There are at most $6n\log\log n$ variables in $\Pi$, and so at most $12n\log\log n$ literals involving those variables. If we choose a literal at random and set it to 1, then the probability this assignment sets a given large clause to 1 is at least $1/r$, where $r = (96/c)\log n\log\log n \leq 1920\log n\log\log n$. Hence, the average number of large clauses in $\Sigma$ set to 1 is at least $|\Sigma|/r$.

Choose a literal $l$ achieving at least this average, and set it to 1. Suppose that $l$ involves a variable $v_j$, where $v \in G$. Set the remaining variables in the set of variables $\{v_1, \ldots, v_k\}$ so as to maximize $\sharp G[v := b]$, where $b = v_1 \oplus \cdots \oplus v_k$. Now extend this assignment to produce a pebbling assignment for $\Pi$ whose projection to $\mathrm{Peb}(G)$ is $[\![v := b]\!]$. Then the set $\Sigma'$ resulting from this restriction contains at most $(1-1/r)|\Sigma|$ large clauses, and by Lemma 2.2, $\sharp G[v := b]$ is at most one less than $\sharp G$.

If we repeat this procedure $\lfloor N/2\rfloor$ times, resulting in a restriction $\mu$, then the restricted set contains at most

$$(1-1/r)^{\lfloor N/2\rfloor}2^{N/[3840(\log n)\log\log n]}$$

large clauses. Since the ratio $\lfloor x\rfloor/x$ tends to 1 as $x \to \infty$, we can assume the inequality

$$\lfloor N/2\rfloor \geq (0.9N)/2.$$

Hence, we can estimate the lefthand term in the upper bound on the number of large clauses by

$$\begin{aligned}(1-1/r)^{\lfloor N/2\rfloor} &\leq (e^{-1/r})^{(0.9N)/2}\\ &\leq \exp\left(-\frac{0.9N}{3840(\log n)\log\log n}\right).\end{aligned}$$

Consequently, the number of large clauses remaining after the restriction is bounded by

$$\exp\left(-\frac{(0.9-\ln 2)N}{3840(\log n)\log\log n}\right) < 1,$$

showing that $\Sigma\restriction\mu$ contains no large clauses. By construction, the pebbling number of $G\restriction\pi(\mu)$ is at least $N/2$. □

**4.2. Large clauses.** The next lemma shows that large clauses must occur in a regular refutation of the restricted pebbling clauses, provided that the pebbling number of the underlying graph is still large. This is the only part of the proof where the regularity restriction is employed.

LEMMA 4.2. *Let $\mathcal{R}$ be a regular resolution refutation of $\Pi\restriction\mu$, where $\mu$ is a pebbling assignment, and $G\restriction\pi(\mu)$ has pebbling number $\geq N/2$. Then $\mathcal{R}$ contains a clause with at least $N/8$ literals.*

**Proof.** Viewing $\mathcal{R}$ as a branching program, we describe a strategy for constructing a path in $\mathcal{R}$, starting with the root, and concurrently constructing a full assignment to certain variables in $\Pi\restriction\mu$. The strategy is as follows. We suppose that the path has been constructed as far as a node $p$, and that $\sigma$ is the current full assignment. We extend the path and the assignment according to these rules:

1. If the clause labelling $p$ is derived by weakening, then continue the path to the unique parent node; the assignment remains unchanged.
2. If the variable queried at $p$ is already assigned a value by $\sigma$, answer the query according to $\sigma$, and continue the path according to this answer.
3. If the variable queried at $p$ is not assigned a value by $\sigma$, then it must be associated with a node $v \in G\restriction\pi(\sigma)$. Extend $\sigma$ to a pebbling assignment $\sigma'$ so that $\pi(\sigma')$ contains $[\![v := b]\!]$, choosing $b$ so as to maximize the pebbling number of $G\restriction\pi(\sigma')$. Then extend the path in accordance with $\sigma'$.

Continue according to these rules until $\lfloor N/4 \rfloor$ nodes in $G$ have been queried (that is to say, variables attached to the nodes have been queried), let $C$ be the clause at the end of the resulting path, and $\tau$ the resulting assignment.

By Lemma 2.4, there are at least $N/4$ vertices $v \in G\restriction\pi(\tau)$ for which there is a $v$-critical assignment for $\mathrm{Peb}(G\restriction\pi(\tau))$. If $\phi$ is such a critical assignment, then $\pi(\tau)\cup\phi$ is a $v$-critical assignment for $\mathrm{Peb}(G)$. Let $A$ be the set of all nodes in $G$ satisfying this condition, and $B$ the set of variables queried in the path from the root of $\mathcal{R}$ to $C$. Since $|A|, |B| \geq \lfloor N/4 \rfloor$, by Lemma 3.4, setting $d = c/4$, $|\rho(\mathrm{Clauses}(A)) \cap B| \geq N/8$.

Let $x$ be a variable in $\rho(\mathrm{Clauses}(A)) \cap B$. We claim that $x$ must occur in $C$. Suppose not. By assumption, there is a $D \in \mathrm{Clauses}(v)$, for some $v \in A$, so that $\rho(D) = x$, and $D \vee x, D \vee \overline{x} \in \Pi$. Since $x$ was queried at some point in the path from the root of $\mathcal{R}$ to $C$, either $\tau(x) = 0$ or $\tau(x) = 1$. Let's assume that $\tau(x) = 0$ (the case $\tau(x) = 1$ is symmetrical). By Lemma 3.3, there is a $D$-critical assignment $\phi$ for $\mathrm{Peb}^*(G)^{k\oplus}$ that extends $\tau$, and so is a $D \vee x$-critical assignment for $\Pi$. Extend the path in $\mathcal{R}$ from $C$ by answering queries in accordance with $\phi$. This path must terminate in a node labelled with $D \vee x$. But since $x$ does not occur in $C$, it follows that it must have been resolved on twice along the path, violating regularity. This contradiction proves that $x$ must occur in $C$, showing that $C$ contains at least $N/8$ literals. □

**4.3. Lower Bound.** In this final section, we bring together the earlier lemmas to prove the main result of the paper.

THEOREM 4.3. *Let $\Pi_1, \Pi_2, \ldots, \Pi_i, \ldots$ be the sequence of contradictory sets of clauses based on the pointed graphs $G_1, G_2, \ldots, G_i, \ldots$ of Paul, Celoni and Tarjan, where $n = n(i)$ is the size of the graph $G_i$. Then:*

1. *There are resolution refutations of $\Pi_i$ with size $O(n(\log n)^5)$.*
2. *Every regular resolution refutation of $\Pi_i$ has size $2^{\Omega(n/[(\log n)^2 \log\log n])}$*

**Proof.** We can construct a short refutation of the set of clauses $\mathrm{Peb}(G)$ by imitating a pebbling strategy for $G$. That is to say, starting from the unit clauses corresponding to the source nodes in $G$, we can successively deduce the unit clauses corresponding to the pebbles placed in a strategy until the root is reached, resulting in the empty clause.

The refutation constructed in this way is a unit resolution proof (where at least one of the premisses in every resolution step is a unit clause), with $O(n)$ steps and in which every clause contains at most three literals. We can imitate this refutation to produce a refutation of $\mathrm{Peb}(G)^{k\oplus}$; in this refutation, a single resolution step in the original refutation corresponds to multiple resolution steps in the new refutation. Let

us suppose that in the original refutation of $\mathrm{Peb}(G)$, the clause $b \vee c$ was inferred from $a$ and $\overline{a} \vee b \vee c$, where $a, b, c$ are literals. Then in the new refutation, we infer $\{b \vee c\}^{k\oplus}$ from $\{\overline{a} \vee b \vee c\}^{k\oplus}$ and $\{a\}^{k\oplus}$.

The set of clauses $\{a\}^{k\oplus} \cup \{\overline{a}\}^{k\oplus}$ consists of all the clauses in a fixed set of $k$ variables, so it takes $O(2^k) = O((\log n)^{5/3})$ steps to deduce the empty clause from this set. Hence, a single clause in $\{b \vee c\}^{k\oplus}$ can be derived in $O((\log n)^{5/3})$ steps. There are $2^{2\lceil \delta \log\log n \rceil} = O((\log n)^{10/3})$ clauses in $\{b \vee c\}^{k\oplus}$. It follows that the derivation of $\{b \vee c\}^{k\oplus}$ takes $O((\log n)^{10/3}(\log n)^{5/3}) = O((\log n)^5)$ resolution steps, showing that the entire refutation of $\mathrm{Peb}(G)^{k\oplus}$ has size $O(n(\log n)^5)$.

To complete the proof of the first half of the theorem, we need to show how to produce a short proof of the clauses in $\mathrm{Peb}(G)^{k\oplus}$ from $\Pi$. The first step is to deduce all of the clauses $C$ in $\mathrm{Peb}^*(G)^{k\oplus}$ from $C \vee \rho(C)$ and $C \vee \neg\rho(C)$; this takes $O(n(\log n)^5)$ steps.

We still have to deduce the clauses in $\mathrm{Peb}(G)^{k\oplus}$ from $\mathrm{Peb}^*(G)^{k\oplus}$. If $v$ is a source node, then $\mathrm{Clauses}(v)$ is the set of four clauses $\{v \vee x_v \vee y_v, v \vee \overline{x_v} \vee y_v, v \vee x_v \vee \overline{y_v}, v \vee \overline{x_v} \vee \overline{y_v}\}$. We can deduce the clauses $v \vee x_v$, $v \vee \overline{x_v}$ and hence $v$ in three resolution inferences. Consequently, we can deduce $v^{k\oplus}$ from $\mathrm{Clauses}(v)^{k\oplus}$ in $O((\log n)^5)$ steps; the analysis is the same as in the previous paragraph. A similar but simpler analysis applies to the sink node, so the entire resolution derivation of $\mathrm{Peb}(G)^{k\oplus}$ from $\mathrm{Peb}^*(G)^{k\oplus}$ takes $O(n(\log n)^5)$ steps.

Thus we have shown that there is a refutation of $\Pi$ containing $O(n(\log n)^5)$ resolution inferences. This upper bound is optimal (apart from the constant), since $\Pi$ contains $O(n(\log n)^5)$ clauses.

For the second part of the theorem, let us assume that $\mathcal{R}$ is a regular resolution refutation of $\Pi$, with size less than $2^{N/[3840(\log n)\log\log n]}$. By Lemma 4.1 there is a pebbling assignment $\mu$ so that $\mathcal{R}\restriction\mu$ contains no large clauses, but $G\restriction\pi(\mu)$ has pebbling number at least $N/2$. However, Lemma 4.2 shows that $\mathcal{R}\restriction\mu$ must contain a large clause, showing that a regular refutation of this size cannot exist. $\square$

It is interesting to ask how close Theorem 4.3 comes to the optimum. We already observed in §1 that if $R$ is the minimum size of a resolution refutation of a set of clauses, then the size of a regular refutation is bounded above by $2^{O(R\log\log R/\log R)}$. If we define $S = n(\log n)^5$, then we find that the lower bound on regular refutations has the form $2^{\Omega(S/(\log S)^7 \log\log S)}$. So, the separation we have proved is certainly much closer to the optimum than previous bounds, but there is definitely room for improvement.

I would like to express my thanks to Allen van Gelder, whose comments on the earlier version resulted in a substantially better paper. In addition, the comments of two anonymous referees resulted in further significant improvements.

## REFERENCES

[1] M. Alekhnovich, J. Johannsen, T. Pitassi, and A. Urquhart, *An exponential separation between regular and general resolution*, Theory of Computing, 3 (2007), pp. 81–102. Preliminary version in Proceedings of the 34th Annual ACM Symposium on Theory of Computing: May 19-21 2002, Montréal, Québec, Canada.

[2] N. Alon and J. H. Spencer, *The Probabilistic Method*, John Wiley, 1992.

[3] E. Ben-Sasson, *Size Space Tradeoffs For Resolution*, Proceedings of the 34th ACM Symposium on the Theory of Computing, (2002), pp. 457 – 464.

[4] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson, *Near optimal separation of tree-like and general resolution*, Combinatorica, (2004), pp. 585–603. Preliminary version, ECCC TR00-005, 2000.

[5] B. Bollobás, *Modern Graph Theory*, Springer-Verlag, 1998. Graduate Texts in Mathematics 184.

[6] M. L. Bonet and N. Galesi, *Optimality of size-width tradeoffs for resolution*, Computational Complexity, 10 (2001), pp. 261–276. Preliminary version: Proceedings 40th FOCS, 1999.

[7] M. Davis, G. Logemann, and D. Loveland, *A machine program for theorem proving*, Communications of the Association for Computing Machinery, 5 (1962), pp. 394–397. Reprinted in [14], Vol. 1, pp. 267-270.

[8] A. Goerdt, *Regular resolution versus unrestricted resolution*, SIAM Journal on Computing, 22 (1993), pp. 661–683.

[9] W. Huang and X. Yu, *A DNF without regular shortest consensus path*, SIAM Journal on Computing, 16 (1987), pp. 836–840.

[10] A. Kamath, R. Motwani, K. Palem, and P. Spirakis, *Tail bounds for occupancy and the satisfiability threshold conjecture*, Random Structures and Algorithms, 7 (1995), pp. 59–80.

[11] B. Krishnamurthy, *Short proofs for tricky formulas*, Acta Informatica, 22 (1985), pp. 253–275.

[12] C. McDiarmid, *Concentration*, in Probabilistic Methods for Algorithmic Discrete Mathematics, M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, eds., Springer, 1998, pp. 195–248. Algorithms and Combinatorics 16.

[13] W. Paul, R. Tarjan, and J. Celoni, *Space bounds for a game on graphs*, Mathematical Systems Theory, 10 (1977), pp. 239–251.

[14] J. Siekmann and G. Wrightson, eds., *Automation of Reasoning*, Springer-Verlag, New York, 1983.

[15] G. Stålmarck, *Short resolution proofs for a sequence of tricky formulas*, Acta Informatica, 33 (1996), pp. 277–280.

[16] G. Tseitin, *On the complexity of derivation in propositional calculus*, in Studies in Constructive Mathematics and Mathematical Logic, Part 2, A. O. Slisenko, ed., Consultants Bureau, New York, 1970, pp. 115–125. Reprinted in [14], Vol. 2, pp. 466-483.

[17] A. Urquhart, *The complexity of propositional proofs*, The Bulletin of Symbolic Logic, 1 (1995), pp. 425–467.

[18] ———, *Regular and general resolution: An improved separation*, in Theory and Applications of Satisfiability Testing – SAT 2008, H. K. B. annd Xishun Zhao, ed., Springer-Verlag, 2008, pp. 277–290. Lecture Notes in Computer Science 4996.