The Complexity of Resolution Refinements

Joshua Buresh-Oppenheim
Computer Science Department
University of Toronto
Toronto, ON
bureshop@cs.toronto.edu

Toniann Pitassi*
Computer Science Department
University of Toronto
Toronto, ON
toni@cs.toronto.edu

Abstract

Resolution is the most widely studied approach to propositional theorem proving. In developing efficient resolution-based algorithms, dozens of variants and refinements of resolution have been studied from both the empirical and analytic sides. The most prominent of these refinements are: DP (ordered), DLL (tree), semantic, negative, linear and regular resolution. In this paper, we characterize and study these six refinements of resolution. We give a nearly complete characterization of the relative complexities of all six refinements. While many of the important separations and simulations were already known, many new ones are presented in this paper; in particular, we give the first separation of semantic resolution from general resolution. As a special case, we obtain the first exponential separation of negative resolution from general resolution. We also attempt to present a unifying framework for studying all of these refinements.

1. Introduction

The satisfiability problem for boolean formulas in conjunctive normal form is one of the most important problems in theoretical computer science. Its complementary problem, propositional theorem proving (find a proof of unsatisfiability for an unsatisfiable formula) plays an important role in many areas including artificial intelligence and model checking.

There has been a tremendous amount of research aimed at understanding the satisfiability problem and developing algorithms for satisfiability testing and propositional theorem proving. Much of this research has centered around the method of *resolution*, invented by Robinson [21] in the 1960's. Resolution is a sound and complete refutation system that can be applied to both propositional as well as to higher order theorem proving. Resolution is the most widely studied approach to propositional theorem proving and to satisfiability testing. There is a large body of research exploring *resolution algorithms*—these are algorithms that take as input an unsatisfiable CNF formula F and output a resolution refutation of F. Most satisfiability algorithms studied in the literature are resolution-based. That is, they take as input a formula F and output a resolution refutation of F if and only if F is unsatisfiable.

In developing efficient resolution algorithms, dozens of variants and refinements of resolution have been studied, vfrom both the empirical and analytical sides to determine which particular variant is most efficient for both propositional as well as for higher order theorem proving. The most prominent of these variants are: Davis-Putnam (ordered), DPLL (tree), semantic, negative, linear and regular resolution. Resolution was actually pre-dated by the Davis-Putnam and DPLL procedures which are still the most widely used in propositional theorem proving. The general idea of these procedures is to convert a problem on n variables to problems on one less variable by eliminating a variable. The former [9], which we call DP (Davis-Putnam) or ordered resolution, does this by applying all possible uses of the resolution rule on a given variable to eliminate it. The latter [8] which we call DLL and is the form used today, branches based on the possible truth assignments to a given variable. Although at first DPLL does not look like resolution, an easy argument shows that it is equivalent to the special class of tree-like resolution proofs. As a proof system, resolution is strictly stronger than DP [13, 1] which is strictly stronger than DPLL [23]. However, DPLL is still the most popular, because of its proof search properties.

The regularity restriction was first introduced by Tseitin in an important article [22], the published version of a talk given in 1966. A regular resolution refutation is a resolution refutation such that along each path from the root to a

* Research partially supported by NSF grant CCR-9457782, US-Israel BSF Grant 95-00238, and an NSERC University Faculty Award.

	Neg	Sem	Lin	Ord	Reg	Tree
Neg	Yes	No †	No †	No †	No †	Yes
Sem	Yes	Yes	No †	No †	No †	Yes
Lin	Yes*	Yes*	Yes	Yes*	Yes*	Yes
Ord	No	No	No †	Yes	No	No
Reg	No †	No †	No †	Yes	Yes	Yes
Tree	No †	No †	No	No	No	Yes

Table 1. The relative power of six resolution refinements

leaf (in the underlying directed acyclic graph), each variable is resolved upon at most once. This restriction is very natural in the sense that many algorithms such as DPLL and DP are special cases of regular resolution proofs. The main result of Tseitin's paper is an exponential lower bound for regular resolution proofs of contradictory formulas based on graphs. Subsequently researchers tried to extend Tseitin's result to general resoltuion by showing that regular resolution can simulate general resolution efficiently. However, [13, 1] showed that such a simulation is impossible by giving an exponential separation between regular and general resolution. There was a fifteen year gap before the first superpolynomial lower bound for unrestricted resolution was obtained by Haken [14].

Another refinement that defied analysis is linear resolution. This restriction, where the underlying directed acyclic graph must be linear, is the basis for procedural semantics for Horn logic programs, and underlies effective implementations strategies such as Prolog.

Finally, Negative resolution is a special case of the more general family of semantic resolution refinements that can efficiently simulate hyper-resolution. In the same way that resolution underlies standard algorithms for propositional satisfiability testing, hyper-resolution can be seen to underlie the standard algorithms for a more general class of constraint satisfaction problems.

In this paper, we study these popular refinements of resolution. We give a nearly complete characterization of the relative complexity of all six refinements discussed above. As mentioned earlier, many of the important separations and simulations were already known, although several are new to this paper. Table 1 summarizes the characterization: the cell at row i and column j answers the question "does refinement i p-simulate refinement j?" Every "No" entry means that not only does refinement i not p-simulate refinement j, but there is an exponential separation between the two. The symbol †indicates a new result, as far as we know. "Yes*" is a qualified "Yes" entry, which we shall explain later. For precise definitions of the refinements and of p-simulation, see section 2.1.

We attempt to present a unifying framework for analyzing these various refinements. We use only three fundamental combinatorial principles to witness all of the abovementioned separations. Aside from nearly completing the picture of the many resolution refinements, we hope that the proofs from our work and earlier work may yield further insight for developing algorithms for satisfiability testing and propositional theorem proving.

2. Definitions

2.1. Resolution and its refinements

The *resolution principle* says that if C and D are clauses and x is a variable, then any assignment that satisfies both of the clauses $C \vee x$ and $D \vee \bar{x}$ also satisfies $C \vee D$. The clause $C \vee D$ is said to be a *resolvent* of the clauses $C \vee x$ and $D \vee \bar{x}$ derived by *resolving on* the variable x. A *resolution derivation* of a clause C from a CNF formula F consists of a sequence of clauses in which each clause is either a clause of F, or is a resolvent of two previous clauses, and C is the last clause in the sequence; it is a *refutation* of F if C is the empty clause Λ.

The *size* of a refutation is the number of resolvents in it. The *width* of a clause is the number of literals occurring in the clause. The width of a refutation is the maximum width of all clauses occurring in the refutation.

We can represent a resolution refutation as a directed acyclic graph (dag) where the nodes are the clauses in the refutation, each clause of F has out-degree 0, and any other clause has two arcs pointing to the two clauses that produced it. The clause Λ is the only source in this graph. The arcs pointing to $C \vee x$ and $D \vee \bar{x}$ are labeled with the literals x and $\bar{x}$ respectively. It is well known that resolution is a *sound* and *complete* propositional proof system, i.e., a formula F is unsatisfiable if and only if there is a resolution refutation for F.

A *negative resolution* refutation of F is a resolution refutation with the additional restriction that all resolutions must be negative. A resolution step $C \vee x$ and $D \vee \bar{x}$ implies $C \vee D$ is negative whenever D contains only negative literals. Negative resolution is also called *negative hyperresolution*. *Positive resolution* is the dual of negative resolution, where one of the two premises in each resolution step must contain only positive literals. More generally, given a formula F over n variables and an assignment $\alpha \in \{0,1\}^n$, an α-refutation of F is a resolution refutation such that, when two clauses are resolved, at least one of them must be falsified by α. A refutation of F is called *semantic* if it is a α-refutation for any $\alpha \in \{0,1\}^n$.

A *linear resolution* refutation of F is a resolution refutation with the additional restriction that the underlying dag must be linear. That is, the proof consists of a sequence of clauses $C_1, C_2, \ldots, C_m$ such that C_m is the empty clause, for every $1 \leq i \leq m$, either C_i is an initial clause, or C_i is derived from C_{i-1} and an initial clause, or C_i is derived from C_{i-1} and C_j, for some $j < i-1$.

A *regular resolution* refutation of F is a resolution refutation such that on any path from Λ to a clause in F, no variable appears more than once as an arc-label. We call a regular resolution refutation *ordered* if every sequence of variables labelling a path from Λ to a clause in F respects the same ordering on the variables. Ordered resolution is also called *Davis-Putnam*.

Finally, a *tree-like resolution* refutation is one in which the underlying dag is a tree. Tree-like resolution is also called *Davis-Logemann-Loveland*.

All of these restricted forms of resolution are known to be sound and complete. They were each defined in an effort to facilitate the following problem: given an unsatisfiable F, find a refutation of F. The restrictions on the form of the refutation serve to narrow the search-space of this problem. We explore the question of how these restrictions compare with each other in terms of the sizes of the minimal refutations that can be found within them. To be more precise, we use the notion of p-simulation:

DEFINITION 2.1: For two resolution refinements (or, in general, two proof systems) R_1 and R_2, we say R_1 p-simulates R_2 if there is a polynomially computable function f such that for any R_2-refutation π of the formula F, $f(\pi)$ is an R_1-refutation of F.

2.2. Hard Formulas

We present three major categories of unsatisfiable formulas that yield all the separations listed above. Each category is based on a simple combinatorial principle whose negation can be easily encoded in propositional logic. For some of the lower bounds, the direct encodings need to be modified in a syntactically-inspired way so that they are still unsatisfiable, but harder to refute.

The first category encodes implication in directed, acyclic graphs (dags). The formulas were originally introduced by [17]. More recently, they were used by Raz and McKenzie [20], and subsequently in [4, 3, 6]. Let G be a directed, acyclic graph, with bounded fan-in (often 2), n vertices, and a single sink vertex.

The implication graph formulas encode the following contradictory statement: "All of the source vertices are colored red, the sink is colored blue, and if all the predecessors of a vertex are red, so is the vertex itself."

The formula associated with G, $IMP(G)$ has one variable, x_i, for every node i in G, and the following clauses:
(1) for each source node i in G, (x_i);
(2) for the sink node s in G, $(\bar{x}_s)$;
(3) for every set of nodes $i_1, \ldots, i_k, j$, such that the edges $(i_1, j), \ldots, (i_k, j)$ are present in G, we have the clause $(\bar{x}_{i_1} \vee \ldots \vee \bar{x}_{i_k} \vee x_j)$.

The natural way to refute the above formula/clauses is to begin at the source vertices, and derive successively that each layer of vertices must be true, until finally we can conclude that each sink vertex must be true. This gives us the desired contradiction since the sink vertex is false. For any such graph G, this natural refutation can be formalized as a linear-size tree-like, ordered resolution refutation.

We also define a more general formula, $IMP'(G)$, which was introduced by [3]. Now there are two variables $x_{i,0}$ and $x_{i,1}$ associated with a vertex i in G. The formula is the following conjunction of clauses:
(1) for each source vertex i in G, $(x_{i,0} \vee x_{i,1})$;
(2) for the sink vertex s in G, $(\bar{x}_{s,0})$ and $(\bar{x}_{s,1})$;
(3) for every $i_1, \ldots, i_k, j$ such that $(i_1, j), \ldots, (i_k, j)$ are edges in G, we have the following clauses stating that if one of the variables associated with each of the i's is true, then one of the variables associated with j is true. I.e. for each $(a_1, \ldots, a_k) \in \{0,1\}^k$: $(\bar{x}_{i_1,a_1} \vee \ldots \bar{x}_{i_k,a_k} \vee x_{j,0} \vee x_{j,1})$.

Another source of hard formulas comes from the principle that any total ordering on n elements must have a maximum. The first version was introduced by [18] and we'll call it GT_n: let $X = \{x_{ij} \mid i, j \in [n], i \neq j\}$. We assert the following axioms:
(1) for each i, j distinct, $x_{ij} = \bar{x}_{ji}$;
(2) for i_1, i_2, i_3 distinct, $x_{i_1 i_2} \wedge x_{i_2 i_3} \supset \bar{x}_{i_3 i_1}$;
(3) for each j, $\bigvee_{k \in [n], k \neq j} x_{kj}$.

[1] introduced a variation on GT_n, which we'll call GT'_n: let X be as above. Fix an arbitrary enumeration (with repetitions) on X using the pair of $i(r), j(r)$ such that $X = \{x_{i(r)j(r)} \mid r \in \{0, \ldots, n^2 - 1\}\}$. We have the axioms:
(1) for i, j distinct, $x_{ij} = \bar{x}_{ji}$;
(2) for each j, $\bigvee_{k \in [n], k \neq j} x_{kj}$;
(3) for i_1, i_2, i_3 distinct and and $r = n(i_1 + i_2) + i_1 + i_3 \bmod n^2$, $\bar{x}_{i_1 i_2} \vee \bar{x}_{i_2 i_3} \vee \bar{x}_{i_3 i_1} \vee x_{i(r)j(r)}$ and $\bar{x}_{i_1 i_2} \vee \bar{x}_{i_2 i_3} \vee \bar{x}_{i_3 i_1} \vee \bar{x}_{i(r)j(r)}$.

The final principle was introduced by [4]. It states that, given a supply of m pearls, each of which are colored either red or blue, if n pearls are chosen and placed in a line with the first one red and the last one blue, there must be a blue pearl that directly follows are red pearl on the line. More formally, let p_{ij} denote pearl j is at position i on the line and let r_j denote that pearl j is colored red. SP_{nm} is the clauses:

(1) for each i, $\bigvee_{j=1}^{m} p_{ij}$;
(2) for each k and each i, j distinct, $\bar{p}_{ik} \vee \bar{p}_{jk}$;
(3) for each i and k, j distinct, $\bar{p}_{ij} \vee \bar{p}_{ik}$;
(4) for each j, $p_{1j} \rightarrow r_j$;
(5) for each j, $p_{nj} \rightarrow \bar{r}_j$;
(6) for each i and j, k distinct, $p_{ij} \wedge r_j \wedge p_{i+1\,k} \rightarrow r_k$.

Again, the direct statement of the combinatorial principle is not enough to get the lower bounds we need. SP_{nm} can be modified to a CNF SP'_{nm}, similarly to the way GT_n is modified, so that it can be made harder to refute.

3. Operations on formulas and proofs

DEFINITION 3.1: For a given resolution refinement and a CNF formula F, the width of F, $width(F)$, is the width of the minimum-width refutation of F in that resolution refinement. The size of F, $size(F)$, is the size of the minimum-size refutaton of F in that resolution refinement.

DEFINITION 3.2: An *assignment* for a formula F (sometimes called a *restriction*) is a Boolean assignment to some of the variables in the formula; the assignment is *total* if all the variables in the formula are assigned values. If C is a clause, and ρ an assignment, then we write $C\lceil\rho$ for the result of applying the assignment to C, that is, $C\lceil\rho = 1$ if $\rho(l) = 1$ for some literal l in C, otherwise, $C\lceil\rho$ is the result of removing all literals set to 0 by ρ from C (with the convention that the empty clause is identified with the Boolean value 0). If F is a CNF formula, then $F\lceil\rho$ is the conjunction of all the clauses $C\lceil\rho, C$ a clause in F. If $\pi = C_1, \ldots, C_m$ is a resolution derivation, then $\pi\lceil\rho$ is the subsequence of $C_1\lceil\rho, \ldots, C_m\lceil\rho$ where $C_i\lceil\rho$ is removed if it is identically 1.

Lemma 1: Let F be a k-CNF and let π be a semantic (respectively negative, regular, ordered, tree-like) resolution refutation of F. Let ρ be a restriction to the variables of F. Then $\pi\lceil\rho$ contains a semantic (respectively negative, regular, ordered, tree-like) resolution refutation of $F\lceil\rho$.

DEFINITION 3.3: Let β be an assignment to all the variables of a set of formulas $\mathcal{F}$. The set of formulas $\beta(\mathcal{F})$ is the same as $\mathcal{F}$ except that if $\beta(x) = 1$, then all literals x are replaced by $\bar{x}$ and (simultaneously) all literals $\bar{x}$ are replaced by x.

The following equivalence is straight-forward in linear, regular, ordered and tree-like resolution. While semantic resolution is somewhat sensitive to the renaming of variables, formulas that are hard for all α remain hard under renaming. It does not hold in negative resolution.

Lemma 2: Let F be a CNF over n variables and let $\beta \in \{0,1\}^n$. In semantic resolution, $width(\beta(F)) = width(F)$ and $size(\beta(F)) = size_{sem}(F)$.

Proof: Let π be an α-refutation of F. Let $\beta(\alpha)$ be the assignment $\beta \oplus \alpha$. Then $\beta(\pi)$ is a $\beta(\alpha)$-refutation of $\beta(F)$ with the same size and width as π. Hence $width(\beta(F)) \leq width(F)$ and $size(\beta(F)) \leq size(F)$. A dual argument shows the opposite inequalities. $\square$

The following operation on CNFs is due to Alekhnovich and Razborov and was used by [2]. We shall see that it is ideal for transforming a formula of large width into a formula of large size. Note that in what follows we use x^1 to denote the literal x and x^0 to denote the literal $\bar{x}$.

DEFINITION 3.4: Let F be a CNF formula over a set of variables $\{x_1, x_2, \ldots, x_n\}$. Let $\{x'_1, x'_2, \ldots, x'_n\}$ be a disjoint set of variables. The *xorification* of a literal x_i^a, for $a \in \{0,1\}$ is the formula $XOR(x_i^a) = x_i \oplus x'_i \oplus a$. The xorification of a clause $C = \bigvee_{i \in I} \ell_i$, is the CNF equivalent to $\bigvee_{i \in I} XOR(\ell_i)$. The xorification of F, $XOR(F)$, is the conjunction of the xorifications of each clause in F.

If F is an unsatisfiable k-CNF with m clauses, then $XOR(F)$ is an unsatisfiable $2k$-CNF with at most $2^k m$ clauses. Furthermore, let ρ be a partial assignment that, for each i, assigns a value to exactly one of the two variables x_i, x'_i. Then $XOR(F)\lceil\rho$ is equivalent (up to renaming of variables) to $\beta(F)$, where β is some assignment to the variables not set by ρ. Alekhnovich and Razborov use this fact to prove the following result for general resolution. Perhaps even more remarkable is that this property holds in many restricted forms of resolution, some of which do not admit general size-width tradeoffs as general resolution does.

Lemma 3: If F is a k-CNF, π_1 is a minimum-width semantic (respectively, regular, ordered, tree-like) resolution refutation of F, and π_2 is a minimum-size semantic (respectively, regular, ordered, tree-like) resolution refutation of $XOR(F)$, then $size(\pi_2) \geq exp(\Omega(width(\pi_1) - k))$.

Proof: Let $w = width(\pi_1)$. Assume π_2 has size S. Let $x_1, \ldots, x_n$ be the variables of F. Peform the following probabilistic experiment: let ρ be an assignment that, for each i, chooses a variable uniformly from $\{x_i, x'_i\}$ and a value uniformly from $\{0,1\}$, and assigns the chosen value to the chosen variable. If C is a clause of width at least w in π_2, then C appears in $\pi_2\lceil\rho$ with probability at most $(3/4)^w$. The expected number of such wide clauses that remain in $\pi_2\lceil\rho$ is at most $S(3/4)^w$, since there are S clauses in total in π_2. If $S < (4/3)^w$, then this quantity is less than 1, so there must exist a ρ that eliminates all the wide clauses. So (by lemma 1) $\pi_2\lceil\rho$ is a semantic (respectively, regular, ordered, tree-like) refutation of $XOR(F)\lceil\rho$ with width less than w (if $k < w$). By lemma 2, this gives a semantic (respectively, regular, ordered, tree-like) refutation of F with width less than

w, which contradicts the assumption that π_1 has minimum width. □

DEFINITION 3.5: Let $F_0, \ldots, F_{\ell-1}$ be k-CNFs over the set of variables $\{x_1, \ldots, x_n\}$ and let $\ell = 2^c$. Let $Y = \{y_0, \ldots, y_{c-1}\}$ be a disjoint set of variables. For $0 \le i < \ell$, let $b(i)$ be the interpretation of i as a bit-string of length c and let $b(i)(j)$ be the j-th bit in this string. Let F_i' be the set of clauses

$$\{C \vee y_0^{b(i)(0)} \vee \ldots \vee y_{c-1}^{b(i)(c-1)} \mid C \in F_i\}.$$

Finally, define $join(F_0, \ldots, F_{\ell-1}, Y)$ as the set of clauses $\bigcup_{i=0}^{\ell-1} F_i'$.

Lemma 4: Let $F_0, \ldots, F_{\ell-1}$ be k-CNFs over the set of variables $\{x_1, \ldots, x_n\}$ and let $\ell = 2^c$. Let $Y = \{y_0, \ldots, y_{c-1}\}$ be a disjoint set of variables. Let F be $join(F_0, \ldots, F_{\ell-1}, Y)$. In general (respectively, semantic, negative, regular, tree-like) resolution, the following relationships hold: $width(F) \ge \max_{i=0}^{\ell-1}\{width(F_i)\}$ and $size(F) \ge \max_{i=0}^{\ell-1}\{size(F_i)\}$.

Proof: Consider a refutation π of $join(F_0, \ldots, F_{\ell-1}, Y)$ in any of the mentioned resolution refinements. Let ρ_i be an assignment to Y that sets y_j to $1 - b(i)(j)$ for each $0 \le j < c$. Then, by lemma 1, $\pi\lceil\rho_i$ is a refutation of F_i in the given resolution refinement. Certainly, $width(\pi) \ge width(\pi\lceil\rho_i)$ and $size(\pi) \ge size(\pi\lceil\rho_i)$. □

In most refutation systems, it doesn't help to use tautological formulas. This is true for most of the resolution refinements we consider, but not necessarily for linear, which is why we have definition 3.6.

Lemma 5: Let π be a general (respectively semantic, negative, regular, ordered, tree-like) resolution refutation of a contradictory CNF F of size S. Let F' be the same as F with any clauses containing both a variable and its negation removed. There is a general (respectively semantic, negative, regular, ordered, tree-like) refutation of F' of size at most S.

DEFINITION 3.6: Let F be a CNF over the variables $x_1, \ldots, x_n$. Let G be the following set of $2n^2$ clauses:

$$\{x_i \vee \bar{x}_i \vee x_j^a \mid 1 \le i, j \le n, a \in \{0, 1\}\}.$$

Define $ADDTAUT(F)$ to be the CNF $F \cup G$.

3.1. Graphs with high pebbling measure

DEFINITION 3.7: Let $G = (V, E)$ be a directed, acyclic graph. A configuration is a subset of V. A legal pebbling of a set of vertices X in G is a sequence of configurations, the first being the empty set and the last being X and in which each configuration C' follows from the previous configuration C by one of the following rules: (1) v can be added to C to get C' if all immediate predecessors of v are in C; (2) Any vertex can be removed from C to obtain C'. The *complexity* of a legal pebbling of X is the size of the largest configuration in the sequence. The *pebbling number of a graph G* with a single sink vertex s is the minimal number n such that there exists a legal pebbling of s with complexity n.

Cook [7] showed that a family of pyramid graphs, $\{Pyramid_n\}$, with $n = m + (m-1) + \cdots + 1$ underlying vertices have pebbling measure $\Omega(\sqrt{n})$. These are layered graphs, consisting of m layers, with m source vertices at layer 0, labelled $x_1^0, x_2^0, \cdots, x_m^0$, $m-1$ vertices at layer 1, labelled $x_1^1, \cdots x_{m-1}^1$, and so on with one sink vertex, x_1^{m-1} at layer $m-1$. All nonsource vertices have indegree 2, and in general x_j^{i+1} has parents x_j^i and x_{j+1}^i. We call a layered dag on n nodes *pyramid-like* if it is on the same set of layers and nodes as $Pyramid_n$.

DEFINITION 3.8: A bipartite graph from m_1 inputs to m_2 outputs is called r-expanding, for $1 \le r \le m_1$, if, for every subset X of the inputs of size at most $\lceil m_1/r \rceil$, the size of the neighbor set $\Gamma(X)$ is greater than $|X|$.

Let G be a layered, directed graph with layers $0, \ldots, k-1$. All edges from layer i point to layer $i+1$. Call G an r-expanding, layered graph if, for each $i > 0$, the bipartite graph from layer i to layer $i-1$ is r-expanding.

Lemma 6: Let G be an r-expanding, layered graph on k layers such that layer i contains at least $r(k-i)$ nodes. Then, any node in layer $k-1$ has pebbling measure k.

Proof: Let t be any node in layer $k-1$. In any pebbling of t, there must be a configuration where all paths from a source to t have a pebble on them. Consider the first such configuration C_i. In C_{i-1} there must be a path p from a source s to t that does not contain a pebble. The next move must be to place a pebble on s since every other node in p has an unpebbled predecessor. Let $p = (s = p_0, p_2, \ldots, p_{k-1} = t)$. Consider a maximal set of vertex-disjoint paths from the sources to a node in p. All of these paths must have a pebble on them in configuration C_i since they each represent a path from a source to t. Hence the size of any such set is a lower bound on the pebbling measure of t. We argue that there are at least k paths in one of these sets. Let $X_{k-1} = \{p_{k-1}\}$. Since G is expanding, p_{k-1} must have a predecessor other than p_{k-2}. Call it $pred(p_{k-1})$. Let $X_{k-2} = \{pred(p_{k-1}), p_{k-2}\}$. In general, for $i > 1$, let X_{k-i}, be a subset of size i of layer $k-i$. Since $|X_{k-i}|$ is at most $1/r$ times the size of layer $k-i$ and since G is expanding, Hall's theorem guarantees that X_{k-i} can be matched into its set of predecessors minus p_{k-i-1}. Let X_{k-i-1} be this set of matched nodes unioned with p_{k-i-1}. The collection of all these $k-1$ matchings form vertex-disjoint paths from the sources to $\{p_1,\ldots,p_{k-1}\}$. The node p_0 forms the last vertex-disjoint path. □

DEFINITION 3.9: Let $Pyr(n,d)$ be a distribution of pyramid-like graphs where each node in layer i, $1 \leq i \leq m$, has parents chosen as follows: choose d nodes randomly and independently with replacement from layer $i-1$ and identify multiple copies of each chosen node.

4. Simulations

We start with some p-simulations that have been known for quite some time.

Lemma 7: Negative resolution p-simulates tree-like resolution. Hence, semantic resolution p-simulates tree-like resolution.

Lemma 8: ([16, 24]) Linear resolution p-simulates tree-like resolution.

Lemma 9: Regular resolution p-simulates tree-like resolution.

The following lemma says that, modulo some minor syntactic issues, linear resolution is as powerful as general resolution. A similar phenomenon was noted by [10].

Lemma 10: Let F be an unsatisfiable CNF such that there is a resolution refutation of size S and width w. Then there is a linear resolution refutation of $ADDTAUT(F)$ of size $O(S \cdot w)$.

Proof: Let $\pi = C_1,\ldots,C_S = \Lambda$ be the resolution refutation of S. Since $C_1 \in F$, we can clearly derive C_1 in linear resolution in size w. Now, assume we have a linear resolution derivation L of size $i \cdot w$ that ends with C_i and includes $C_1,\ldots C_{i-1}$ in order along the line. Assume C_{i+1} is derived from C_j, C_k in π by resolving on x, where $1 \leq j < k \leq i$. If $i = k$, then we can simply add C_{i+1} to the end of L. Otherwise, let $\ell_1,\ldots,\ell_w$ be the literals in C_i. Resolve C_i with the axioms $(x \vee \bar{x} \vee \bar{\ell}_1),\ldots,(x \vee \bar{x} \vee \bar{\ell}_2)$ until the last clause in L is $(x \vee \bar{x})$. Now resolve this last clause with C_j and C_k, so the last clause becomes C_{i+1}. Now assume alternatively that C_{i+1} is an axiom that contains the literal x^a: derive the clause $(x \vee \bar{x})$ as above from C_i and simply resolve the axiom C_{i+1} with it to obtain C_{i+1} at the end of the line. □

Lemma 11: If there is a superpolynomial separation between negative (respectively, semantic, ordered, regular, tree-like) resolution and general resolution, then there is a superpolynomial separation between negative (respectively, semantic, ordered, regular, tree-like) and linear resolution.

Proof: Let $\{F_n\}$ be the family of CNFs that exhibits the separation of negative (respectively, semantic, ordered, regular, tree-like) resolution from general resolution. Assume general resolution refutes F_n in size $S(n)$. Then linear resolution refutes $ADDTAUT(F_n)$ in size at most $2nS(n)$. If there were some polynomial p such that negative (respectively, semantic, ordered, regular, tree-like) resolution refutes $ADDTAUT(F_n)$ in size $p(2nS(n))$, then it would refute F_n in size $p(2nS(n))$ by lemma 5, which contradicts the superpolynomial separation. □

5. Upper Bounds

Not only are all of these restricted forms of resolution complete, but there is a general upper bound on the size of their refutations:

Lemma 12: For any contradictory CNF F over n variables and any ordering of the variables, there is an ordered (respectively, positive) refutation of F of size $2^{O(n)}$.

Lemma 13: Let G be a dag on n nodes. Let j be a node in G with parents $i_1,\ldots,i_k$, such that $k = O(\log n)$. Consider derivations of the clause $x_{j,0} \vee x_{j,1}$ from the clauses $(x_{i_1,0} \vee x_{i_1,1}),\ldots,(x_{i_k,0} \vee x_{i_k,1})$ and the clauses $\{(\bar{x}_{i_1,a_1} \vee \ldots \vee \bar{x}_{i_k,a_k} \vee x_{j,0} \vee x_{j,1}) \mid (a_1,\ldots,a_k) \in \{0,1\}^k\}$:
(1) For any ordering of the variables of these clauses, there is a poly-size ordered derivation that respects this ordering.
(2) There is a poly-size positive derivation.
Moreover, none of these derivations resolve on the variables $x_{j,0}$ or $x_{j,1}$.

Proof: Consider the clauses $\Phi = \{(x_{i_1,0} \vee x_{i_1,1}),\ldots,(x_{i_k,0} \vee x_{i_k,1})\}$ and $\Psi = \{(\bar{x}_{i_1,a_1} \vee \ldots \vee \bar{x}_{i_k,a_k}) \mid (a_1,\ldots,a_k) \in \{0,1\}^k\}$. The conjunction of all these clauses is contradictory and the number of variables is $O(\log n)$, hence lemma 12 guarantees poly-size ordered refutations for any ordering and poly-size positive refutations.

If we add the disjunction $x_{j,0} \vee x_{j,1}$ to every clause in π that is either a member of Ψ or a descendent of Ψ, then we get a derivation of $(x_{j,0} \vee x_{j,1})$ that does not resolve on $x_{j,0}$ or $x_{j,1}$. Since we do not resolve on these added variables, they do not affect the ordered quality of a refutation. Since the added variables occur positively, they do not affect the positive quality of the proof. □

Lemma 14: There are poly-size ordered (respectively, positive) resolution refutations of $IMP^l(G)$ for any dag G with in-degree bounded by $O(\log n)$.

Proof: Let t be the single sink of G (the proof is analogous if there are multiple sinks). Fix any ordering of the variables of $IMP^l(G)$ that respects a topological ordering of G. To construct the refutation, start with the graph G. Label each source j of G with the axiom $x_{j,0} \vee x_{j,1}$. For each non-source node j of G with parents $i_1, \ldots, i_k$, $k = O(\log n)$, replace j with the ordered (respectively, positive) derivation of $(x_{j,0} \vee x_{j,1})$ guaranteed by lemma 13 that respects the ordering. The result is an ordered (respectively, positive) derivation of the clause $(x_{t,0} \vee x_{t,1})$ that has not resolved on $x_{t,0}$ or $x_{t,1}$. Resolve this clause with the axioms $\bar{x}_{t,0}, \bar{x}_{t,1}$ in such a way that respects the fixed ordering (both of these resolution steps are positive). □

Corollary 15: If β is the all-ones assignment to the variables of $IMP^l(G)$, then there are poly-size negative resolution proofs of $\beta(IMP^l(G))$. This also constitutes a poly-size semantic refutation of $IMP^l(G)$.

A similar argument shows the following upper bound:

Lemma 16: There are poly-size ordered resolution refutations of $XOR(IMP(G))$ for any dag G with in-degree bounded by $O(\log n)$.

Lemma 17: Let $F_0, \ldots, F_{\ell-1}$ be CNFs over the set of variables $\{x_1, \ldots, x_n\}$ and let $\ell = 2^c$. Let $Y = \{y_0, \ldots, y_{c-1}\}$ be a disjoint set of variables. Assume that for some ordering of the variables there are poly-size ordered refutations of $XOR(F_i)$ for each i. Then there is an ordered refutation of $F = XOR(join(F_0, \ldots, F_{\ell-1}, Y))$, whose size is polynomial in n and ℓ.

Proof: Fix $0 \leq i < \ell$. Now fix one clause C_i from the set $XOR(y_0^{b(i)(0)} \vee \ldots \vee y_{c-1}^{b(i)(c-1)})$. Consider all the clauses from F that contain C_i as a subclause: ignoring the subclause C_i in each of these, we are left with exactly $XOR(F_i)$. Use the ordered refutation of $XOR(F_i)$ to derive C_i for each choice of i and C_i. At this point, we have not resolved on any of the y variables. Now we are left with the set of clauses:

$$XOR(\bigcup_{i=0}^{\ell-1} \{(y_0^{b(i)(0)} \vee \ldots \vee y_{c-1}^{b(i)(c-1)})\}).$$

This is a contradictory set of clauses on $c = \log \ell$ variables, which must have an ordered refutation of size polynomial in ℓ by lemma 12. □

Lemma 18: [15] There is a tree-like resolution refutation of $SP^l_{n,m}$ of size $nm^{O(\log n)}$.

Lemma 19: [5] GT_n has polysize negative resolution refutations. Hence, GT^l_n has poly-size negative refutations.

6. Lower Bounds

6.1. Tree-like Resolution

[4, 5, 3] showed separations of tree-like resolution from general resolution. The following lower bound is from the last:

Lemma 20: Let G be a graph on n nodes of pebbling measure $p(n)$. Then any tree-like refutation of $IMP^l(G)$ has size $2^{\Omega(p(n))}$.

Lemma 20 with lemma 14 shows that tree-like resolution does not p-simulate ordered or regular resolution. Using corollary 15, we get that tree-like resolution does not p-simulate negative or semantic resolution. By lemma 11, it does not p-simulate linear resolution either.

6.2. Regular Resolution

[13] was the first to show that regular resolution does not p-simulate general resolution. [1] showed an exponential separation between the two:

Lemma 21: Any regular resolution refutation of GT^l_n has size $2^{\Omega(n)}$.

Lemmas 21 and 19 show that regular resolution does not p-simulate negative or semantic resolution. Again, with lemma 11, we get that regular resolution does not p-simulate linear resolution.

6.3. Ordered Resolution

[11] first separated ordered resolution from general. [4] shows the following lower bound:

Lemma 22: Any ordered resolution refutation of $SP^l_{n,m}$, where $m \geq 9n/8$ has size $2^{\Omega(n \log n)}$.

With lemma 18 we get that ordered resolution does not p-simulate tree-like resolution. Hence, by lemma 9 it does not p-simulate regular resolution either. It follows from the regular resolution separations that ordered resolution does not p-simulate negative, semantic or linear resolution.

6.4. Negative Resolution

[12] gave a superpolynomial separation between negative resolution and general resolution. The following is an exponential separation between negative resolution and ordered resolution.

DEFINITION 6.1: The *negation-width* of a clause C is the number of negative literals occurring in C. The negation-width of a resolution refutation P is the maximum negation-width of all clauses in P.

We will begin with an alleged small negative resolution refutation of $IMP'(G)$, where G is a graph with high pebbling number. The following lemma shows that we can always find a restriction ρ such that after applying ρ to P, what remains is a negative refutation of $IMP(G)$, but now with small negation-width. Then we will use particular properties of negative resolution to argue that any negative refutation of $IMP(G)$ requires large negation-width, thus reaching a contradiction.

Lemma 23: For any dag on n nodes G, if there is a negative resolution refutation of $IMP'(G)$ of size at most S, then there is a negative resolution refutation of $IMP(G)$ of negation-width at most w, where $w > \log S$.

Proof: Let P be a negative resolution refutation of $IMP'(G)$ of size at most S. Call a clause of P negation-wide if its negation-width is at least w. Let $C_1, \ldots C_m$ be the set of negation-wide clauses in P, and for each C_j, let s_j be a set of w negative literals occurring in C_j. Clearly m (the number of negation-wide clauses in P) is at most S. We will define a restriction ρ probabilistically as follows. For every $i \in \{1, \ldots, n\}$, choose $x_{i,0}$ with probability $1/2$. Choose $x_{i,1}$ if and only if $x_{i,0}$ is not chosen. The assignment associated with ρ will set $x_{i,0} = 0$ if and only if $x_{i,0}$ is chosen, and otherwise, sets $x_{i,1} = 0$. We want to upper bound the probability that ρ is bad, where a restriction ρ is bad if not all negation-wide clauses in P are set to 1 by ρ. A restriction ρ is good for a particular negation-wide clause C_j if some element in s_j was chosen by ρ. The probability that this does not happen is at most $(1/2)^w$. Therefore the overall probability that ρ is bad is at most $S(1/2)^w$. Since $\log S < w$, this overall probability is less than 1, and therefore there must exist at least one good ρ. Fix a good such ρ and apply the restriction ρ to the entire proof P. What remains will be a negative resolution refutation of $IMP(G)$, of negation width at most w. $\square$

Lemma 24: Any negative resolution refutation of $IMP(G)$ has negation-width at least $\Omega(q)$, where q is the pebbling number of G.

For a proof of a more general lemma, see lemma 28 below. The intuition is that, since the refutation must start with the target axiom, i.e. the only negative axiom, the refutation constitutes a pebbling strategy in reverse. The next theorem follows directly from lemmas 24, 23 and the existence of Cook's pyramid graphs.

Theorem 25: For any graph G with pebbling measure q, any negative resolution refutation of $IMP'(G)$ requires size $2^{\Omega(q)}$. In particular, there exists an infinite sequence of graphs $\{G_n\}$ such that any negative resolution refutation of $IMP'(G_n)$ requires size $2^{\Omega(\sqrt{n})}$.

Theorem 25, with lemmas 14 and 15, show that negative resolution does not p-simulate ordered, regular or semantic resolution. Using lemma 11 again, it follows that negative resolution does not p-simulate linear resolution.

6.5. Semantic Resolution

We now generalize the lower-bound argument for negative resolution so that it works for all of semantic resolution.

Consider an instance of $\beta(IMP(G))$ on some dag G and assignment β. Let π be a α-refutation of $\beta(IMP(G))$ for some assignment α. Let $zeros(\alpha,\beta)$ be the set of nodes v of G such that $\alpha(v) \neq \beta(v)$, and $ones(\alpha,\beta)$ be the remaining nodes. Let G' be the induced subgraph on $zeros(\alpha,\beta)$. For a clause C in π, let $zeros(C,\beta)$ be the variables v in C that appear as $v^{\beta(v)}$ and let $ones(C,\beta)$ be the remainder of the variables. We'll call these literals β-negative and β-positive, respectively.

Lemma 26: Let $C \in \pi$ be a clause with one β-positive literal u such that $u \in zeros(\alpha,\beta)$. Then the set of variables that appear β-negatively in C must contain all the parents of u that are in G'.

Proof: We proceed by induction on the depth of C in the π. The only axiom in which u appears β-positively contains all of u's parents (if there are any) β-negatively. Let C be the resolvent of D and E and assume that it contains u β-positively. Either D or E must contain u β-positively. Assume, wlog, that it is D. Note that α satisfies D because u appears β-positively and $u \in zeros(\alpha,\beta)$. By induction, D must contain all of the G'-parents of u β-negatively. The only way that C could not contain all of these parents is if D and E resolve on one of them, call it v. But this means that v appears β-positively in E and we know that $v \in zeros(\alpha,\beta)$, so α must satisfy E too. Hence D and E cannot be resolved. $\square$

Lemma 27: For any clause C in π, let $S_C = zeros(C,\beta) \cap zeros(\alpha,\beta)$. Assume the pebbling measure of S_C in G' is p. The portion of π from the last occurrence of C to the end of the proof has β-negation-width at least p.

Proof: We proceed by induction on the number of clauses that follow C in π. To begin with, let $C = \Lambda$ be the last clause in π. Since $S_C = \emptyset$, there is nothing to prove. Now let C be an arbitrary clause in π. Assume C is resolved with D on the variable u and that E is their resolvent. If $S_C \subset S_D$, then we are done by the induction hypothesis. Otherwise it must be the case that $u \in zeros(\alpha,\beta)$ and that u appears β-negatively in C and β-positively in D. By lemma 26, D must contain all of u's G'-parents β-negatively. Hence, as a pebbling configuration, S_C follows easily from S_D. □

Lemma 28: Let G be a pyramid-like dag on n nodes and let α,β be assignments to the variables of $IMP(G)$. Let G' be the induced subgraph on $zeros(\alpha,\beta)$. If G' contains a node v of pebbling measure p such that there is a directed path in G from v to the target node t, then any α-refutation π of $\beta(IMP(G))$ requires β-negation-width at least p.

Proof: There is at least one axiom in $\beta(IMP(G))$ that contains v β-negatively. This axiom must appear in π, since without it $\beta(IMP(G))$ is satisfiable (since there is a path in G from v to t). Hence, by lemma 27, π has β-negation-width at least p. □

Lemma 29: For infinitely many n and any $n' \geq n$, there exist assignments $\beta_1,\ldots,\beta_{n'} \in \{0,1\}^n$ and pyramid-like graphs $G_1,\ldots,G_{n'}$ of in-degree $O(\log n)$ on n nodes such that the following holds: for any assignment $\alpha \in \{0,1\}^n$, there exists an i such that any α-refutation of $\beta_i(IMP(G_i))$ requires β_i-negation-width $\Omega(\sqrt{n})$.

Proof: Fix m sufficiently large and let $n = \sum_{i=1}^m i$. Let $n' \geq n$. Choose $\beta_1,\ldots,\beta_{n'}$ randomly and independently from the distribution on $\{0,1\}^n$ such that each bit is 0 with probability $1/2$, and 1 otherwise. Choose $G_1,\ldots,G_{n'}$ randomly and independently from $Pyr(n,d)$, where $d > 5\log_{8/5} m$. Fix $\alpha \in \{0,1\}^n$. Let G'_i be the subgraph of G_i induced by $zeros(\alpha,\beta_i)$. We first show that with high probability, one of the G'_is satisfies the conditions of lemma 6.

Fix i such that $1 \leq i \leq n'$. Consider layers 0 through $m/8$ of G'_i. Layer j is expected to contain $(m-j)/2$ nodes in G'_i. The probability that it contains fewer than half this many is less than $exp(-(m-j)/16)$, by Chernoff's bound. The probability that any of these layers contains less than half its expected number of nodes is at most $(m/8)exp(-7m/128)$. Call this event $A(i,\alpha)$.

Now consider the probability that any subset of layer j of size $s \leq (m-j)/8$ is not expanding into layer $j-1$ for $j > 0$. First we fix a subset S_1 of size s from layer j and then a subset S_2 of size s from layer $j-1$. Then, for each node in S_1 and each G_i-parent v of that node, the probability that v does not appear in G'_i is $1/2$. The probability that v is contained in S_2 is at most $s/(m-j+1)$. The probability that either of these bad events occurs is at most $1/2 + s/(m-j+1) < 5/8$. Hence, the probability that layer j is not expanding into layer $j-1$ is bounded by

$$\sum_{s=1}^{(m-j)/8} \binom{(m-j+1)}{s}^2 \frac{5}{8}^{ds}.$$

For $d > 5\log_{8/5} m$, this is bounded by $1/m^2$. The probability that any layer $0 \leq j \leq m/8$ is not expanding, then, is bounded by $\frac{m}{8}\frac{1}{m^2} < \frac{1}{m}$. Call this event $B(i,\alpha)$. The event $A(i,\alpha) \bar{\cup} B(i,\alpha)$ implies that there is a node in G'_i with pebbling measure $m/8$ by lemma 6.

Finally, let $C(i,\alpha)$ be the event that there is no node v in layer $m/8$ of G'_i such that there is a path in G_i from v to the target t. Let $E(i,\ell)$ be the event that some fixed node in layer ℓ of G_i does not have a path to t, and let $F(i,\alpha)$ be the event that no node in layer $m/8$ of G_i survives in G'_i. Clearly $\Pr(C(i,\alpha)) \leq \Pr(E(i,m/32)) + F(i,\alpha)$. We now compute $\Pr(E(i,\ell))$: a fixed node u in layer ℓ of G_i will not have a path to t if u has no children in layer $\ell+1$ or if none of these children have a path to t. The probability of the former is

$$\left(\frac{m-\ell-1}{m-\ell}\right)^{d(m-\ell-1)}$$

and the probability of the latter is bounded by $E(i,\ell+1)$. Hence, we have $\Pr(E(i,m-1)) = 0$, and

$$\begin{aligned}\Pr(E(i,\ell)) &\leq \Pr(E(i,\ell+1)) + \left(\tfrac{m-\ell-1}{m-\ell}\right)^{d(m-\ell-1)}\\ &\leq \Pr(E(i,\ell+1)) + \tfrac{1}{e^d}\\ &< m/e^d.\end{aligned}$$

Taking $d > 5\log_{8/5} m$ as above, this probability is less than $1/m^4$. So, $\Pr(C(i,\alpha)) \leq 1/m^4 + 1/2^{\frac{m}{8}} \leq 1/m^3$.

For any i, let $D(i,\alpha)$ be the event $A(i,\alpha) \cup B(i,\alpha) \cup C(i,\alpha)$. By lemma 28, $D(\bar{i},\alpha)$ implies that any α-refutation of $IMP(G_i)$ requires β_i-negation-width $m/8 = \Omega(\sqrt{n})$. By a union bound,

$$\Pr(D(i,\alpha)) \leq \frac{m}{8}exp(-7m/128) + \frac{1}{m} + \frac{1}{m^3} < \frac{3}{m}.$$

Let $D(\alpha) = \bigcap_{i=1}^{n'} D(i,\alpha)$. Then $\Pr(D(\alpha)) \leq \left(\frac{3}{m}\right)^{n'}$. The probability that $D(\alpha)$ holds for any $\alpha \in \{0,1\}^n$ is at most $2^n(3/m)^{n'} < 1$. Hence, there exist $G_1,\ldots,G_{n'}$, $\beta_1,\ldots,\beta_{n'}$ such that, for any α there is some i such that any α-refutation of $\beta_i(IMP(G_i))$ requires β_i-negation-width $\Omega(\sqrt{n})$. □

Theorem 30: Fix $n = \sum_{i=1}^m i$ for m sufficiently large. Fix $n' > n$ to be a power of 2. There exist $\beta_1,\ldots,\beta_{n'} \in \{0,1\}^n$ and dags on n nodes $G_1,\ldots,G_{n'}$ such that for any semantic refutation π of $XOR(join(\beta_1(IMP(G_1)),\ldots,\beta_{n'}(IMP(G_{n'})),Y))$, $size(\pi) = exp(\Omega(\sqrt{n}))$. Here $Y = \{y_1,\ldots,y_{\log n'}\}$ is a set of variables disjoint from those of denoting the vertices of the dags.

Proof: The theorem follows immediately from lemmas 29 and 4. $\square$

Theorem 30 and lemmas 16 and 17 show that semantic resolution does not p-simulate ordered or regular resolution. Lemma 11 gives a separation of linear from semantic.

7. Other refinements of resolution

Of course there are many other resolution refinements that we have not mentioned. One of the most popular is *set-of-support resolution*. Let F be an unsatisfiable CNF viewed as a set of clauses. $S \subset F$ is called a set-of-support if $F \setminus S$ is satisfiable. An S-refutation of F is a resolution refutation in which no two resolvents are both from the set $F \setminus S$. S-resolution is known to be complete for any set-of-support S. A refutation is called *set-of-support* if it is an S-refutation for some set of support S. Clearly set-of-support resolution p-simulates general resolution since we can simply choose $S = F$. For any set of support S, however, we can apply an analogue to lemma 10, since any linear refutation makes only one resolution between two initial clauses:

Lemma 31: Let F be an unsatisfiable CNF such that there is a resolution refutation of size S and width w. Then, for any set-of-support S, there is an S-refutation of $ADDTAUT(F)$ of size $O(S \cdot w)$.

8. Acknowledgements

We are grateful to David Mitchell, Nicola Galesi and Alasdair Urquhart for some very helpful discussions.

References

[1] M. Alekhnovich, J. Johannsen, T. Pitassi, and A. Urquhart. An exponential separation between regular and general resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 448–456, New York, May 19–21 2002. ACM Press.

[2] E. Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 457–464, New York, May 19–21 2002. ACM Press.

[3] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 517–526, Atlanta, GA, May 1999.

[4] M. Bonet, J. L. Esteban, N. Galesi, and J. Johannsen. Exponential separations between restricted resolution and cutting planes proof systems. In *Proceedings from 38th FOCS*, pages 638–647, 1998.

[5] M. Bonet and N. Galesi. A study of proof search algorithms for resolution and polynomial calculus. 1999.

[6] J. Buresh-Oppenheim, M. Clegg, R. Impagliazzo, and T. Pitassi. Homogenization and the polynomial calculus. In *27th International Colloquium on Automata, Languages and Programming*, pages 926–937, 2000.

[7] S. A. Cook. An observation on time-storage trade off. In *Conference Record of Fifth Annual ACM Symposium on Theory of Computing*, pages 29–33, Austin, TX, Apr.-May 1973.

[8] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.

[9] M. Davis and H. Putnam. A computing procedure for quantification theory. *Communications of the ACM*, 7:201–215, 1960.

[10] U. Dunker. *Zur Effizienz der Beweissuche in der Logikverarbeitung*. PhD thesis, Universität Paderborn, 1997.

[11] A. Goerdt. Davis-Putnam resolution versus unrestricted resolution. *Annals of Mathematics and Artificial Intelligence*, 6:169–184, 1992.

[12] A. Goerdt. Unrestricted resolution versus n-resolution. *Theoretical Computer Science*, 93:159–167, 1992.

[13] A. Goerdt. Regular resolution versus unrestricted resolution. *SIAM Journal on Computing*, 22(4):661–683, 1993.

[14] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–305, 1985.

[15] J. Johannsen. Exponential incomparability of tree-like and ordered resolution. 2001.

[16] R. Kowalski and D. Kuehner. Linear resolution with selection function. *Artificial Intelligence*, 2:227–260.

[17] D. Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science*, pages 254–266.

[18] B. Krishnamurthy. Short proofs for tricky formulas. *Acta Informatica*, 22:253–275, 1985.

[19] W. J. Paul, R. E. Tarjan, and J. R. Celoni. Space bounds for a game on graphs. *Mathematical Systems Theory*, 10(3):239–251, 1977. Correction, *ibid.* 11(1):85, 1977.

[20] R. Raz and P. McKenzie. Separation of the monotone nc hierarchy. In *Proceedings of 37th IEEE Foundations of Computer Science*, 1997.

[21] J. A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[22] G. S. Tseitin. On the complexity of derivation in the propositional calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part II.* 1968.

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

[24] A. Vellino. *The Complexity of Automated Reasoning*. PhD thesis, University of Toronto, 1989.