Hard Examples for Resolution

ALASDAIR URQUHART

University of Toronto, Toronto, Ontario, Canada

Abstract. Exponential lower bounds are proved for the length-of-resolution refutations of sets of disjunctions constructed from expander graphs, using the method of Tseitin. Since these sets of clauses encode biconditionals, they have short (polynomial-length) refutations in a standard axiomatic formulation of propositional calculus.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*complexity of proof procedures*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*mechanical theorem proving*

General Terms: Algorithms, Performance, Theory, Verification

Additional Key Words and Phrases: Complexity of proofs, hard examples, resolution proofs

1. *Introduction*

We deal in this paper with the length of minimal refutations of contradictory sets of clauses by the resolution method. In a recent remarkable paper, Haken [9] has shown that there is an infinite sequence P_n of sets of clauses, each set P_n having length $O(n^3)$, such that the shortest resolution of P_n requires at least c^n distinct steps, for a fixed $c > 1$. The example P_n is the set of clauses that encodes the statement that $n + 1$ objects can fit exactly into n holes. The fact that P_n is contradictory is thus an expression of the pigeonhole principle, so that they are known as "pigeonhole clauses" [4].

In the present paper we adapt Haken's method of proof to prove an exponential lower bound for minimal refutations of another family of examples, the graph-based sets of clauses originally defined by Tseitin in the earliest paper on the complexity of resolution [17]. Tseitin proved a superpolynomial lower bound for the length of resolution refutations of sets of clauses based on square grids, under the assumption that the refutations satisfied a restriction known as "regularity." Subsequent work by Kirkpatrick [11], Galil [7, 8], and Ben-Ari [1] simplified Tseitin's proof and improved the lower bound for regular resolution. However, great difficulties were experienced in extending Tseitin's argument to unrestricted resolution. We show in this paper that there is an infinite sequence S_n of graph-based sets of clauses of length $O(n)$ that require resolution refutations of length at least c^n for a fixed $c > 1$. The lower bound is proved by the ingenious counting technique first used by Haken. The use of the graph-based clauses enables us to

Author's address: Department of Philosophy, University of Toronto, 215 Huron Street, Toronto, Ont., Canada.

improve on Haken's results in several respects. First, the lower bound is the best possible, except for the choice of the constant c. Second, and more important, these clauses can be shown to have short (polynomial-length) proofs in a standard axiomatic system for the propositional calculus, something which is not known to be true for the pigeonhole clauses [5]. Last, the proof can be modified to show an exponential lower bound for resolution refutations for sets of clauses in which each clause contains three literals; the pigeonhole clauses P_n contain clauses of length n.

2. *Basic Definitions*

We assume an unbounded set $\{a, b, c, \ldots, h, a_1, \ldots\}$ of *variables*. A *literal* is either a variable p or its negation $\sim p$. The *complement* of p is $\sim p$, the complement of $\sim p$ is p; we denote the complement of a literal q by $\bar{q}$. A *clause* is a set of literals, interpreted as the disjunction of the literals contained in it. A literal q is *mentioned* in a clause C if either q or its complement is in C. If S is a set of variables, then a *truth-value assignment* (*tva*) is an assignment of the values 0 or 1 to each variable in S (0 = false, 1 = true). The value (relative to a tva φ) of the literal p ($\sim p$) is $\varphi(p)$ ($1 - \varphi(p)$). The value of a clause C is the maximum of the value of the literals in it. We write Λ for the empty clause; it takes the value 0 for any assignment and thus represents a standard contradiction. A set of clauses is *satisfiable* if there is a tva making all the clauses in the set true; otherwise the set of clauses is *unsatisfiable* or *contradictory*.

We are interested in the complexity of proof systems that show sets of clauses to be contradictory. A proof that shows a set S of clauses to be contradictory we call a *refutation* of S. The basic proof system we consider is the resolution method, which has its origin in the work of Blake [2], but which became widely popular as a theorem-proving procedure in the predicate calculus when combined with a unification algorithm by Robinson [15] to give a single inference rule for problems in predicate logic (after eliminating quantifiers by using Skolem functions).

Two clauses C and D are said to *clash* if there is exactly one literal q in C that appears complemented in D. If C and D clash, then their *resolvent* is the clause $C - \{q\} \cup D - \{\bar{q}\}$. We say that the resolvent is obtained by applying the *resolution rule*, *resolving on*, or *annihilating* the literal q. A *resolution proof* of a clause C from a set of clauses S is a sequence of clauses, each of which is either in S or derived from earlier clauses in the sequence by the resolution rule. A proof of the empty clause Λ from S is a *resolution refutation* of S. We refer to the clauses in S as the *input clauses*. A resolution proof can be viewed as a tree, in which the last clause in the proof is the root, the two offspring of a resolvent are the clauses from which it is inferred by the resolution rule, and the leaves are the clauses in S. We take the length of a resolution proof (i.e., the number of steps in the sequence) as the measure of its complexity; this measure is adequate because no clause in a minimal resolution refutation can be longer than the size of the set of variables in the input clauses. If the proof is viewed as a tree, then the complexity is the number of distinct clauses occurring in the tree; the size of the tree itself may be exponentially bigger, as Tseitin showed [17].

We also wish to consider the complexity of derivations in axiomatic versions of the propositional calculus. This type of proof system is dubbed a "Frege system" by Cook and Reckhow [5] and contains a finite number of axiom schemes and a finite set of rules of inference that are *sound* in the sense that the premises of any instance of the rule logically entail the conclusion of the rule (note that the rule of uniform substitution for variables is *not* sound in this sense). It was shown by

Reckhow in his dissertation [14] that any two such Frege systems based on functionally complete sets of connectives are equally efficient, up to a polynomial. That is to say, there exists a way of translating from one system to the other so that the length of the proof that results from the translation is bounded by a fixed polynomial in the length of the original proof. If the two Frege systems contain the same connectives, this is straightforward. On the other hand, if, for example, one system contains only conjunction and negation, and the other conjunction, negation, and the biconditional, then direct translation is impossible because there is no efficient direct method of translating biconditionals into conjunction and negation alone. However, in his disseration Reckhow shows how to overcome this difficulty by giving a method for indirect translations in such a case. Thus any two Frege systems can *p-simulate* each other (see [5] for a precise definition of this concept), so that all such systems may be considered equally powerful, at least from the point of view of polynomial-versus-nonpolynomial complexity measures. It can also be shown that any natural deduction system and any Gentzen system with the cut rule are equivalent (in the p-simulation ordering) to a fixed Frege system.

The systems equivalent to Frege systems (with respect to the p-simulation ordering) thus include many natural and seemingly quite powerful proof systems. In view of the simulation results reported above, it is sufficient to consider a fixed axiomatic system for the propositional calculus. We choose a formulation that contains the biconditional, conjunction, disjunction, implication, and negation as primitive operators; for definiteness we choose the system given by Kleene [12, pp. 33–34], which has 13 axiom schemes and *modus ponens* as rule of inference.

3. *Critical Truth-Value Assignments and a General Lower Bound*

In this section we define the concept of critical truth-value assignment and give an abstract form of the lower bound argument, which is general enough to include the arguments of both Haken and the present paper.

Let S be a contradictory set of clauses, and R the set of variables mentioned in S. If C is a fixed clause in S, then a tva defined on R is said to be *C-critical* if it makes all the clauses in S true except for C. A tva is *critical* if it is C-critical for some C in S. S is *minimally inconsistent* if C-critical tvas exist for any C in S. The notion of critical tva is not new, in the sense that it is implicit in the concept of minimally inconsistent set of clauses familiar from the literature of automatic theorem proving. However, it was Haken who first showed that the idea gives us considerable insight into the structure of resolution proofs.

Let P be a partial tva for the set of clauses S, that is, a tva defined on a subset of the set of variables mentioned in S. We say that P is a *partial solution for S* if P does not falsify any clause in S, and furthermore P can be extended to a critical tva for S. It is typical of known hard examples for resolution that they have exponentially many partial solutions (relative to the size of the input clauses); roughly speaking, a resolution refutation for such examples has to be long to take account of the huge number of possible partial solutions. We now attempt to make this last remark precise.

Suppose, as above, that we have a fixed contradictory set S of input clauses, which we may assume to be minimally inconsistent, and let REF(S) be a minimal-length resolution refutation of S. Let us suppose that, in addition, we have chosen a fixed family PS of partial solutions for S. Now for each partial solution P in PS, let Θ_P be a property of clauses in REF(S), defined in terms of P. We say that Θ_P is

a *root property* if the last step in REF(S), that is, the empty clause, has Θ_P. A clause in REF(S) is *P-complex* if it is the first clause in REF(S) that has the property Θ_P. A clause in REF(S) is *complex* if it is P-complex for some P in PS.

In order to prove a general lower bound for the length of REF(S), we regard PS as a probability space, with **P** a random variable on the space having the uniform distribution (i.e., we assign all partial solutions in PS the same probability). To prove a lower bound for proof length, we compute an upper bound for the probability Pr(C is **P**-complex), where C is a complex clause.

LEMMA 3.1. *Let PS be a family of partial solutions for S, and Θ_P a root property for each P in PS. Then REF(S) contains at least r^{-1} complex clauses, where r is an upper bound on the probability Pr(C is* **P***-complex) and C is any complex clause in REF(S).*

PROOF. Since Θ_P is a root property, REF(S) must contain a P-complex clause for any partial solution P in in PS. Thus the sum $\sum$ Pr(C is **P**-complex), summing over all complex clauses in the refutation, is ≥ 1. It follows that there must be at least r^{-1} complex clauses in the proof. □

In the remarks we made above about the characteristics of hard examples, we in fact omitted a further key property of the cases that are known to present difficulties for the resolution method. It is not sufficient that there are a large number of partial solutions, as can be shown by simple examples. It is necessary in addition that the sets of clauses be "strongly connected" in the sense that a large number of interconnections (in the form of clashes) link the clauses in the examples. In the pigeonhole and Tseitin examples, this high degree of connectedness manifests itself in part in a property that we now define.

Let S be a set of clauses. We say that S has the *toggling property* if the following condition holds: If C is a clause in S, and φ a C-critical tva for S, then any tva obtained from φ by altering the truth-value of a literal mentioned in C is also a critical tva for S. If ψ is the tva obtained from φ as above, by reversing the value of the literal q, we can then say that ψ is obtained from φ by *toggling* q. If we visualize the tvas for a set of n variables as forming the vertices of an n-dimensional hypercube, and the clauses C in S as labeling the C-critical tvas, then the toggling property amounts to the idea that if we start from any vertex labeled with C and move to an adjacent vertex along an edge labeled by a literal mentioned in C, we arrive at another vertex labeled with a clause from S. Clearly, a set of clauses having this property is "tightly connected" from a logical point of view.

4. *Hard Examples Based on Graphs*

By a *graph* we mean a finite set of points or *vertices*, together with a finite set of edges joining pairs of these vertices. We do not allow loops or multiple edges. Let G be a graph; we wish to consider *labelings* of G, which consist in associating a distinct variable with each edge, then labeling each edge with its associated variable or its negation, and then assigning a *charge* Charge(x) of 0 or 1 to each vertex x in G. Let G' be a labeled graph corresponding to G. The *total charge* Charge(G') of G' is the sum modulo 2 of the charges attached to the vertices of G. In what follows, it will prove convenient to identify literals with the edges they label, so that, for instance, we shall speak of the literal that links two vertices x and y. The set of clauses $S(G')$ associated with the labeled graph G' is defined as follows: Let x be a vertex of G, and Lit(x) the set of literals attached to x. Then Clauses(x) is defined to be the set of all clauses C that mention exactly the literals in Lit(x), and

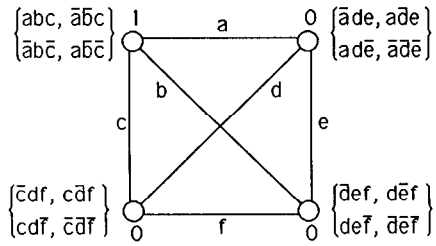

FIGURE 1

the parity of the number of literals in Lit(x) that occur complemented in C is opposite to Charge(x). The set of clauses $S(G')$ is defined to be the union of all the sets Clauses(x) for x in G'. If x has n literals attached to it, then Clauses(x) contains 2^{n-1} clauses. Figure 1 shows Clauses(G') for an odd labeling of the complete graph on four vertices. It is easy to see that Clauses(x) is the conjunctive normal form of the modulo 2 equation $q_1 \oplus \cdots \oplus q_n = \text{Charge}(x)$, where $q_1, \ldots, q_n$ are the literals attached to x; we denote this equation by $E(x)$. This observation immediately gives us a method of deciding which labeled graphs give us contradictory sets of clauses.

LEMMA 4.1. *If G is a connected graph, then the set of clauses $S(G')$ is contradictory if and only if* Charge(G') = 1.

PROOF. Assume Charge(G') = 1. Then if we sum the left-hand side (LHS) of all the mod 2 equations associated with the vertices of G, the result is 0, because each literal is attached to exactly two vertices and hence occurs twice in the sum. On the other hand, the right-hand side (RHS) sums to 1, by assumption, so the set of equations and hence the set of disjunctions are contradictory.

Conversely, suppose Charge(G') = 0. Let $E(G')$ be the set of all the equations $E(x)$ for x in G'. If q is a literal labeling an edge in G', then we can rewrite the set of equations $E(G')$ with respect to the literal q by replacing q by its complement in the two equations in which it occurs and by complementing the RHS of these two equations. The resulting set of equations is clearly equivalent to $E(G')$ and is identical with $E(G'')$, where G'' is the labeled graph obtained from G' by interchanging q and its complement and by complementing the charge of the vertices x and y which it links. We refer to this operation as *transferring a charge from x to y*. Now if x and y are two vertices in G with Charge(x) = Charge(y) = 1, then there is a chain of vertices $x = v_1, \ldots, v_n = y$ that form a path from x to y. Thus if we successively transfer a charge from v_1 to v_2 to $\cdots$ v_n, then the result is an equivalent set of equations associated with a labeling that has two fewer vertices with an odd charge. By repeating this process, we finally obtain a set of equations in which the RHS is uniformly 0. A satisfying assignment is obtained by making all literals take the value 0. □

Henceforth, we assume G' is a connected graph with an odd labeling (that is, Charge(G') = 1), and that each edge is labeled with a distinct variable. It is very easy to describe the critical tvas for $S(G')$. If φ is a tva to the edges of G, then we write Charge(φ, x) for the mod 2 sum of the values under φ of the edges attached to x, where x is a vertex of G. We say that φ is *x-critical* if it is C-critical for some C in Clauses(x).

LEMMA 4.2. *A tva φ for $S(G')$ is x-critical iff* Charge(φ, y) = Charge(y) *for all vertices in G, except for x.*

PROOF. If Charge(φ, y) = Charge(y), then the mod 2 equation corresponding to y is satisfied; hence all clauses in Clauses(y) are satisfied. Since $S(G')$ is contradictory, Charge(x) cannot equal Charge(φ, x). □

If G is a connected graph, we define the *cyclomatic number* $C(G)$ of G as the number of edges of G minus the number of vertices of G plus 1. This is the rank of the cycle space of G [18, pp. 197–200] and is equal to the number of edges not in T, where T is any spanning tree for G. Let us say that a partial tva φ to the edges of G' is *nonseparating* if the graph that results from G' by deleting all the edges assigned a value by φ is connected.

LEMMA 4.3. *If φ is a nonseparating partial tva to the edges of G', and x is any vertex of G, then φ may be extended to an x-critical tva for $S(G')$.*

PROOF. Fix a spanning tree for G which does not contain any edge assigned a value by φ; this is possible because φ is nonseparating. Assign values arbitrarily to any edge not in the spanning tree that has not yet been assigned a value. We must now extend this assignment to an x-critical tva ψ. Proceeding from the leaves of the tree inward toward x, assign values to the edges adjacent to vertices y other than x so that Charge(ψ, y) = Charge(y). The resulting tva is uniquely determined by the values given to edges not in the spanning tree and must be x-critical because $S(G')$ is contradictory. □

COROLLARY 4.4. *There are exactly $2^{C(G)}$ x-critical tvas for $S(G')$, where x is any vertex in G.*

PROOF. If T is any spanning tree for G, then there are exactly $2^{C(G)}$ nonseparating partial tvas that assign values to all the edges not in the spanning tree. The x-critical tvas that extend these partial tvas form the set of all x-critical tvas because of the uniqueness of the extensions. □

LEMMA 4.5. *If x and y are adjacent vertices in G' joined by the literal q, then if φ is an x-critical tva for $S(G')$, then the tva that results from φ by toggling q is y-critical.*

PROOF. This is immediate from Lemma 4.2. □

COROLLARY 4.6. *$S(G')$ has the toggling property.*

We conclude this section by observing that the pigeonhole clauses also have the toggling property. These clauses are formulated using doubly subscripted variables P_{ij}, where $1 \leq i \leq n+1$, $1 \leq j \leq n$, which are to be interpreted as "Object number i is in hole j." There are $n+1$ clauses of length n that say that object i must be in one of the n holes, and $(n^3 + n^2)/2$ two-literal clauses that say that no hole can contain more than one object. There are two types of critical tvas for this set of clauses. If C is a clause of the first type, which says that object i must be in one of the holes, then a tva that is C-critical corresponds to placing all the objects except i into the set of holes. If D is a clause of the second type, then a D-critical tva corresponds to filling all the holes with objects so that one hole contains two objects. Toggling the appropriate variables amounts to putting the missing object in one of the holes (in the first case) or removing one of the two objects from the hole with two (in the second). Thus the pigeonhole clauses have the toggling property. This fact is already implicit in Haken's proof [9, Lemma 2.3], although Haken's use of the term "critical" differs from ours in that he defines the critical tvas to consist only of the first type of tva mentioned above.

5. *Sets of Clauses Based on Expander Graphs*

To find a hard set of examples for resolution, we must find a sequence of graphs G_m that are highly connected, but each vertex is of limited degree, so that there are not too many clauses associated with G_m. We choose for this purpose the sequence of graphs originally used by Galil to prove a lower bound for regular resolution, with a small modification to simplify the proof. We start with a sequence of expander graphs H_m; the graph H_m is a bipartite graph in which each vertex has degree at most 5, such that each of its sides contains m^2 vertices (for brevity we shall write $n = m^2$). The particular family of expander graphs we use was first defined by Margulis [13]. Margulis proved the expanding property stated in the next lemma.

LEMMA 5.1. *There is a constant $d > 0$ such that, if V_1 is contained in one side of H_m, $|V_1| \leq n/2$, and V_2 consists of all the vertices in the other side of H_m that are connected to vertices of V_1 by an edge, then $|V_2| \geq (1 + d)|V_1|$.*

PROOF. See Gabber and Galil [6], who also provide a numerical lower bound for the expansion factor d. □

The graphs that we use to construct our clauses are obtained from H_m by the following modification. We add $n - 1$ edges to each side of the graph so that each side forms a connected chain. We call the new edges *side edges*, and the edges in H_m *middle edges*. The resulting graph G_m obviously still satisfies Lemma 5.1, and each vertex in it has degree at most 7. Choose an odd labeling G'_m of G_m. Our sequence S_m of hard examples is $S(G'_m)$. S_m contains at most $128n$ clauses of length at most 7, so the entire set of clauses has length $O(n)$.

In order to apply our earlier lower bound argument, we have to specify the set PS of partial solutions for S_m and also appropriate root properties of clauses in REF(S_m). Let us suppose that the sides of G_m are listed each in a fixed order, so that we can speak of corresponding vertices. Let d be the constant in Lemma 5.1, and let f be $d/16$. A partial solution is then specified by first choosing $\lfloor fn \rfloor$ vertices from one side of G_m, together with the corresponding vertices on the opposite side, and then assigning truth values arbitrarily to the middle edges attached to at least one of the chosen vertices. The set PS_m of partial solutions for S_m consists of all partial tvas obtained in this way. If P is a member of PS_m, we write $V(P)$ for the set of $2\lfloor fn \rfloor$ vertices chosen as the underlying vertices for P. Before we specify what our root properties are, we must verify that the tvas we have specified are indeed partial solutions as we defined that concept in Section 3 above.

LEMMA 5.2. *Let P be a member of the set PS_m, and x a vertex of G_m. Then P can be extended to an x-critical tva for S_m.*

PROOF. Since there is at least one middle edge not assigned a value by P, the partial tva P is nonseparating, so that the conclusion follows by Lemma 4.3. □

We now specify a root property for every P in PS_m. If C is a clause and φ a tva, then we say that C *covers* φ if φ falsifies C. If P is in PS_m and C is a clause, we define the set of vertices covered by C with respect to P, *Cover*(C, P), as the set of all vertices x such that there is an x-critical tva extending P covered by C, and x is not in $V(P)$. A clause C has the property Θ_P if $|\mathrm{Cover}(C, P)| \geq n/4$. Since the empty clause covers all tvas and $1 - 2f > 1/4$ (since $d \leq 4$), this is a root property.

LEMMA 5.3. *If C is a P-complex clause, there are at least fn vertices contained in one side of G_m such that C mentions at least one middle literal attached to each vertex.*

PROOF. If C is an input clause in Clauses(y), then C can cover only y-critical tvas, so we may assume that C is inferred from earlier clauses D and E by the resolution rule. Now if φ is a tva covered by C, then it must be covered by D or E. Thus $\text{Cover}(C, P) \subseteq \text{Cover}(D, P) \cup \text{Cover}(E, P)$. Since C is P-complex, both $|\text{Cover}(D, P)|$ and $|\text{Cover}(E, P)|$ are less than $n/4$. Thus $|\text{Cover}(C, P)| < n/2$. Now let $\text{Cover}(C, P) = W_1 \cup W_2$, where W_1 and W_2 are contained in opposite sides of G_m and $|W_1| \geq |W_2|$. Let Y_2 be the vertices connected to W_1 by a middle edge that are not in W_2. Since $|\text{Cover}(C, P)| \geq n/4$, $|W_1| \geq n/8$; so by Lemma 5.1, $|Y_2| \geq dn/8$. Now let Z_2 be the vertices in Y_2 that are not in $V(P)$. By the definition of f, there are at least fn vertices in Z_2.

We have to establish that, if y is a vertex in Z_2, then a middle literal attached to y is mentioned in C. Now, by definition, there is a literal attached to a middle edge that links y to a vertex x in W_1. By definition, there is an x-critical tva φ that extends P and is covered by C. Now if the literal attached to this edge is not mentioned in C, then the tva obtained from φ by toggling it is also covered by C. By Lemma 4.5, the new tva is y-critical. Since neither x nor y is in $V(P)$, it is also an extension of P. This contradicts the fact that y is not in W_2. □

We have now proved that for every partial solution P there is a P-complex clause that contains at least fn literals. To prove an exponential bound on proof length, we have to prove an exponentially small upper bound on the probability $\Pr(C \text{ is } \mathbf{P}\text{-complex})$, where the probability space is as defined in Section 3. To accomplish this, from probability theory we need two results that give an exponentially small upper bound for the tail of the hypergeometric distribution.

To fix notation for the next two lemmas, let $\mathbf{B}$ be a random variable with the binomial distribution with parameters s, p, and let $\mathbf{H}$ be a random variable with the hypergeometric distribution with parameters s, M, N, where $M/N = p$. That is, $\Pr(\mathbf{B} = k)$, $k \leq s$, represents the probability of obtaining k "good items" when taking a sample size s with replacement, where the probability of obtaining a good item at each draw is p. The probability $\Pr(\mathbf{H} = k)$, $k \leq s$, represents the probability of obtaining k "good items" for the same sample size when the sampling is done without replacement from a population of size N containing M good items, where $M/N = p$. We shall assume that $\mathbf{B}$ represents sampling from the same population as $\mathbf{H}$. Uhlmann has shown that over a wide range of values, the tail of the hypergeometric distribution is dominated by the tail of the corresponding binomial distribution.

LEMMA 5.4. *For any $c \geq 0$, if $c(s-1)^{-1}N(N+1)^{-1} + (N+1)^{-1} \leq p$, then $Pr(\mathbf{H} \leq c) < Pr(\mathbf{B} \leq c)$.*

PROOF. See Johnson and Kotz [10, p. 151, formula (23)]. □

THEOREM 5.5. *For $h \geq 0$, $Pr(p - \mathbf{B}/N \geq h) < exp(-2Nh^2)$.*

PROOF. This is a result of Okamoto [10, p. 69, formula (53.2)]. □

If C is a complex clause, define the *good vertices* associated with C, *Good*(C), as the vertices that are contained in a chosen side of H_m and have a middle edge attached to them, which is mentioned in C. By Lemma 5.3, we can choose a side of H_m so that $|\text{Good}(C)|$ is at least $\lceil fn \rceil$.

LEMMA 5.6. *There is a constant $a < 1$ such that, for n sufficiently large, $Pr(C$ is* **P***-complex) $\leq a^n$ for any complex clause C.*

PROOF. Define a random variable **O** that represents the overlap between Good(C) and V(**P**); that is, **O** = |Good(C) $\cap$ V(**P**)|. We can then rewrite the probability Pr(C is **P**-complex) as the sum

$$\sum_{k=0}^{\lfloor fn \rfloor} \Pr(\mathbf{O} = k) \cdot \Pr(C \text{ is } \mathbf{P}\text{-complex given } \mathbf{O} = k). \qquad (*)$$

Now the variable **O** has a hypergeometric distribution representing sampling without replacement from a population of size n containing at least $\lceil fn \rceil$ good objects, taking samples of size $\lfloor fn \rfloor$. Let **Q** be a random variable with the binomial distribution representing sampling with replacement from the same population. Substituting in the inequality of Lemma 5.4, we have $N = n$, $s = \lfloor fn \rfloor$, $M = \lceil fn \rceil$. Now we can choose g so that for n sufficiently large the inequality

$$\lfloor gn \rfloor (\lfloor fn \rfloor - 1)^{-1} n(n+1)^{-1} + (n+1)^{-1} \leq \frac{\lceil fn \rceil}{n}$$

holds, and $0 < g < f$. Then Pr(**O** $\leq \lfloor gn \rfloor$) < Pr(**Q** $\leq \lfloor gn \rfloor$), by Lemma 5.4. Letting $h = f - g$, we have Pr(**Q** $\leq \lfloor gn \rfloor$) < $\exp(-2nh^2)$, by Lemma 5.5. We have thus proved that the sum of the terms in (*) from 0 to $\lfloor gn \rfloor$ is bounded by i^{-n} for a fixed $i > 1$. It remains to compute Pr(C is **P**-complex given **O** > $\lfloor gn \rfloor$). Now if the overlap between the good vertices of C and $V(P)$ is at least gn, then, since an arbitrary assignment of truth values to the middle edges attached to $V(P)$ results in a partial solution in PS_m, it follows that C can cover at most 2^{-gn} of the partial solutions defined on $V(P)$. Thus the sum of the remaining terms in (*) is bounded by 2^{-gn}, so that there is a constant $a < 1$ such that the entire sum is bounded by a^n for sufficiently large n, as claimed. □

THEOREM 5.7. *There is a constant $c > 1$ such that for sufficiently large m, any resolution refutation of S_m contains c^n distinct clauses, where S_m is of length $O(n)$, $n = m^2$.*

PROOF. By Lemmas 3.1 and 5.6. □

We conclude this section by indicating briefly how the proof may be adapted for examples in which each clause contains exactly three literals. To produce such examples, we transform the graph G_m by replacing each vertex of degree k by a ring of k vertices, joining the resulting rings by edges corresponding to the old vertices in G_m, so that the result is a graph of degree 3. This transformation is due to Kirkpatrick [11] and was used by Galil [8] to produce a set of clauses very similar to the set we are now considering. The proof of the exponential lower bound for the sets of clauses derived from the new sequence of graphs is essentially the same as before, except that, in the definition of the partial solutions and the corresponding root properties, we count the number of rings of vertices rather than the vertices themselves, as in Galil [8].

6. *Short Proofs for S_m in an Axiomatic Calculus*

Although the graph-based examples present considerable difficulties for resolution-based theorem provers, this fact does not reflect any real inherent difficulty in the problems, but rather the inefficient way in which the resolution procedure deals

with biconditionals. This fact is demonstrated by the existence of short refutations for these examples in axiomatic systems of propositional calculus.

An efficient decision procedure for problems involving only biconditionals has been known since the work of Leśniewski in the 1920s (see Church [3, pp. 143, 159] for details). Here we simply verify that Leśniewski's decision procedure can be formalized in a (relatively) efficient manner in a standard axiomatic formulation of propositional calculus.

LEMMA 6.1. *The set of clauses S_m has a refutation of length $O(n^4)$ in Kleene's axiomatic system for propositional calculus.*

PROOF. Our first move is to convert the set of clauses Clauses(x) attached to each vertex of G_m to the corresponding biconditional. This takes $O(n)$ steps. Now by chaining the resulting set of biconditionals form a long biconditional incorporating all these formulas; this part of the proof takes $O(n)$ steps also. By using the associative and commutative laws for the biconditional, we can "bubble up" repeated variables to the front of the big expression and then eliminate the double occurrence. Each "bubbling" sequence takes $O(n^2)$ steps, so that when we finally deduce a contradiction, we have taken $O(n^3)$ steps for our refutation, with each step having length $O(n)$.

THEOREM 6.2. *Resolution cannot p-simulate any Frege systems.*

PROOF. This follows from the preceding theorem and the remarks on the mutual simulation of Frege systems in Section 2. □

REFERENCES

1. BEN-ARI, M. A simplified proof that regular resolution is exponential. *Inf. Proc. Lett. 10* (1980), 96–98.
2. BLAKE, A. Canonical expressions in Boolean algebra. Ph.D. dissertation. Dept. of Mathematics, Univ. of Chicago, Aug. 1937. Reviewed in *J. Symbolic Logic 3* (1938), 93.
3. CHURCH, A. *Introduction to Mathematical Logic,* vol. 1. Princeton Univ. Press, Princeton, N.J., 1956.
4. COOK, S. A. A short proof of the pigeon-hole principle using extended resolution. *ACM SIGACT News 8* (Oct.–Dec. 1976), 28–32.
5. COOK, S. A., AND RECKHOW, R. The relative efficiency of propositional proof systems. *J. Symbolic Logic 44* (1979), 36–50.
6. GABBER, O., AND GALIL, Z. Explicit constructions of linear size superconcentrators. In *Proceedings of the 20th Annual Symposium Foundations of Computing Science.* IEEE, New York, 1979, pp. 364–370.
7. GALIL, Z. On the complexity of regular resolution and the Davis–Putnam procedure. *Theor. Comput. Sci. 4* (1977), 23–46.
8. GALIL, Z. On resolution with clauses of bounded size. *SIAM J. Comput. 6* (1977), 444–459.
9. HAKEN, A. The intractability of resolution. *Theor. Comput. Sci. 39* (1985), 297–308.
10. JOHNSON, N., AND KOTZ, S. *Discrete Distributions.* Houghton Mifflin, New York, 1969; current edition, Wiley. New York, 1970.
11. KIRKPATRICK, D. G. Topics in the complexity of combinatorial algorithms. Ph.D. dissertation, Tech. Rep. No. 74, Dept. of Computer Science, Univ. of Toronto, Toronto, Ont. Canada, Dec. 1974.
12. KLEENE, S. C. *Mathematical Logic.* Wiley, New York, 1967.
13. MARGULIS, G. A. Explicit construction of concentrators. *Prob. Inf. Transm. 9* (1973), 325–332.
14. RECKHOW, R. *On the lengths of proofs in the propositional calculus.* Ph.D. dissertation. Dept. of Computer Science, Univ. of Toronto, Toronto, Ont., Canada, 1975.
15. ROBINSON, J. A. A machine oriented logic based on the resolution principle. *J. ACM 12,* 1 (1965), 23–41. (Reprinted in Reference [16], Vol. 1.)

16. SIEKMANN, J., AND WRIGHTSON, G., EDS. *Automatization of Reasoning.* Springer-Verlag, Berlin, 1983.
17. TSEITIN, G. S. On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic*, Part 2. Consultants Bureau, New York–London, 1968, pp. 115–125. (Reprinted in Reference [16], Vol. 2.)
18. TUTTE, W. T. *Graph Theory.* Addison-Wesley, Reading, Mass., 1984.

RECEIVED JUNE 1985; REVISED JANUARY 1986; ACCEPTED FEBRUARY 1986