An Exponential Lower Bound For the Pure Literal Rule

By

Khaled M. Bugrara
Northeastern University
Boston, Massachusetts 02115

and

Paul W. Purdom, Jr.
Indiana University
Bloomington, IN 47405

TECHNICAL REPORT NO. 207

An Exponential Lower Bound
For the Pure Literal Rule

by

Khaled M. Bugrara, Northeastern University
and Paul W. Purdom, Jr., Indiana University

December, 1986

An Exponential Lower Bound for the Pure Literal Rule

1. Introduction

A *pure* literal is a literal in a logic formula (usually in Conjunctive Normal Form) that occurs only positively or only negatively. The Davis-Putnam procedure [1] was developed to find one solution to a logic formula, and it contains several techniques for speeding up the typical solution time. One of these techniques is the *pure literal rule*: a variable that occurs only positively or only negatively (a *pure literal*) needs to have only one value considered, the value that makes all of its clauses true.

Goldberg [2] developed an algorithm that consisted of just splitting (trying each value for a variable, simplifying the resulting subproblems, and considering each subproblem recursively) and the pure literal rule. He claimed that this algorithm took polynomial average time. An error was corrected in [3] (concerning the case where a variable did not appear at all), and Goldberg's result was extended to a wider class of random problems. A detailed upper bound analysis [6] showed that there was an extensive region where the average time was polynomial.

This paper has the first lower bound analysis for this algorithm. We show that there is also an extensive region where the average time for this algorithm is exponential. In many cases the current lower bound is much lower than the best upper bound [6]. Presumably there are random problems that take exponential time when solved by either backtracking or by the pure literal rule, but our current results are not strong enough to show this. Those random sets where we show that the pure literal rule requires exponential average time can be solved by backtracking in polynomial average time [5, 7].

2. Random model

We have v variables, t clauses, and with probability p a literal is in a clause. A random clause is formed by including each literal with probability p. A predicate consists of t independently selected random clauses. Since there are $2v$ literals altogether, and since each literal is included with probability p, the expected clause size is equal to $2vp$. One should note that the expected clause size is linear in v whenever p is fixed; the expected clause size is fixed whenever p varies inversely with v, i.e. $p = O(v^{-1})$.

The average time for the pure literal rule is given by the following recurrence equation [3]:

$$A(t,v) = tv + (1-p)^{2t}A(t,v-1) + 2\sum_{i\geq 1}\binom{t}{i}p^i(1-p)^{t-i}A(t-i,v-1),$$
$$A(t,0) = A(0,t) = 0,$$

where tv is the time required for one step of the algorithm. The summation expression accounts for all possibilities of eliminating one variable leaving two subproblems each of size $t-i$, $1 \leq i \leq t$. The $(1-p)^{2t}A(t,v-1)$ term accounts for the case where the selected variable does not appear in the predicate.

3. Analysis

We consider a lower limit on $A(t,v)$ by dropping the $(1-p)^{2t}A(t,v-1)$ term. We obtain

$$L(t,v) = tv + 2\sum_{i\geq 1}\binom{t}{i}p^i(1-p)^{t-i}L(t-i,v-1).$$

$L(t,v)$ is a lower limit on $A(t,v)$ since $(1-p)^{2t}A(t,v-1)$ is positive. Define the exponential generating function

$$H_v(z) = \sum_{t\geq 0} L(t,v)\frac{z^t}{t!}. \tag{2.1}$$

Multiplying by $z^t/t!$ and summing over t gives

$$\sum_{t\geq 0} L(t,v)\frac{z^t}{t!} = \sum_{t\geq 0}\frac{vz^t}{(t-1)!} + 2\sum_{t\geq 0}\sum_{i\geq 1}\frac{p^i z^i}{i!}\frac{(1-p)^{t-i}L(t-i,v-1)z^{t-i}}{(t-i)!}. \tag{2.2}$$

Using eq. (2.1) on eq. (2.2) gives

$$H_v(z) = vze^z - 2H_{v-1}([1-p]z) + 2e^{pz}H_{v-1}([1-p]z). \tag{2.3}$$

The right most term in eq. (2.3) comes from the right most term in eq. (2.2) except the sum over i is starting at zero instead of one. The middle term in the right side accounts for the fact that eq. (2.2) does not have an $i = 0$ term.

Iterating with $H_0 = 0$, we get

$$H_v(z) = \sum_{0 \le i \le v-1} 2^i (v-i)(1-p)^i z e^{(1-p)^i z} \prod_{0 \le j \le i-1} (e^{p(1-p)^j z} - 1).$$

Expanding the exponentials and multiplying out [4], we find

$$e^{(1-p)^i z} \prod_{0 \le j \le i-1} (e^{p(1-p)^j z} - 1) = \sum_{n \ge 1} \Bigg(\sum_{\substack{k_i \ge 0 \\ k_0, \ldots, k_{i-1} \ge 1 \\ k_0 + \cdots + k_i = n}} \frac{p^{k_0 + \cdots + k_{i-1}} (1-p)^{0k_0 + 1k_1 + \cdots + ik_i}}{k_0!\, k_1! \ldots k_i!} \Bigg) z^n.$$

Thus

$$H(z) = vze^z + \sum_{n \ge 1} \Bigg(\sum_{1 \le i \le v-1} b^i (v-i) \sum_{\substack{k_i \ge 0 \\ k_0, \ldots, k_{i-1} \ge 1 \\ k_0 + \cdots + k_i = n}} \frac{(n+1)!}{k_0!\, k_1! \ldots k_i!} p^{k_0 + \cdots + k_{i-1}} (1-p)^{k_1 + \cdots + ik_i} \Bigg) \frac{z^{n+1}}{(n+1)!},$$

where $b = 2(1-p)$. Note that $L(t, v)$ is equal to the coefficient of $z^t/t!$ in $H(z)$. So

$$L(t,v) = tv + \sum_{1 \le i \le v-1} b^i (v-i) \sum_{\substack{k_i \ge 0 \\ k_0, \ldots, k_{i-1} \ge 1 \\ k_0 + \cdots + k_i = t-1}} \frac{p^{k_0 + \cdots + k_{i-1}} (1-p)^{k_1 + \cdots + ik_i}}{k_0!\, k_1! \ldots k_i!} t!.$$

Simplifying the inner sum, using inclusion-exclusion and the multinomial theorem [4], we find that

$$\begin{aligned} L(t,v) = tv + t \sum_{1 \le i \le v-1} b^i (v-i) \Bigg[1 &- \sum_{0 \le r_0 < i} [1 - p(1-p)^{r_0}]^{t-1} \\ &+ \sum_{0 \le r_0 < r_1 < i} [1 - p(1-p)^{r_0} - p(1-p)^{r_1}]^{t-1} - \cdots \Bigg]. \end{aligned} \tag{2.4}$$

If one approximates the expression in brackets with the leading terms, then the error is of the same sign and no larger in absolute value than the first term not included, because each term corrects for items that were over counted in the previous term.

4. Approximations

In this section we derive a lower limit on the size of the summation expression derived for $L(t, v)$. We are interested in the values of p and t (for large values of v) that will lead to an exponential lower limit on the size of $L(t, v)$.

Taking the first two terms in eq. (2.4) and interchanging the order of summation gives

$$\begin{aligned}
L(t,v) &\geq \sum_{0\leq i\leq v-1} b^i(v-i)\left[1-\sum_{0\leq r<i}\{1-p(1-p)^r\}^{t-1}\right] \\
&= \sum_{0\leq i<v}(v-i)b^i - \sum_{0\leq r<v-1}\left\{1-p(1-p)^r\right\}^{t-1}\sum_{r<i<v}(v-i)b^i \\
&= \frac{b^{v+1}}{(b-1)^2}-\frac{v}{b-1}-\frac{b}{(b-1)^2} \\
&\quad -\sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1}\left[\frac{b^{v+1}}{(b-1)^2}-\left\{(b-1)v-(b-1)r+1\right\}\frac{b^{r+1}}{(b-1)^2}\right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{b^{v+1}}{(b-1)^2}-\frac{v}{b-1}-\frac{b}{(b-1)^2}-\frac{b^{v+1}}{(b-1)^2}\sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1} \\
&\quad +\sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1}\left\{\frac{v-r}{b-1}+\frac{1}{(b-1)^2}\right\}b^{r+1}
\end{aligned}$$

The total of the terms multiplied by b^{r+1} is positive, so

$$L(t,v)\geq \frac{b^{v+1}}{(b-1)^2}-\frac{v}{b-1}-\frac{b}{(b-1)^2}-\frac{b^{v+1}}{(b-1)^2}\sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1}$$

To decide whether this is exponential for large v we look at the terms multipled by b^{v+1}. Using $S = \sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1}$, to make the lower limit large, we need

$$e^{av} < \frac{b^{v+1}}{(b-1)^2}(1-S)$$

for some fixed a. This will be true when $b>1$ unless $S\approx 1$ or $b\approx 1$. Assuming that $p<1/2$, we have exponential time whenever $S<1$. Now

$$\begin{aligned}
S = \sum_{0\leq r\leq v-2}\left\{1-p(1-p)^r\right\}^{t-1} &\leq v\left\{1-p(1-p)^{v-2}\right\}^{t-1} \\
&= ve^{(t-1)\ln[1-p(1-p)^{v-2}]} \\
&\leq ve^{-(t-1)p(1-p)^{v-2}}.
\end{aligned}$$

So Goldberg's version of the pure literal rule takes exponential time whenever

$$ve^{-(t-1)p(1-p)^{v-2}} < 1,$$

or

$$t > 1+\frac{\ln v}{p(1-p)^{v-2}}.$$

The value of p that gives the smallest lower bound for t can be obtained by setting, $\partial t/\partial p$ to zero, i.e.

$$\frac{\ln v}{p(1-p)^{v-2}}\left[\frac{-1}{p}+\frac{v-2}{1-p}\right]=0, \quad \text{or} \quad p=\frac{1}{v-1},$$

which gives

$$t > 1+\frac{(v-1)\ln v}{[1-1/(v-1)]^{v-2}} = \Omega(v\ln v).$$

4. Conclusion

We have established a large region where the pure literal rule requires exponential average time. This is the first nontrivial lower bound on the pure literal rule algorithm. The typical problem in this region has an exponentially small number of solutions. This is the region where backtracking algorithms (which verify unsatisfiability) are known to be effective [7].

The remaining problem on the analysis of the average time of this algorithm is to find upper and lower bounds that are closer together. It is particularly important to determine whether there is a region where both the pure literal rule algorithm and backtracking take exponential average time. The approach used to find our lower bound shows promise of giving both better upper bounds and better lower bounds.

Acknowledgement.

We thank Professor Cynthia Brown for the benefit of her earlier unpublished work on this problem. We thank Mr. You-Fang Pan for carefully checking the results.

References.

1. M. Davis, G. Logemann, and D. Loveland, *A machine program for theorem proving*, Comm. ACM **5** (1962) pp. 394–397.
2. Allen Goldberg, *Average Case Complexity of the Satisfiability Problem*, Proc. of the Fourth Workshop on Automated Deduction (1979) pp. 1–6.
3. Allen Goldberg, Paul Walton Purdom, Jr., and Cynthia A. Brown, *Average Time Analysis of Simplified Davis-Putnam Procedures*, Information Processing Letters **15** (1982) pp. 72–75. See **16** (1983) p. 213 for corrections to the typesetting errors of the original paper.
4. Donald E. Knuth, *The Art of Computer Programming, Vol. 1, Fundamental Algorithms*, Addison-Wesley, Reading, second edition (1973), p88, eq. (9).
5. Paul Walton Purdom, Jr. and Cynthia A. Brown, *Polynomial Average-Time Satisfiability Problems*, Indiana University Computer Science Report No. 118, Bloomington Ind. (1981).
6. Paul Walton Purdom, Jr. and Cynthia A. Brown, *The Pure Literal Rule and Polynomial Average Time*, SIAM J. Comp. **14** (Nov., 1985).
7. Paul Walton Purdom, Jr., *Search Rearrangement and Polynomial Average Time*, Artificial Intelligence **21** (1983) pp. 117–133.