Minimum 2CNF Resolution Refutations in Polynomial Time

Joshua Buresh-Oppenheim and David Mitchell

Simon Fraser University
jburesho@cs.sfu.ca, mitchell@cs.sfu.ca

Abstract. We present an algorithm for finding a smallest Resolution refutation of any 2CNF in polynomial time.

1 Introduction

The problem of deciding satisfiability of propositional 2-CNF formulas (2-SAT), is an important tractable case of SAT. The first polynomial-time algorithm for 2-SAT was given by Cook [4]. Linear time algorithms were given by Even Itai and Shamir [6] and, subsequently, Aspvall Plass and Tarjan [2]. For an unsatisfiable formula, a small and simple certificate, or proof of unsatisfiability, may be interpreted as an explanation for the unsatisfiability. Such explanations are central in a number of applications. Cook's algorithm constructs a tree-like Resolution refutation of an unsatisfiable formula. The algorithm of [6], and a later algorithm by del Val [5], involve schemes for applying unit Resolution, and can easily be modified to output tree-like Resolution refutations. The algorithm of [2] provides a certificate in the form of a graph labelled with clauses which are easily seen to be an unsatisfiable subset of the given clauses. A tree-like Resolution refutation can easily be extracted from this graph.

In [3] we gave polytime algorithms for finding a smallest tree-like Resolution refutation and a smallest unsatisfiable subformula of an unsatisfiable 2CNF (the latter is itself an efficiently verifiable certificate since 2SAT is in linear time). Here we give a polytime algorithm for finding a smallest general Resolution refutation. All three algorithms are dynamic programming algorithms based on the implication graph associated with a 2CNF. The algorithm for finding a tree-like refutation runs in time $O(n^2m)$, where n is the number of underlying variables and m is the number of clauses, while the algorithm presented here for general refutations runs in time $O(n^6m)$. In [3] we showed that minimum tree-like Resolution refutations provide a 2-approximation of the smallest general Resolution refutation. Hence, in practice it may often be better to use the faster algorithm to obtain an approximation. Nonetheless, we consider solving the general case an interesting theoretical problem. In particular, we note the contrast with the case of Horn formulas, for which the size of the smallest Resolution refutation is NP-hard to determine, or even to approximate within any constant factor [1]. This difference is especially interesting in light of the similarities in standard algorithms for 2-SAT and Horn-SAT. One may observe that we make essential

use of the symmetries in 2-CNFs deriviation, as exhibited by the "dual" paths in the implication graph, that do not exist in the Horn case.

The algorithms for finding minimum-size certificates provided here and in [3] are more complicated, and of higher time-complexity, than the linear-time algorithms. However, in some applications the size and simplicity of the certificates provided may justify the extra cost. For example, if we have a very large formula and the certificate must be interpreted by a human user, or we have plenty of time to preprocess the formula and the certificate produced will be used repeatedly in the future, then finding a sublinear size certificate, even if it takes a relatively long time, may be better than finding a linear size certificate quickly. Another example is provided by certain abstraction-based model checking techniques in hardware and software verification. At each stage in a sequence of stages a certificate of unsatisfiablility of one formula is used in the creation of a new, larger formula. The size and complexity of the certificates produced is very important in the success of the overall process. In general, the formulas used in this application are not 2CNF formulas, but they often have a very large fraction of 2-clauses. We envision being able to take advantage of the methods for constructing minimum refutations of 2CNF formulas in developing more effective algorithms than currently available for this context.

2 Preliminaries

Throughout, let $\mathcal{C}$ be a collection of 2-clauses (that is, clauses with at most two literals) over an ordered set of variables $\{x_1, ..., x_n\}$. Say $|\mathcal{C}| = m$. As first suggested by [2], $\mathcal{C}$ can be represented as a directed graph $G_{\mathcal{C}}$ on $2n$ nodes, one for each literal. If $(a \vee b) \in \mathcal{C}$ for literals a, b, then the edges $(\bar{a}, b)$ and $(\bar{b}, a)$ appear in $G_{\mathcal{C}}$ (note that literals a and b can be the same). Both of these edges are labelled by the clause $(a \vee b)$. For an edge $e = (a, b)$, let $dual(e)$, the dual edge of e, be the edge $(\bar{b}, \bar{a})$. For literals a, b, define $\mathcal{P}_{ab}$ to be the set of all directed paths from a to b in $G_{\mathcal{C}}$. If c is also a literal, let $\mathcal{P}_{abc}$ be the set of all directed paths that start at a, end at c and visit b at some point. For $P_1 \in \mathcal{P}_{ab}$ and $P_2 \in \mathcal{P}_{bc}$, we denote by $P_1 \circ P_2 \in \mathcal{P}_{abc}$ the concatenation of the two paths. For a path $P = (e_1, ..., e_k) \in \mathcal{P}_{ab}$, let $dual(P) \in \mathcal{P}_{\bar{b}\bar{a}}$ be the path $(dual(e_k), ..., dual(e_1))$.

Proposition 1 ([2]). *$\mathcal{C}$ is unsatisfiable if and only if there is a variable x such that both $\mathcal{P}_{x\bar{x}}$ and $\mathcal{P}_{\bar{x}x}$ are not empty.*

Actually, note that for any literals a, b and variable x, a pair of paths $P_1 \in \mathcal{P}_{a\bar{a}x}$ and $P_2 \in \mathcal{P}_{b\bar{b}\bar{x}}$ are contradictory (for one thing, they imply the existence of a pair of paths such as those in the proposition). This motivates the following definition: Two paths P_1 and P_2 are called *end-contradictory* if there are literals a and b and a variable x (x, $\bar{x}$ need not be distinct from $\bar{a}$, $\bar{b}$) such that $P_1 \in \mathcal{P}_{a\bar{a}x}$ and $P_2 \in \mathcal{P}_{b\bar{b}\bar{x}}$.

We will be interested in finding such pairs of paths of a particularly simple form. First we will need to establish several definitions about directed paths in $G_{\mathcal{C}}$. Note that in $G_{\mathcal{C}}$ even a simple path may contain two edges with the same

clause label. Let $clauses(P)$ denote the set of clause-labels underlying the edges of a directed path P. We define $|P|$, the *size* of the path P, to be $|clauses(P)|$. In contrast, let $length(P)$ denote the length of P as a sequence. Call a path P *singular* if it does not contain two edges that have the same clause label. Therefore, a path P is singular if and only if $|P| = length(P)$. Given two paths P_1, P_2, let $\ell(P_1, P_2)$ denote the quantity $|clauses(P_1) \cup clauses(P_2)|$.

Definition 1. *Let $suf(P)$ be the maximal singular suffix of P. For a path $P \in \mathcal{P}_{a\bar{a}b}$ ($\bar{a}$ and b need not be distinct), let $extend(P)$ be the following path in $\mathcal{P}_{\bar{b}b}$: let P' be the portion of P that starts at the last occurrence of $\bar{a}$ and goes to the end. Then $extend(P)$ is $dual(P') \circ P$.*

Definition 2. *Given a path $P \in \mathcal{P}_{a\bar{a}b}$, let $sing(P)$ be the following operation: first let $P' = extend(P)$. Now, while there is a repeated edge in P', remove the segment of P' after the first occurrence of the edge through the second occurrence. When there is no longer a repeated edge, take the suffix of the resulting path.*

It is clear that $sing(P)$ is singular and that $clauses(sing(P)) \subseteq clauses(P)$. Also, if P_1 and P_2 are end-contradictory, then so are $sing(P_1)$ and $sing(P_2)$ and $\ell(sing(P_1), sing(P_2)) \leq \ell(P_1, P_2)$.

Definition 3. *Let P be a singular path that starts at literal a. Define $core(P)$ as the subpath of P that starts at a and ends at the first occurrence of $\bar{a}$ (or at the end of P if there is none).*

A *segment* of a path is a consecutive subsequence of the path's sequence. For two singular paths P_1 and P_2, a *primal shared segment* is a common segment. A *dual shared segment* of P_1 with respect to P_2 is a segment t of P_1 such that $dual(t)$ is a segment of P_2. A *shared segment* is either a primal or dual shared segment. For two disjoint segments s and t of P, say $s \prec_P t$ if s appears before t in P. For two singular paths P_1 and P_2, let $k(P_1, P_2)$ be the number of maximal shared segments (primal or dual) of P_1 and P_2.

We assume the reader is familiar with Resolution derivations. We simply mention that Resolution derivations can be viewed as DAGs whose nodes are the clauses in the derivation (we assume all occurrences of a particular clause are identified to one node). In a derivation of a single clause C, C is the only source and the sinks are the axioms used in the derivation. Each non-axiom clause has fanout two: it points to the two clauses whose resolvent it is. A Resolution refutation is a derivation of the empty clause Λ. The size of a derivation is the number of clauses (nodes) in it.

Proposition 2. *Any Resolution derivation of a single clause that uses ℓ axioms must have size at least $2\ell - 1$.*

Let $P \in \mathcal{P}_{ab}$. Let $IR(P)$ be the Input Resolution derivation that starts by resolving the clauses labelling the first two edges in P and then proceeds by resolving the latest derived clause with the clause labelling the next edge in the sequence P. This is a derivation of either $(\bar{a} \vee b)$ or simply (b) (if the path goes through literal $\bar{a}$). It is not hard to see that the size of the derivation $IR(P)$ is $2 \cdot length(P) - 1$.

3 Characterizing Minimum Resolution Refutations

Let π be a Resolution derivation from $\mathcal{C}$ that includes the clause $(\bar{a} \vee b)$. Then π defines a path in $G_{\mathcal{C}}$ from a to b (and from $\bar{b}$ to $\bar{a}$). The underlying edges of this path are exactly the elements of $\mathcal{C}$ that appear as sinks in π and are reachable from $(\bar{a} \vee b)$. More formally, we have the following definition:

Definition 4. *Let a, b be literals over distinct variables. Let π be a Resolution derivation containing $(\bar{a} \vee b)$. If $(\bar{a} \vee b)$ is a sink in π, then let $ResPath(\pi, (\bar{a} \vee b), a \rightarrow b)$ equal the edge (a, b), and let $ResPath(\pi, (\bar{a} \vee b), \bar{b} \rightarrow \bar{a})$ equal the edge $(\bar{b}, \bar{a})$. Otherwise, assume $(\bar{a} \vee b)$ has children $(\bar{a} \vee c)$ and $(\bar{c} \vee b)$, for some literal c, in π. Then set $ResPath(\pi, (\bar{a} \vee b), a \rightarrow b)$ to $ResPath(\pi, (\bar{a} \vee c), a \rightarrow c) \circ ResPath(\pi, (\bar{c} \vee b), c \rightarrow b)$. Set $ResPath(\pi, (\bar{a} \vee b), \bar{b} \rightarrow \bar{a})$ to $ResPath(\pi, (\bar{c} \vee b), \bar{b} \rightarrow \bar{c}) \circ ResPath(\pi, (\bar{a} \vee c), \bar{c} \rightarrow \bar{a})$. If the variable underlying a precedes the variable underlying b in the order of variables, then let $ResPath(\pi, (\bar{a} \vee b)) = ResPath(\pi, (\bar{a} \vee b), a \rightarrow b)$.*

Now assume that the clause (a) appears in some Resolution derivation π. Again, if (a) is a sink, let $ResPath(\pi, (a))$ be the edge $(\bar{a}, a)$. Otherwise, if the children of (a) are $(a \vee x)$ and $(a \vee \bar{x})$ for some variable x, then set $ResPath(\pi, (a))$ to $ResPath(\pi, (a \vee x), \bar{a} \rightarrow x) \circ ResPath(\pi, (a \vee \bar{x}), x \rightarrow a)$. Otherwise, if the children of (a) are $(a \vee b)$ and $(\bar{b})$ for some literal b, then set $ResPath(\pi, (a))$ to $ResPath(\pi, (\bar{b})) \circ ResPath(\pi, (a \vee b), \bar{b} \rightarrow a)$.

Finally, given a Resolution refutation π that ends by resolving (x) and $(\bar{x})$, let $ResPath(\pi)$ be the pair $(ResPath(\pi, x), ResPath(\pi, \bar{x}))$.

Notice that, for a Resolution refutation π, the pair of paths in $ResPath(\pi)$ are end-contraditory. This justifies our strategy of reducing the search for a minimum Resolution refutation to a search for a pair of end-contradictory paths that satisfy certain criteria.

Definition 4 demonstrates that there is a pretty deep correspondence between Resolution derivations over $\mathcal{C}$ and paths in $G_{\mathcal{C}}$. Will we exploit this correspondence heavily throughout, but here we pause to illustrate one salient aspect of it. Consider a fragment of a Resolution derivation π such as that in figure 1. Let $Q = ResPath(\pi, C)$. Then, going backwards along the main path in the derivation, each successive clause C_i corresponds to an extension of the segment Q, called Q_i. In particular, the resolution with each clause D_i extends Q_i either from its beginning or from its end.

Definition 5. *A* joint derivation *of two clauses $(\bar{a} \vee b)$ and $(\bar{c} \vee d)$ (again, $\bar{a}$, $\bar{c}$ need not be distinct from b, d) from $\mathcal{C}$ is a Resolution derivation that uses $\mathcal{C}$ as axioms and such that $(\bar{a} \vee b)$ and $(\bar{c} \vee d)$ appear in the derivation and are the only clauses with fanin 0.*

Definition 6. *Consider a joint derivation π of $(\bar{a} \vee b)$ and $(\bar{c} \vee d)$ from $\mathcal{C}$. A* shared clause *in this derivation is any clause C in π such that there are paths in π from $(\bar{a} \vee b)$ to C and from $(\bar{c} \vee d)$ to C, respectively. A* top-shared *clause is a shared clause C such that there is a path from $(\bar{c} \vee d)$ to C that contains no other shared clause.*

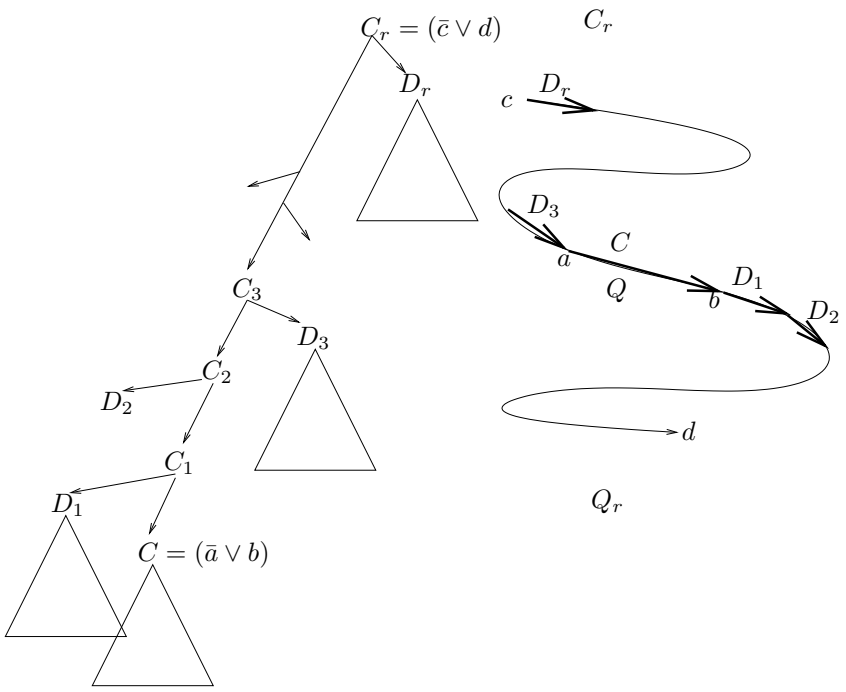

Fig. 1. Extending a path through Resolution.

Lemma 1. *Let π be a joint derivation of $(\bar{a} \vee b)$ and $(\bar{c} \vee d)$ from $\mathcal{C}$ containing ℓ sinks. Assume further that π has k top-shared clauses. Then π has size at least $2\ell + k - 2$.*

Proof. Consider the sinks that are descendants of $(\bar{a} \vee b)$; say there are ℓ_1 of them (and $\ell_2 = \ell - \ell_1$ remaining sinks). Let S be the set of top-shared clauses. All of the descendant sinks of S are among these ℓ_1 sinks. The subgraph induced by all clauses in π reachable from $(\bar{a} \vee b)$ constitutes a Resolution derivation of $(\bar{a} \vee b)$ from ℓ_1 sinks. Therefore, by Proposition 2, this subgraph must contain at least $2\ell_1 - 1$ clauses. Now consider the subgraph induced by all clauses reachable from $(\bar{c} \vee d)$ where we exclude any shared clause that is not a top-shared clause. This constitutes a Resolution derivaton of $(\bar{c} \vee d)$ from $\ell_2 + k$ sinks (modulo removal of edges between top-shared clauses). Therefore, it must contain at least $2(\ell_2 + k) - 1$ clauses. These two derivations (of $(\bar{a} \vee b)$ and $(\bar{c} \vee d)$) are clause disjoint except for the k clauses in S. Therefore, the entire joint derivation contains at least $(2\ell_1 - 1) + (2(\ell_2 + k) - 1) - k = 2\ell + k - 2$ clauses. □

Let $P_1 \in \mathcal{P}_{ab}$ and $P_2 \in \mathcal{P}_{cd}$ and assume that the sum of the lengths of these paths is L. Let $t_1, ..., t_k$ be shared segments (primal or dual) of P_1 and P_2. Define $JointDerive(P_1, P_2, t_1, ..., t_k)$ to be the following joint derivation of $\bar{a} \vee b$ (or possibly just (b)) and $\bar{c} \vee d$ (or possibly just (d)) from $\mathcal{C}$: for each i,

construct $IR(t_i)$; assume this is a derivation of the clause $\bar{x}_i \vee y_i$. Assume that removing the t_i segments from P_1 yields h_1 intermediate nonempty segments $\{r_j\}_{j=1}^{h_1}$. Likewise, there are h_2 intermediate nonempty segments $\{s_j\}_{j=1}^{h_2}$ in P_2. Derive each r_j and s_j using $IR(r_j)$ and $IR(s_j)$. These $k + h_1 + h_2$ derivations have combined size $2(L - \sum_{i=1}^{k} length(t_i)) - (k + h_1 + h_2)$. Now use the results of the t_i and r_j derivations to derive $(\bar{a} \vee b)$ (or (b)) in an input fashion by adding $k + h_1 - 1$ new clauses. Likewise, derive $(\bar{c} \vee d)$ (or just (d)) by adding $k + h_2 - 1$ new clauses. In total, we have $2(L - \sum_{i=1}^{k} length(t_i)) + k - 2$ clauses.

Let $P_1 \in \mathcal{P}_{ab}$ and $P_2 \in \mathcal{P}_{cd}$ be singular. Let $t_1, ..., t_k$ be the maximal shared segments (primal or dual) of P_1 and P_2. Define the *canonical joint derivation* $CJD(P_1, P_2)$ to be $JointDerive(P_1, P_2, t_1, ..., t_k)$.

Lemma 2. *Let $P_1 \in \mathcal{P}_{ab}$ and $P_2 \in \mathcal{P}_{cd}$ be singular paths and assume that if a and b have distinct underlying variables, then a's variable precedes b's in the ordering (likewise for c and d). $CJD(P_1, P_2)$ is a joint derivation of clauses C_1 and C_2, where C_1 is either $(\bar{a} \vee b)$ or just (b), and C_2 is either $(\bar{c} \vee d)$ or just (d). Moreover, $CJD(P_1, P_2)$ has minimum size over all joint derivations π of C_1' and C_2' where $ResPath(\pi, C_1') = P_1$ and $ResPath(\pi, C_2') = P_2$.*

Proof. Consider any joint derivation π of C_1' and C_2'. Let ℓ be the number of distinct axioms underlying P_1 and P_2 and let k be the number of maximal shared segments. π must have at least ℓ sinks. If π has at least k top-shared clauses it cannot have size smaller than $CJD(P_1, P_2)$ by Lemma 1. Now assume it has $k' < k$ top-shared clauses. Each top-shared clause corresponds to a shared segment of P_1 and P_2. The other shared clauses correspond to subsegments of these shared segments. Therefore, there must be $k - k'$ maximal shared segments such that no subsegment is represented by a shared clause in π. Each such maximal shared segment contains at least one axiom which is not shared in π. Therefore, the number of sinks in π is at least $\ell + k - k'$, so π must have size at least $2(\ell + k - k') + k' - 2 = 2\ell + k - 2 + (k - k') > 2\ell + k - 2$. □

We now show the crucial fact that the paths underlying a minimum Resolution refutation are, without loss of generality, singular. The proof goes by a fairly intense case analysis, which we only sketch here. We do, however, offer some intuition. In [3], we show that, for any derivable, nonempty clause C, there is a smallest derivation of C that is $IR(P)$ for some singular path P in $G_{\mathcal{C}}$. In other words, multiple use of clauses, even axioms, is not helpful. A Resolution refutation is essentially a joint derivation of (x) and $(\bar{x})$ for some variable x. As also shown in [3], independent minimum derivations of (x) and $(\bar{x})$ are sometimes almost twice as large as the minimum joint derivation of the two, so the sharing of clauses between the two derivations can be crucial. Here we simply rule out any benefit of sharing a clause within one side (e.g. the portion used to derive (x)) of the joint derivation.

Lemma 3. *Assume there is a Resolution refutation, π, of $\mathcal{C}$ of size s. Then there is a Resolution refutation of $\mathcal{C}$, π', of size $\leq s$, such that both paths in $ResPath(\pi')$ are singular.*

Proof (sketch). Assume π ends by resolving x and $\bar{x}$. If either $ResPath(\pi, (x))$ or $ResPath(\pi, (\bar{x}))$ is not singular, then there is a clause C in π such that there are at least two paths from x to C or from $\bar{x}$ to C, respectively. Call such a clause *repeated*. If there are k distinct paths from x to C, we say that C is repeated k times with respect to x, or that C has k occurrences with respect to x.

Let C be a repeated clause in π that has no repeated ancestor (if there are no repeated clauses, we are done). We will show how to locally transform π so that we eliminate one occurrence of C and do not add occurrences of any other clause.

Assume without loss of generality that C is repeated with respect to x. Let D be an ancestor of C in π such that there are exactly two distinct paths from D to C and such that no descendant of D has two distinct paths to C. Let r_1 and r_2 denote the two paths from D to C. C must have two distinct literals, say, $(\bar{c} \vee d)$. It may be the case that there is one clause, C_2, on r_2 such that there is one path from $\bar{x}$ to C_2 that is edge-disjoint from r_2 (likewise for C_1 and r_1). There cannot be more than one such clause or one such path by the way we chose C. We will generally assume that C_1 and C_2 exist since the proof is simpler if they don't. Therfore, let D' be a clause reachable from $\bar{x}$ such that there is a path from D' to C_1 (call it r_3) and a node-disjoint path from D' to C_2 (call it r_4). Let r_{31} be r_3 concatenated with the suffix of r_1 from C_1 to C, and let r_{42} be r_4 concatenated with the suffix of r_2 from C_2 to C. So r_{31} and r_{42} are the two distinct paths from D' to C. This entire setup is illustrated in figure 2. We will assume for simplicity that both D and D' contain two distinct literals; the proof is similar if they don't.

Let $Q = ResPath(\pi, C)$, $P = ResPath(\pi, D)$ and $P' = ResPath(\pi, D')$. There are several cases based on how C occurs in P and P'. For instance, the r_1 occurrence of C corresponds to a segment of P that is either Q or $dual(Q)$. Also, the r_1 occurrence of C could either precede or succeed the r_2 occurrence in P. We illustrate one case: assume that the r_2 occurrence of C succeeds the r_1 occurrence in P and that both are Q. Assume that the r_{42} occurrence of C succeeds the r_{31} occurrence in P' and that the r_{42} occurrence is Q while the r_{31} occurrence is $dual(Q)$ (see figure 3).

As described above (after Definition 4), each resolution along, say, path r_2 from C to D corresponds to an extension of (an extension of) the r_2 occurrence of Q. Call a clause in π a neighbor of r_2 if it is a child of any clause in r_2 (except C), but is not in r_2 itself. Let $B^2_1, B^2_2, ..., B^2_{b_2}$ be the neighbors of r_2 that correspond to extending the r_2 occurrence of Q towards the beginning of P and let $E^2_1, ..., E^2_{e_2}$ be the neighbors of r_2 that correspond to extending the r_2 occurrence of Q towards the end of P. Likewise for r_1 and $B^1_1, ..., B^1_{b_1}$ and $E^1_1, ..., E^1_{e_1}$, respectively. Let $B^3_1, ..., B^3_{b_3}$ and $E^3_1, ..., E^3_{e_3}$ be the neighbors of r_3 that extend the r_{31} occurrence of $dual(Q)$ towards the beginning and end of P', respectively (likewise for $B^4_1, ..., B^4_{b_4}$, $E^4_1, ..., E^4_{e_4}$ and the r_{41} occurrence of Q).

Let $IR(B^1)$ be the input derivation that proceeds by resolving $B^1_1, ..., B^1_{b_1}$ in order and let B^1 denote the final clause in this derivation (likewise for all the B^i's and E^i's). It must be the case that D is the result of resolving C with B^1 and E^2

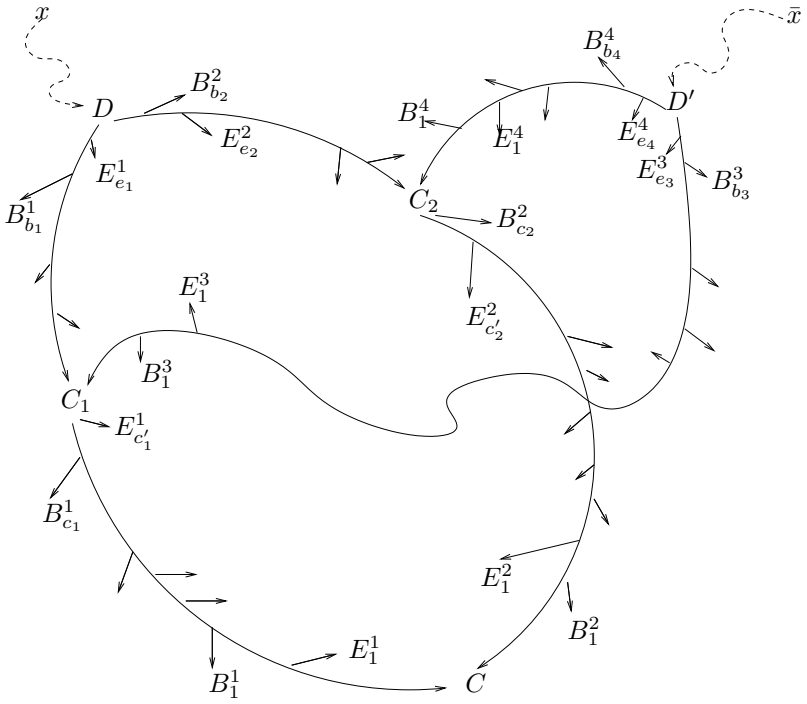

Fig. 2. Original derivation.

(see figure 4). To derive D' (or, in fact, something stronger), let $B^1(c_1)$ denote the $(c_1 - 1)$th derived clause in B^1 and let E^{31} denote the result of resolving $B^1(c_1)$ with E^3. Likewise, let B^{42} denote the result of resolving $B^2(c_2)$ with B^4. Finally, let E^{42} denote the result of resolving $E^2(c'_2)$ with E^4. The clause that results from resolving C with E^{31}, B^{42} and E^{42} successively must be a subclause of D'. Now we must compare the size of the modified derivation with the size of the original derivation. In the original, each B^i_j and E^i_j clause gives rise to a new derived clause, so there are $K = \sum_{i=1}^{4} b_i + e_i$ derived clauses along the paths $r_1, ..., r_4$. In the modified derivation, the total number of derived clauses in the input derivations B^i and E^i is $\sum_{i=1}^{4}(b_i - 1) + (e_i - 1) = K - 8$. To finish deriving D, we create one intermediate derived clause; to finish D', we create five. Therefore the modified derivation is no bigger.

□

Now we show that we can assume the pair of singular paths underlying a minimum Resolution refutation obeys special properties. In light of Lemma 2, we call a pair of singular, end-contradictory paths P_1, P_2 *minimum* if they minimize the expression $f(P_1, P_2) \equiv 2\ell(P_1, P_2) + k(P_1, P_2) - 1$. In other words, they generate a minimum size refutation (the -1 term in the expression replaces the -2 in the size of $CJD(P_1, P_2)$ because we count the empty clause).

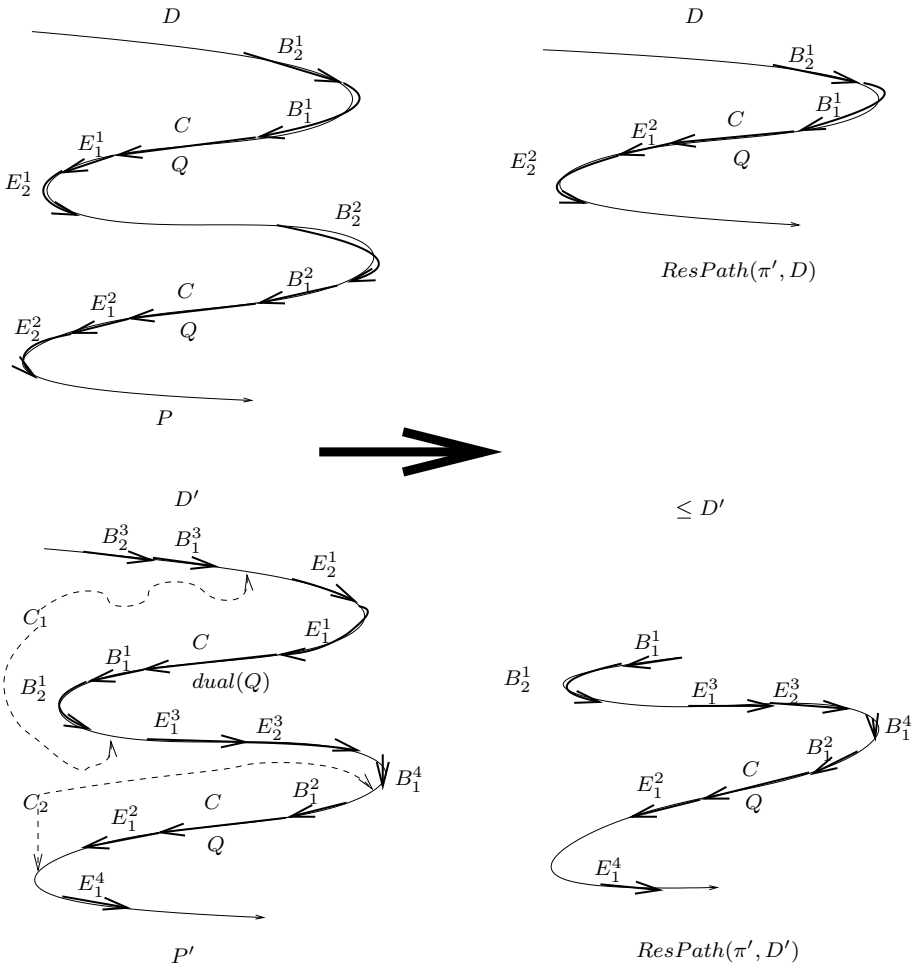

Fig. 3. Transformation of paths.

Consider the following properties of two singular paths P_1 and P_2.
Property I: Let $s_1 \prec_{P_1} \cdots \prec_{P_1} s_k$ be the maximal primal shared segments of P_1 and P_2. Then $s_k \prec_{P_2} \cdots \prec_{P_2} s_1$.
Property II: Let $t_1 \prec_{P_1} \cdots \prec_{P_1} t_\ell$ be the maximal dual shared segments of P_1 with respect to P_2. Then $dual(t_1) \prec_{P_2} \cdots \prec_{P_2} dual(t_\ell)$.
Property III: Let $s_1 \prec_{P_1} \cdots \prec_{P_1} s_k$ be the maximal primal shared segments of P_1 and P_2 and let $t_1 \prec_{P_1} \cdots \prec_{P_1} t_\ell$ be the maximal dual shared segments of P_1 with respect to P_2. For any i, j, $t_i \prec_{P_1} s_j$ if and only if $dual(t_i) \prec_{P_2} s_j$.
Property IV: All shared segments of P_1 and P_2 occur in $core(P_1)$ and $core(P_2)$.

Lemma 4. *Every minimum pair of singular, end-contradictory paths must satisfy Properties I-IV.*

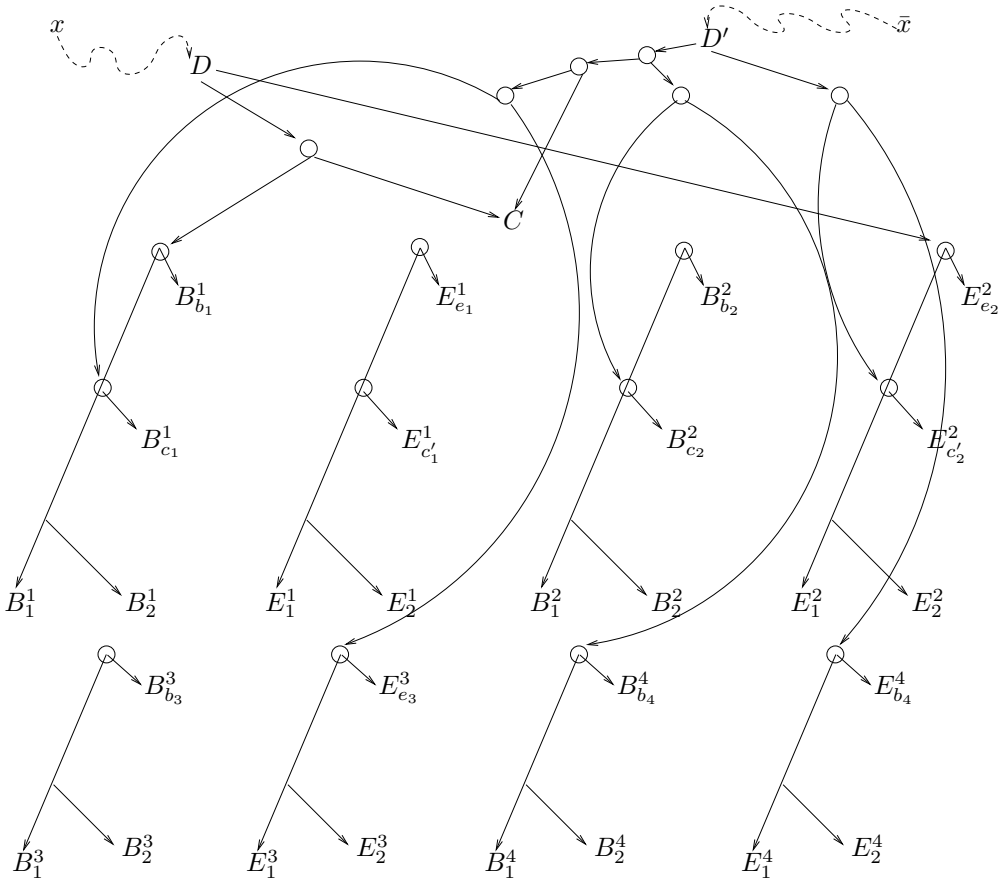

Fig. 4. Modified derivation.

Proof. Our general strategy is to take a pair of singular, end-contradictory paths P_1, P_2 that violate one of the properties and transform them into a pair of singular, end-contradictory paths P'_1, P'_2 such that $f(P'_1, P'_2) < f(P_1, P_2)$.

Consider Property I. If P_1 and P_2 violate the property, then there is some $i < j$ such that $s_i \prec_{P_2} s_j$. Let P'_1 be the segment of P_1 starting at the beginning of s_i and ending at the end of s_j. Likewise, let P'_2 be the segment of P_2 that starts at the beginning of s_i and ends at the end of s_j. Assume, without loss of generality, that $\mathit{length}(P'_1) \leq \mathit{length}(P'_2)$. Let P''_2 be the path P_2 with P'_2 replaced by P'_1. It must be the case that P''_2 is singular since otherwise there would have been a shared segment in between s_i and s_j in P_2. Futhermore, P_1 and P''_2 are clearly end-contradictory. Finally, $f(P_1, P''_2) < f(P_1, P_2)$ since both the number of underlying clauses and the number of maximal shared segments have gone down. Property II follows in the same way by looking at P_1 and $\mathit{dual}(P_2)$.

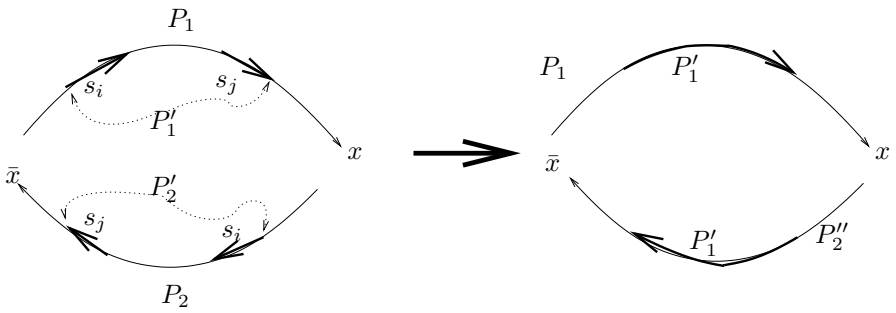

Fig. 5. Forcing Property I.

Consider Property IV. Let P_1, P_2 be singular, end-contradictory paths. Assume, without loss of generality, that $core(P_1) \neq P_1$. Let $core(P_1)$ go from a to $\bar{a}$ and let P_1 end at x. Assume P_2 starts at b and ends at $\bar{x}$ (b may equal x). Let s be the maximal shared segment that ends as late as possible in P_1. Assume s goes from c to d such that d occurs after $\bar{a}$ in P_1. If s is a primal shared segment, then let P_1' be the segment of P_1 that goes from a to d. Let Q_1 be the segment of P_1 that goes from d to x and let Q_2 be the segment of P_2 that goes from d to $\bar{x}$. Let $P_2' = Q_2 \circ dual(Q_1)$. Note that P_1' and P_2' are end-contradictory and singular. Also, $f(P_1', P_2') < f(P_1, P_2)$ since the number of shared segments has gone down. If s is a dual shared segment, then again let P_1' be the segment of P_1 that goes from a to d. If $\bar{d}$ occurs at or after $\bar{b}$ in P_2, then let P_2' be the segment of P_2 that goes from b to $\bar{d}$. Otherwise, let Q_1 be the segment of P_1 from d to x. Let Q_2 be the (possibly empty) segment of P_2 from $\bar{b}$ to $\bar{x}$, and let Q_3 be the segment of P_2 from b to $\bar{d}$. Set $P_2' = Q_1 \circ dual(Q_2) \circ Q_3$. Again, P_1' and P_2' are singular and end-contradictory and $f(P_1', P_2') < f(P_1, P_2)$.

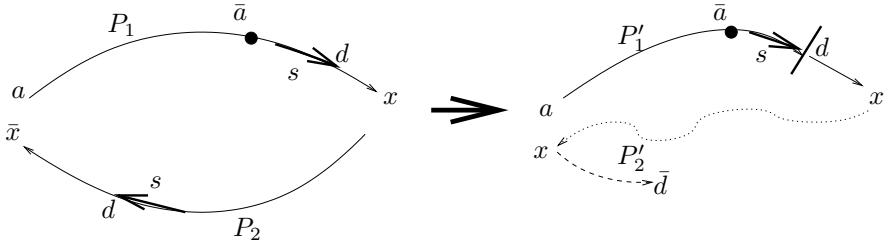

Fig. 6. Forcing Property IV.

Finally, consider Property III. We assume that Properties I, II and IV hold. If P_1 and P_2 violate the property, then there is a primal shared segment s and a dual shared segment t such that, without loss of generality, $t \prec_{P_1} s$, but $s \prec_{P_2} dual(t)$, and furthermore there are no shared segments between t and s

in P_1. Let c, d be the endpoints of t, and g, h the endpoints of s. Let Q_1 be the segment of P_1 from d to h, and let Q_2 be the segment of P_2 from h to $\bar{c}$. Let $P_1' = Q_1 \circ Q_2$. Note that P_1' is singular. Let Q_3 be the segment of P_1 from the end of $core(P_1)$ to the end of P_1 (say P_1 ends at x). Let Q_4 be the segment of P_2 from $\bar{c}$ to the end and let Q_5 be the segment of P_1 from the beginning to c. Let $P_2' = Q_4 \circ dual(Q_3) \circ Q_5$. P_2' may not be singular, so let $P_2'' = sing(P_2')$. Clearly P_1' and P_2'' are end-contradictory and singular. Also, $f(P_1', P_2'') < f(P_1, P_2)$.

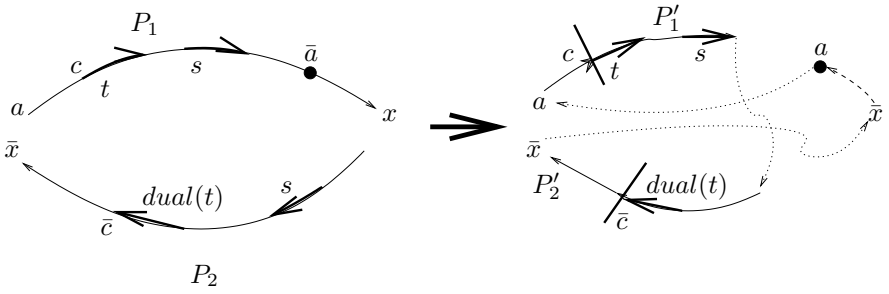

Fig. 7. Forcing Property III.

□

4 The Algorithm

Our algorithm for finding a minimum Resolution refutation will use dynamic programming in a similar way that, say, the Bellman-Ford algorithm does. It would be sufficient to find a minimum pair of singular, end-contradictory paths P_1, P_2, but it is unclear how to limit our search to singular paths, since arbitrary extensions of singular paths are not necessarily singular. On the other hand, if we have two non-singular, end-contradictory paths, there does not seem to be a simple characterization of the size of a smallest Resolution refutation in terms of the lengths of the paths and the lengths of any shared segments. We get around this problem by defining a generalized cost of two arbitrary paths such that the cost is at least the size of the minimum joint derivation based on the paths, but is equal to this size in the case where both paths are singular. Therefore, optimizing over all pairs of end-contradictory paths with respect to this generalized cost must find a minimum since we know that the minimum is achieved by a pair of singular paths.

Another ingredient to the algorithm is that we can focus on pairs of paths that obey properties I-IV (we will explain what this means for non-singular paths shortly). In particular, the structure provided by these properties allows us to do dynamic programming where the recursion is on the number of shared segments between a pair of paths. The recursion is based on the following idea. The reason a pair of paths P_1 and P_2 that minimize the cost function may not each be of

minimum length is that, while longer, they benefit by sharing more clauses. If we demand that P_1 and P_2 have a shared segment with specified endpoints, however, then that segment should be as short as possible; likewise, for any segment of, say, P_1 with specified endpoints that is guaranteed not to overlap any shared segment. By doing this, we isolate segments of P_1 and P_2 that we can locally optimize and then concentrate on the remainder of the paths.

For two paths P_1 and P_2, define $cost(P_1, P_2, k)$ to be the minimum of the expression

$$length(P_1) + length(P_2) - \sum_{i=1}^{r} length(s_i) - \sum_{j=1}^{q} length(t_j)$$

over all choices of $s_1, ..., s_r, t_1, ..., t_q$, $r + q = k$, such that $s_1 \prec_{P_1} \cdots \prec_{P_1} s_r$ are (possibly empty) primal shared segments of P_1 and P_2, $t_1 \prec_{P_1} \cdots \prec_{P_2} t_q$ are (possibly empty) dual shared segments of P_1 with respect to P_2, all of the s_i's and t_j's are edge-disjoint from one another and they obey Properties I-III. Given four literals a, b, c, d and a natural number k, define $cost(a, b, c, d, k)$ to be the minimum over all paths $P_1 \in \mathcal{P}_{ab}$ and $P_2 \in \mathcal{P}_{cd}$ of $cost(P_1, P_2, k)$.

The algorithm will compute $cost(a, b, c, d, k)$ for all literals a, b, c, d and all $0 \leq k \leq m$, and will store with each entry a pair of paths and set of shared segments that achieve that cost. To find a minimum Resolution refutation, we search for litarals a, b, x and a number k that minimize

$$2(cost(a, \bar{a}, b, \bar{b}, k) + cost(\bar{a}, x, \bar{b}, \bar{x}, 0)) + k - 1.$$

The reason for the two *cost* terms is Property IV, which assures us that we need not consider any shared segments outside of the cores of the paths. For fixed k, let P_1, P_2 be the pair of paths that minimize the first term in this expression and let $s_1, ..., s_r, t_1, ..., t_q$ be the shared segments. Let P_1', P_2' be the paths that minimize the second term in this expression. Let $Q_1 = P_1 \circ P_1'$ and let $Q_2 = P_2 \circ P_2'$. Then $JointDerive(Q_1, Q_2, s_1, ..., s_r, t_1, ..., t_q)$ is minimum for this value of k. We then simply optimize over all values of k.

To begin, for all literals a, b, set $B[a, b]$ to the length of a shortest path in $\mathcal{P}_{ab}$. This can be done using Bellman-Ford, for example. For all literals a, b, c, d, set $cost(a, b, c, d, 0)$ to $B[a, b] + B[c, d]$. To compute a general entry in $cost()$ where k is nonzero, let P_1 and P_2 be the paths that achieve the minimum corresponding to the entry in question. By Properties I-III, there are three cases. **(1)** There are no dual shared segments of P_1 with respect to P_2. Therefore, the first shared segment in P_1 (in order of appearance) is a primal shared segment s_1 that is the last shared segment in P_2. **(2)** The first shared segment in P_1 is a dual shared segment t_1 and $dual(t_1)$ is the first shared segment in P_2. **(3)** The last shared segment in P_1 is a dual shared segment t_q and $dual(t_q)$ is the last shared segment in P_2.

Therefore, to compute $cost(a, b, c, d, k)$, we take the minimum over all literals x, y of the minimum of **(1)** $B[a, x] + B[y, d] + B[x, y] + cost(y, b, c, x, k - 1)$; **(2)** $B[a, x] + B[b, \bar{y}] + B[x, y] + cost(y, b, \bar{x}, d, k - 1)$; **(3)** $B[y, b] + B[\bar{x}, d] + B[x, y] +$

$cost(a, x, c, \bar{y}, k-1)$. The algorithm for computing the size of a smallest Resolution refutation is summarized in figure 8. It is not hard to see that it runs in time $O(n^6 m)$. As mentioned above, one can produce a minimum refutation by keeping track of the paths and shared segments that achieve the minima. This adds nothing to the asymptotic complexity.

For all literals a, b
  $B[a, b] \longleftarrow \min\{length(P) \mid P \in \mathcal{P}_{ab}\}$
For all literals a, b, c, d
  $cost(a, b, c, d, 0) \longleftarrow B[a, b] + B[c, d]$
For $k = 1$ to m do
  For all literals a, b, c, d
    For all literals x, y
      $tmp \longleftarrow \min\{B[a, x] + B[y, d] + B[x, y] + cost(y, b, c, x, k-1),$
        $B[a, x] + B[b, \bar{y}] + B[x, y] + cost(y, b, \bar{x}, d, k-1),$
        $B[y, b] + B[\bar{x}, d] + B[x, y] + cost(a, x, c, \bar{y}, k-1)\}$
      If $tmp < cost(a, b, c, d, k)$ then $cost(a, b, c, d, k) \longleftarrow tmp$

Output $\min_{0 \le k \le m} \min_{a,b,x} 2(cost(a, \bar{a}, b, \bar{b}, k) + cost(\bar{a}, x, \bar{b}, \bar{x}, 0)) + k - 1$

Fig. 8. Computing the size.

References

1. M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is NP-hard to linearly approximate. *JSL: Journal of Symbolic Logic*, 66, 2001.
2. Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, March 1979.
3. J. Buresh-Oppenheim and D. Mitchell. Minimum witnesses for unsatisfiable 2CNFs. In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2006.
4. S.A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.
5. Alvaro del Val. On 2-SAT and Renamable Horn. In *AAAI'2000, Proc. 17th (U.S.) National Conference on Artificial Intelligence*. AAAI Press/The MIT Press, 2000.
6. S. Even, A. Itai, and A Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4), 1976.