On the Comparative Complexity of Resolution and the Connection Method*

by
W. Bibel

Technical Report 88-6

February 1988

On the Comparative Complexity of Resolution and the Connection Method*

W. Bibel

University of British Columbia and
Canadian Institute for Advanced Research

Abstract

Quadratic proofs of the pigeonhole formulas are presented using the connection method proof techniques. For this class of formulas exponential lower bounds are known for the length-of-resolution refutations. This indicates a significant difference in the power of these two proof techniques. While short proofs of these formulas are known using extended resolution, this particular proof technique, in contrast to both the connection method and resolution, seems not suitable for the actual proof search.

1 Introduction

Comparing the performance of different proof systems has generally been acknowledged as a very hard research problem. In fact, concrete mathematical results have been achieved only under drastic idealizations that are far from reality. One such idealization focuses on the length of proofs (in propositional logic) as a measure of comparison thus totally ignoring the cost of search for obtaining them (see [2], Section IV.3, for other approaches). This approach has also been driven by its relevance to the P–NP–problem.

It was long conjectured that no short resolution refutations exist of the pigeonhole formulas P_n. These encode the principle that $n+1$ pigeons cannot fit exactly into n holes. Haken [9] has finally settled this conjecture by providing an exponential lower bound on the number of distinct steps needed for the shortest resolution refutation of P_n. Recent papers [12,7] have provided similar results for more classes of formulas.

*submitted

This is certainly not good news for Automated Theorem Proving (ATP), which to a large extent relies on resolution as its main proof technique. Namely, what is the value of a proof technique that already takes an exponential amount of steps just to write down the proof for an intuitively simple problem, let alone has so many more steps usually wasted in the search for the proof. There is little comfort in that there exists a variant of resolution, called *extended* resolution, which offers short proofs of the pigeonhole formulas, as Cook [8] has shown. This is because the extension rule involved in it requires the introduction and appropriate definition of new proposition atoms, for which task there seems to be no mechanism available other than an exhaustive search through the infinite space of all possible formulas.

Each of these two proof techniques thus has a significant defect worth conceptually grasping. Resolution is ***computationally inadequate*** in the sense that it fails to provide a proof in a short (polynomial) amount of time at least for some formulas that are inherently simple (i.e. they permit polynomial proofs in other systems). Extended resolution, on the other hand, generates an ***infinitely branching*** search graph, and for that reason again is computationally inadequate. One would like to have a proof mechanism that, besides being sound and complete also is computationally adequate.

The connection method [2,4], like resolution, has a state–dependent, finitely branching search graph, so it does not suffer from the deficiency experienced with extended resolution. Further it can simulate (the search for) resolution proofs [3], so it is at least as computationally adequate as resolution. The question is whether it has the potential to do significantly better than resolution. The present paper provides a positive answer to this question simply by providing short (quadratic) proofs of the pigeonhole formulas.

The situation for ATP might not be that bad after all. First, even resolution does quite well on a wide range of problems of importance (just think of PROLOG). Second, other techniques such as (some variant of) the connection method still offer the perspective of possibly being computationally adequate.

The following section summarizes the terminology needed for the presentation. Section 3 presents the connection proofs for the pigeonhole formulas and an analysis of their lengths. The final section evaluates the significance of the result.

2 Preliminaries

This paper deals with propositional logic represented in the set-theoretic form commonly used in ATP [2]. That is, we have *variables* x and negated variables

$\bar{x}$ which both are called *literals*. *Clauses* are sets of literals, and *matrices* are sets of clauses. This construction of matrices may be iterated, i.e. clauses may also contain matrices, resulting in *non-normal form* matrices.

Matrices are usefully displayed in a 2-dimensional way as illustrated by the examples in the next section. Thereby an appropriate visual grouping is applied in the case of non–normal form matrices. *Paths* (through a matrix) are sets of literals collected while traversing the matrix from left to right thereby paying attention to the nested grouping just mentioned. A path is *complementary* if it contains the subset $\{x, \bar{x}\}$, called a *connection*, for some variable x. A matrix is *complementary* if all its paths are complementary. Depending on the interpretation the reader prefers, complementarity means *validity* or *inconsistency* of the represented propositional formula.

The connection method performs theorem proving by checking the paths in some systematic way for complementarity. Various such ways have been developed. Section II.3 in [2] describes one using linear chaining with an operation called *extension*. In particular in the more advanced ways, advantage is drawn extensively from the fact that often the complementarity of a path implies that of other paths. There are a number of results known in this respect.

One such result is *Prawitz' reduction* [11] which says the following. If the matrix contains two clauses of the form $\{x\} \cup c$ and $\{\bar{x}\} \cup d$ then all paths containing literals both from c and d may safely be ignored. As a clause substitution rule applicable to non-normal form matrices it may be expressed in the following way.

$$c \cup \{x\},\ d \cup \{\bar{x}\} \ \Rightarrow\ \{\{d, \{x\}\}, \{c, \{\bar{x}\}\}\}$$

Other such results are the following well–known reduction rules.

PURE. Any clause containing a literal not contained in any connection may be deleted.

SUBSUMPTION. Any clause containing another one may be deleted.

UNIT. If a matrix contains a *unit* clause (with one literal only), then, after *resolving* upon all connections containing this literal, all clauses *naming* its variable may be deleted.

ISOL. If the literals in a connection are not contained in any other connection, then, after resolving upon this connection, the parent clauses may be deleted.

FACTOR. If a number of clauses share a common variable, they may be replaced by a single clause with one occurrence of this literal obtained by application of the associativity law.

MONOTONE. A path containing another one may be ignored. Consequently, if a matrix is a superset of another one, its complementarity is implied by that of the smaller one.

RENAME. Consistent renaming of variables does not affect complementarity.

Each of these reductions is illustrated in the subsequent section. Note that they can be executed fast so that their costs are sort of negligeable. In what follows, we are dealing with an infinite sequence P_n, $n \geq 1$, of sets of clauses that is defined in the following way.

$$P_n = (\bigcup_{i=1}^{n+1} \{\{\bar{x}_{1i}, \ldots, \bar{x}_{ni}\}\}) \cup (\bigcup_{i=1}^{n} \bigcup_{1 \leq j < k \leq n+1} \{\{x_{ij}, x_{ik}\}\})$$

The set of the first $n+1$ (purely negative) clauses is abbreviated by P_n^-, that of the remaining $(n^3 + n^2)/2$ ones by P_n^+. P_n contains $n \cdot (n+1)$ variables. The matrices for P_1 and P_2 are depicted in the next section. P_n encodes the statement that $n+1$ 'pigeons' can fit exactly into n 'holes' which obviously expresses an inconsistency. For this encoding, x_{ij} may be read as 'hole i is occupied by pigeon j'.

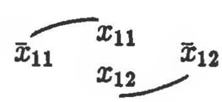
3 Short Connection Proofs for P_n

In the present section we prove the following result.

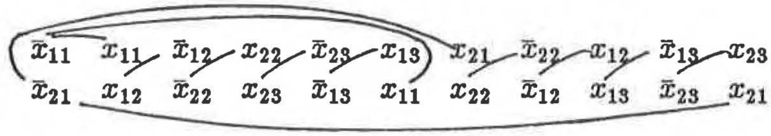
Lemma. There is a short proof for P_n in the connection method, the length of which is quadratic in the number of clauses in P_n, $n \geq 1$.

The proof is by induction on n.

There are two extensions needed for P_1 as illustrated by the two connections in its matrix representation.



There are ten extensions needed for P_2 as illustrated again by the twelve resulting connections in its matrix representation where two clauses are listed twice for an easier depiction.



So, as the induction hypothesis, let us assume that there is a proof for P_{n-1} of appropriate length. We will reduce P_n to P_{n-1} by a number of reduction operations. Although this will, of course, be done for arbitrary n, each of these operations will at the same time be illustrated for the generic case $n = 3$. Its matrix P_3 is the following one.

$$
\begin{array}{cccc}
\bar{x}_{11} & \bar{x}_{12} & \bar{x}_{13} & \bar{x}_{14} \\
\bar{x}_{21} & \bar{x}_{22} & \bar{x}_{23} & \bar{x}_{24} \\
\bar{x}_{31} & \bar{x}_{32} & \bar{x}_{33} & \bar{x}_{34}
\end{array}
\quad
\begin{array}{cccccccccccccccccc}
x_{11} & x_{11} & x_{11} & x_{12} & x_{12} & x_{13} & x_{21} & x_{21} & x_{21} & x_{22} & x_{22} & x_{23} & x_{31} & x_{31} & x_{31} & x_{32} & x_{32} & x_{33} \\
x_{12} & x_{13} & x_{14} & x_{13} & x_{14} & x_{14} & x_{22} & x_{23} & x_{24} & x_{23} & x_{24} & x_{24} & x_{32} & x_{33} & x_{34} & x_{33} & x_{34} & x_{34}
\end{array}
$$

The reduction of P_n to P_{n-1} will in turn be proved by induction. For that purpose we consider the following sequence of matrices Q_{nm}, $m = 1, \ldots, n+1$.

$$
\begin{aligned}
&\Big(\bigcup_{i=1}^{m}\{\{\bar{x}_{1i},\ldots,\bar{x}_{ni}\}\}\Big) \cup \Big(\bigcup_{i=m+1}^{n+1}\{\{\bar{x}_{1i},\ldots,\bar{x}_{n-1i}\}\}\Big) \\
&\qquad \cup \Big(\bigcup_{i=1}^{n-1}\ \bigcup_{1\le j<k\le n+1}\{\{x_{ij},x_{ik}\}\}\Big) \cup \Big(\bigcup_{i=n}^{n}\ \bigcup_{1\le j<k\le m}\{\{x_{ij},x_{ik}\}\}\Big)
\end{aligned}
$$

By comparison with the definition of P_n, given in Section 2 it is obvious from this representation that $Q_{n(n+1)} = P_n$. The special case $n = 3$ will be displayed in matrix form shortly.

The base case for the induction on m will be handled further below. So we now turn our attention to the induction step in which Q_{nm} will be reduced to $Q_{n(m-1)}$. For that purpose we focus our attention to those clauses in Q_{nm} that contain the variable x_{nm} which are the following ones.

$$
\begin{array}{c}
\bar{x}_{1m} \\ \vdots \\ \bar{x}_{n-1m} \\ \bar{x}_{nm}
\end{array}
\quad
\begin{array}{ccc}
x_{n1} & \cdots & x_{nm-1} \\
x_{nm} & & x_{nm}
\end{array}
$$

Factoring reduces this to the following submatrix consisting of two clauses.

$$
\begin{array}{c}
\bar{x}_{1m} \\ \vdots \\ \bar{x}_{n-1m} \\ \bar{x}_{nm}
\end{array}
\quad
\begin{array}{c}
x_{n1} \ \cdots \ x_{nm-1} \\
x_{nm}
\end{array}
$$

Prawitz reduction allows us to restrict our attention to the set of paths through the entire matrix that are defined by the following two (left and right) submatrices.

$$
\begin{array}{ccccc}
 & & & \bar{x}_{1m} & \\
x_{n1} & \cdots & x_{nm-1} & \vdots & x_{nm} \\
 & & & \bar{x}_{n-1m} & \\
\bar{x}_{nm} & & & &
\end{array}
$$

Notice that superimposing the left submatrix on the right one results in the previous submatrix. For each of these two submatrices along with the remaining clauses of Q_{nm} complementarity has to be established which may be done independently (unless there is good reason for combining the two tasks). Hence we will talk of the *first*, corresponding to the left submatrix, and the *second* case. In the special case of $Q_{34} = P_3$ the first case is given by the following matrix.

$$
\bar{x}_{34} \quad x_{31} \quad x_{32} \quad x_{33} \quad
\begin{array}{ccc} \bar{x}_{11} & \bar{x}_{12} & \bar{x}_{13} \\ \bar{x}_{21} & \bar{x}_{22} & \bar{x}_{23} \\ \bar{x}_{31} & \bar{x}_{32} & \bar{x}_{33} \end{array}
\quad
\begin{array}{ccccccccc} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{31} & x_{32} \\ x_{14} & x_{14} & x_{14} & x_{24} & x_{24} & x_{24} & x_{32} & x_{33} & x_{33} \end{array}
\quad P_2^+
$$

$\bar{x}_{nm}$ is pure, so this unit clause may be deleted. Similarly, x_{im}, $i = 1, \ldots, n-1$, are all pure as well, hence all clauses containing them may be deleted. Further, all clauses subsumed by the unit clauses x_{ni}, $i = 1, \ldots, m-1$, will be deleted as well. These unit clauses may also be used to remove themselves and the literals $\bar{x}_{ni}$ from the negative clauses $\{\bar{x}_{1i}, \ldots, \bar{x}_{ni}\}$ by ISOL–reduction, $i = 1, \ldots, m-1$. In the resulting matrix we rename the variables by exchanging the names $x_{i(n+1)}$ and x_{im} wherever they occur, $i = 1, \ldots, n-1$. The result is P_{n-1} to which the induction hypothesis (for n) can be applied.

Thus having solved the first case, we now turn our attention to the second one. Again, as an aid to the reader, we depict the matrix for the special case of $Q_{34} = P_3$.

$$
\begin{array}{ccc} \bar{x}_{11} & \bar{x}_{12} & \bar{x}_{13} \\ \bar{x}_{21} & \bar{x}_{22} & \bar{x}_{23} \\ \bar{x}_{31} & \bar{x}_{32} & \bar{x}_{33} \end{array}
\quad
\begin{array}{c} \bar{x}_{14} \\ \bar{x}_{24} \end{array}
\quad x_{34} \quad
\begin{array}{ccccccccc} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{31} & x_{32} \\ x_{14} & x_{14} & x_{14} & x_{24} & x_{24} & x_{24} & x_{32} & x_{33} & x_{33} \end{array}
\quad P_2^+
$$

The variable x_{nm} is pure, whence the unit clause may be deleted. The remaining matrix by construction is just $Q_{n(m-1)}$ as defined further above. Hence in this second case the induction hypothesis (on m) can be applied again.

For illustration of the base case for the m–induction still to be proven, let us present the matrix of Q_{31} which is the following one.

$$
\begin{array}{c} \bar{x}_{11} \\ \bar{x}_{21} \\ \bar{x}_{31} \end{array}
\quad
\begin{array}{ccccccccccccccc} \bar{x}_{12} & \bar{x}_{13} & \bar{x}_{14} & x_{11} & x_{11} & x_{11} & x_{12} & x_{12} & x_{13} & x_{21} & x_{21} & x_{21} & x_{22} & x_{22} & x_{23} \\ \bar{x}_{22} & \bar{x}_{23} & \bar{x}_{24} & x_{12} & x_{13} & x_{14} & x_{13} & x_{14} & x_{14} & x_{22} & x_{23} & x_{24} & x_{23} & x_{24} & x_{24} \end{array}
$$

Q_{n1} contains P_{n-1} as a submatrix upto naming the variables. Indeed, after exchanging the variable names x_{i1} and x_{in+1} wherever they occur, for $i = 1, \ldots, n-1$, we explicitly have $P_{n-1} \subset Q_{n1}$. The monotonicity property thus settles the base case by way of the induction hypothesis, which is the last piece needed for the full proof.

Let us now turn our attention to the length of the connection proof obtained for P_n. For the reduction of P_n to P_{n-1} we had to carry out n steps. Each of these involved identifying and processing the first and the second case. The identification is of length $O(n)$. $O(n^2)$ steps are needed for performing each of these first cases which adds up to $O(n^3)$ steps for this first–case part. This then turns out to be the overall complexity, since it majorizes also the constant amount needed to carry out each of the second cases and the linear time needed for the renaming in the base case. Since n such reduction steps have to be carried out and since the number ℓ of clauses is $O(n^3)$, the overall complexity is $O(\ell \cdot n)$. With a little more effort invested in handling the first case (since it is basically the same each time) and on the actual calculation of the complexity we could have done even with a linear proof, but for the purposes of this paper it would not have made any difference anyway.

4 Conclusions

In the previous section we have presented short proofs of the pigeonhole formulas that are based on proof techniques provided by the connection method. In a sense this amounts to the presentation of a simple exercise. But the significance of this contribution lies not in these proofs themselves, rather in the consequences of their existence for the evaluation of the connection method. In particular, since there are no such proofs for resolution according to Haken's result, these proofs provide the first solid fact establishing a significant computational distinction between resolution and the connection method in favor of the latter.

The crucial feature that avoids the exponential explosion in our proofs lies in their capability to take advantage of subsumption on a global level. If a subproblem is subsumed by another one, (by monotonicity) it can be ignored. Put in a more general (and technical) way, if a path is subsumed by another one in the same problem, it can be ignored during the proof process. Resolution has no such global capabilities. To understand this correctly one has to keep in mind that the resolvent is *not* a subproblem in this sense. Rather one has to add the resolvent to the given set of clauses in order to obtain the resulting subproblem [5].

From a mathematician's viewpoint that feature in essence may be regarded as the capability of recognizing lemmas and applying them more than once. Because of its significance this capability is built into our theorem prover PROTHEO [1] that is based on the connection method.

With the observation of a crucial difference between resolution and the connection method, a belief expressed in [10] (p. 379) turns out to be false for the present state of affairs. The belief was that 'resolution strategies may be able to mimic the matrix reduction strategies of importance'. It is still possible, though, that some future improved form of resolution will catch up again.

The connection proofs of the previous section are presented in a way, that does not exactly follow the connection calculus in [2]. For the purpose of this paper, using the Prawitz matrix reduction rule made it easier for the reader. Since this rule is incorporated in the more advanced forms of the connection method (see [2], Section IV.6), it is obvious that we could as well have stuck strictly to the connection method.

Our result also underlines once again the importance of the simplification operations such as subsumption. In particular, it reminds us of renaming and the global subsumption operation that seem to have been neglected in actual implementations. If a feasible proof mechanism exists at all, it might necessarily consist of a combination of a basic logical operation (such as extension in the connection method or Prawitz reduction) with a (hopefully small) number of special reduction and preprocessing operations. But even if it does not exist, the same might be true for a computationally adequate mechanism (recall this concept from the Introduction) that fails for *intrinsically hard* problems only. Under this aspect one might question the practical relevance of possible future work on the complexity of any proof method unless it takes these operations into account (if they make a difference).

Of course, one would now like to see further natural questions being settled. The most obvious one is, how does the connection method perform on other hard examples for resolution such as those in [12]. Also, which are the hard examples for the connection method (if any — who knows)? Further it seems that the technique of factoring used also in our proofs above has a similar power in the context of the connection method as the generalized matrix reduction method [6]. To see the reason for such a belief, the interested reader might have a look at example 3.6 there. Factoring the variables x_5 and x_6 in that example and pursuing similarly as in the proofs of the previous section, result in the same sort of short proof that is demonstrated there with the generalized matrix reduction method. In fact, factoring along with the possibility of deleting paths containing the empty set may substitute subsumption which leads to the question whether there are more general and fewer principles that permit the substitution of some of the reduction rules mentioned in this paper.

Acknowledgements. I appreciate inspiring discussions with D. Kirkpatrick, H. Levesque, and A. Mackworth.

References

[1] S. Bayerl, E. Eder, F. Kurfess, R. Letz, and J. Schumann. An implementation of a PROLOG–like theorem prover based on the connection method. In Ph. Jorrand and V. Sgurev, editors, *AIMSA'86, Artificial Intelligence II — Methodology, Systems, Applications*, pages 29–36, North–Holland, Amsterdam, 1987.

[2] W. Bibel. *Automated Theorem Proving.* Vieweg Verlag, Braunschweig, second edition, 1987.

[3] W. Bibel. A comparative study of several proof procedures. *Artificial Intelligence*, 12:269–293, 1982.

[4] W. Bibel. Matings in matrices. *Comm. ACM*, 26:844–852, 1983.

[5] W. Bibel. On matrices with connections. *Journal of ACM*, 28:633–645, 1981.

[6] W. Bibel. Tautology testing with a generalized matrix reduction method. *Theor. Comput. Sci.*, 8:31–44, 1979.

[7] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journ. ACM*, to appear.

[8] S. A. Cook. A short proof of the pigeon hole principle using extended resolution. *ACM SIGACT News*, 8:28–32, Oct.-Dec. 1976.

[9] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[10] D.W. Loveland. A unifying view of some linear Herbrand procedures. *J.ACM*, 19:366–384, 1972.

[11] D. Prawitz. A proof procedure with matrix reduction. In M. Laudet et al., editors, *Symposium on Automatic Demonstration, Lecture Notes in Comp. Sci. 125*, pages 207–214, Springer, Berlin, 1970.

[12] A. Urquhart. Hard examples for resolution. *J.ACM*, 34:209–219, 1987.

February 6, 1988