An Average Time Analysis of Backtracking

by

Cynthia A. Brown

and

Paul Walton Purdom, Jr.

Computer Science Department
Indiana University
Bloomington, Indiana 47405

TECHNICAL REPORT NO. 86

AN AVERAGE TIME ANALYSIS OF BACKTRACKING

CYNTHIA A. BROWN
PAUL WALTON PURDOM, JR.

NOVEMBER, 1979

Research reported herein was supported in part by the National Science Foundation under grant number MCS 79 06110

An Average Time Analysis of Backtracking

Cynthia A. Brown

and

Paul Walton Purdam, Jr.

Abstract. Formulas are given for the expected number of nodes in the backtrack tree that is generated while searching for all the solutions of a random predicate. The most general formulas apply to selection from any set of predicates that obeys the following conditions. Each predicate is the conjunction of t terms selected from a set of terms T. For any subset $T' \leq T$, the probability that the predicate contains only terms from T' depends only on the size of T'. The set T must remain unchanged if each variable x_i is replaced by $p_i(x_i)$, where p_i is a permutation function. The time needed to evaluate the general formulas is proportional to v, the number of variables in the predicate. More detailed consideration is given to predicates whose terms are random disjunctive clauses with s literals, $t = v^{\alpha}$ for some $1 < \alpha < s$, and the random selections are done with repetition. For this case the expected number of nodes is

$$\left[4\left(\frac{2\pi}{s(s-1)}\right)^{\frac{1}{2}}\left(\frac{2\ell n\ 2}{2}\right)^{\frac{-s+2}{2(s-1)}} v^{\frac{s-\alpha}{2(s-1)}} + O\left(v^{-\frac{s-\alpha}{2(s-1)}} + v^{\frac{s-\alpha}{2(s-1)}+\frac{s(1-\alpha)}{s-1}}\right)\right]$$

$$\exp\left[(s-1)\left(\frac{2\ell n\ 2}{s}\right)^{\frac{s}{s-1}} v^{\alpha+\frac{s}{s-1}(1-\alpha)} + O\left(v^{\alpha+\frac{2s}{s-1}(1-\alpha)}\right)\right] .$$

Thus the average time for backtracking on this model is exponential with a sublinear exponent. More terms for the coefficient and exponent are given in the text.

Research reported herein was supported in part by the National Science Foundation grant # MCS79 06110.

1. Introduction

Many problems can be regarded as a search for all the solutions to an equation of the form $P(x_1, \ldots, x_v) = \text{true}$, where P is a v-ary predicate and each x_i has a finite set of possible values. The most straightforward way to solve such a problem is by enumerating and testing each combination of possible values of the variables. If each variable has i values, there are i^v possible solutions, so the exponential time required for complete enumeration makes this method impractical for all but the smallest problems.

Suppose that, in addition to P, there are intermediate predicates $\{P_k(x_1, \ldots, x_k)\}_{1 \le k \le v}$, where $P = P_v$, such that if $P_{k-1}(x_1, \ldots, x_{k-1})$ is false, then $P_k(x_1, \ldots, x_k)$ is false for all values of x_k. Then the technique of backtracking can be used to try to reduce the size of the space to be searched. The basic backtracking algorithm can be stated as follows:

1. [Initialize] Set $k \leftarrow 0$.
2. [Test] If $P_k(x_1, \ldots, x_k)$ is false, go to 6.
3. [New Level] Set $k \leftarrow k + 1$.
4. [Solution?] If $k > v$, then $x_1, \ldots, x_v$ is a solution. Go to 7.
5. [First value] Set $x_k \leftarrow$ the first value of x_k and go to 2.
6. [Next value] If x_k has more values, set $x_k \leftarrow$ the next value for x_k and go to 2.
7. [Backtrack] Set $k \leftarrow k - 1$. If $k > 0$, go to 6; otherwise stop.

The values that are tested can be represented by a tree, as shown

in Figure 1. Knuth [1] gives a more complete introduction to backtracking.

If the intermediate predicates can be evaluated quickly and are often false for small values of k, then backtracking takes considerably less time than complete enumeration, but if the intermediate predicates are ineffective, backtracking can take considerably longer. Although there are effective intermediate predicates for many problems, no general theory has been developed for finding them, and the dependence of the running time on the intermediate predicates as well as on the original predicate makes it difficult to do realistic analyses. In particular, a naive worst case analysis, where each intermediate predicate for $k < v$ is true, is uninteresting.

The first problem in studying backtracking is to choose a model domain of problems that are both representative and amenable to analysis. Since there is no consensus on what constitutes a typical backtracking problem, we avoid introducing arbitrary assumptions into the analysis as long as possible. Our first requirement is that the sets of predicates we consider have natural intermediate predicates. Each predicate P is the conjunction of t terms. The corresponding k-th intermediate predicate is the conjunction of those terms from P that contain only variables x_1 through x_k. Some of these sets of predicates contain NP complete problems and have $\Theta(2^v)$ worst case solution time when using backtracking. We obtain quite general formulas for the average solution time for these predicates. The formulas can be evaluated in time $O(v)$.

As a concrete illustration for our formulas, and as a model for

more detailed investigation, we use the problem of finding all solutions of conjunctive normal form formulas. These sets of formulas fit our general model, and their natural intermediate predicates have a simple form that lends itself to analysis. They contain NP complete problems. Moreover, the trees generated by these formulas have a shape typical of those encountered by the authors in our own experience with backtracking. Thus, these sets of formulas are a good model for analysis.

It is difficult to grasp the behavior of our general formulas as t and v become large. Therefore, for one type of random conjunctive normal form predicate, we derive asymptotic results, using $t = v^{\alpha}$ for some α. These results show that for such problems the average solution time using backtracking is exponential in v to a power that is less than one. Comparing this with the time exponential in v required for exhaustive search, it is evident that backtracking saves considerable time for nearly all problems in the class.

It is interesting to compare our results with those of Goldberg [2]. He analyzes the average time of the Davis-Putnam procedure [3]. In his analysis the procedure is simplified so that it is quite similar to backtracking. One important difference remains: if each continuation of a node eventually leads to a solution, that node is not explored further. This pruning of entire subtrees can lead to a huge savings if a large proportion of the predicates in the model have many solutions. His model has this property and gives a polynomial average time; the models we investigate asymptotically do not have this property.

We foresee two main uses for our results. To obtain an accurate estimate for the running time of a given backtrack program, the method of Knuth [1] as modified by Purdom [4] should be used. But to decide whether it will be useful to attack a problem by backtracking, before investing the effort in writing a program, the formulas in this paper provide a rough guide. The second use for these results will be in a theoretical comparison of ordinary backtracking to various modifications of backtracking [5, 6]. The analyses of the modified algorithms remain to be done.

2. Notation and Description of Model.

In our model each predicate P is the conjunction of t terms selected randomly from a set T of possible terms. Intermediate predicate P_k is the conjunction of the terms of P that use only variables $x_1, \ldots, x_k$. The random process for selecting terms must be such that the probability that P contains only terms from the set T_1, where $T_1 \subseteq T$, is proportional to $Q(|T_1|,t)$ for some function Q. (The generalization to weighted sets is straightforward.) Two important cases are:

$$Q(|T_1|,t) = \begin{cases} |T_1|^t & \text{(selection with replacement)} \quad (1) \\ \binom{|T_1|}{t} & \text{(selection without replacement)} \quad (2) . \end{cases}$$

Here Q is the number of ways the terms of P can be chosen; in all cases the probability is $Q(|T_1|,t) / Q(|T|,t)$.

Let d_i be the number of possible values for variable x_i. The set of terms T must be invariant under any operation that replaces

each variable x_i with $p_i(x_i)$, where p_i is a permutation. (For binary variables each p_i is either the not or identity function.) Let $F(k)$ be the number of terms in T that use only variables $x_1, \ldots, x_k$ and that are false when those variables have each been assigned some value. The invariant condition on T implies that the same number of terms are false for any set of values of the variables $x_1, \ldots, x_k$.

Let E be the total number of terms in T. If T consists of disjunctive clauses, where each clause contains s literals randomly selected from the v variables and their negation, then

$$F_s(k) = \begin{cases} k^s, & k \geq 0 \\ \binom{k}{s}, & k \geq 0 \end{cases}, \quad E_s = \begin{cases} 2^s v^s & \text{(selection with replacement)} \quad (3) \\ 2^s \binom{v}{s} & \text{(selection without replacement)} \quad (4) \end{cases}$$

with $F_s(-1) = 0$ in both cases.

Figure 1 shows a set of predicates over two variables with two terms, where the terms have been selected without replacement. The terms are clauses with one literal per clause. The backtrack tree for each predicate is also shown.

3. Tree size.

What is the expected number of nodes in a backtrack tree? Consider the tree in which each node on level i has degree d_{i+1} (the root is level 0). The node corresponding to $x_1 = y_1$, $x_2 = y_2$, $\ldots$, $x_k = y_k$ ($x_{k+1}, \ldots, x_v$ not set) is reached by exactly those predicates that have no terms that are false for those particular values

of the variables. There are $Q(E - F(k-1),t)$ such predicates. Altogether level k contains $\prod_{1\le i\le k} d_i$ nodes. (If $d_i = d$ for all i, then level k has d^k nodes.) The uniformity conditions on the set of predicates imply that the total number of nodes in all the backtrack trees is the product of the number of predicates and the number of nodes, summed over k. Dividing this by $Q(E,t)$, the number of backtrack trees, gives the expected number of nodes in a tree:

$$A(v,t) = \sum_{0\le k\le v} \left(\prod_{1\le i\le k} d_i \right) Q(E - F(k-1),t) / Q(E,t) \quad . \tag{5}$$

The expected number of solutions is

$$S(v,t) = \left(\prod_{1\le i\le v} d_i \right) Q(E - F(v),t) / Q(E,t) \quad . \tag{6}$$

Formulas 5 and 6 apply to any method of selecting predicates that obeys the restrictions of section 2. Formulas for particular cases are obtained by using appropriate versions of Q and F. The following examples are for terms consisting of disjunctive clauses with s literals per term.

$$A_s(v,t) = \begin{cases} 1 + \sum_{1 \le k \le v} 2^k \left(1 - \left(\frac{k-1}{2v}\right)^s\right)^t & \text{(terms and literals selected with replacement),} \quad (7) \\ 1 + \sum_{1 \le k \le v} 2^k \left(\binom{2^s v^s}{t} - (k-1)^s\right) \Big/ \binom{2^s v^s}{t} & \text{(terms selected without replacement, literals with),} \quad (8) \\ 1 + \sum_{1 \le k \le v} 2^k \left(1 - 2^{-s}\binom{k-1}{s} \Big/ \binom{v}{s}\right)^t & \text{(terms selected with replacement, literals without), and} \quad (9) \\ 1 + \sum_{1 \le k \le v} 2^k \left(\binom{2^s\binom{v}{s} - \binom{k-1}{s}}{t}\right) \Big/ \binom{2^s\binom{v}{s}}{t} & \text{(terms and literals selected without replacement).} \quad (10) \end{cases}$$

4. Exactly v Variables.

As illustrated in Figure 1, processes consistent with the assumptions of the previous section may generate some predicates with less than v variables. To study predicates with exactly v variables we replace the requirement that the set T be invariant under permutations of the values of the variables with a more restrictive assumption: the number $F(k)$ of terms that are false when any k variables are set is independent of which variables are set and of the values assigned to the variables. We also require that the number of terms that use no more than k particular variables be $E(k)$, independent of which k variables are considered. Under these assumptions the number of predicates that reach a particular node on level k and that do not use variable x_j is

$$Q(E(v-1) - F(k-2),t) \quad \text{for} \quad j \leq k-1 \ , \text{ and} \tag{11}$$

$$Q(E(v-1) - F(k-1),t) \quad \text{for} \quad j \geq k \ .$$

The number of predicates that reach a particular node on level k and that do not use j of the variables, where i of the j variables have indices less than k, is

$$\binom{k-1}{i}\binom{v-k+1}{j-1} Q(E(v-j) - F(k-i-1),t) \ , \tag{12}$$

where the binomials account for the number of ways the i variables less than k and the $j-i$ variables greater than or equal to k can be selected. Using the principle of inclusion and exclusion, the number of predicates that use all v variables and reach a particular node on level k is (for $k \geq 1$)

$$\sum_{i,j} (-1)^{v-j} \binom{k-1}{i}\binom{v-k+1}{j-i} Q(E(v-j) - F(k-i-1),t)$$
$$= \sum_{i,j} (-1)^{j} \binom{k-1}{i}\binom{v-k+1}{j-i} Q(E(j) - F(i),t) \ . \tag{13}$$

Multiplying by $\prod_{1\leq i\leq k} d_i$, summing over k, and dividing by the number of predicates gives the average number of nodes for the backtrack trees for predicates that use all v variables:

$$A(v,t) = 1 + \frac{\sum_{1\leq k\leq v} \left(\prod_{1\leq i\leq k} d_i\right) \sum_{i,j} (-1)^{j}\binom{k-1}{i}\binom{v-k+1}{j-i} Q(E(j) - F(i),t)}{\sum_{j} (-1)^{j}\binom{v}{j} Q(E(j),t)} \ . \tag{14}$$

5. Asymptotic Results.

For the asymptotic analysis we consider formulas in conjunctive normal form, where each clause has s literals and t, the number of

terms, is v^{α} . We require that both literals and terms be selected randomly with replacement. The expected tree size is therefore given by equation 7. We compute an asymptotic expression for the number of nodes in the backtrack tree for fixed α and s as v becomes large. Cook's construction in his NP completeness paper [7] produces predicates in conjunctive normal form, where the number of terms increases as $v^{3/2}$. The number of literals per term also increases with v , but Cook's predicates can easily be converted to a form with three literals per term. The set of predicates we consider in this section is therefore NP complete. This does not, of course, necessarily imply that the average time to solve a problem in the set will be large: the average time for any NP complete set of problems can be made arbitrarily low by adding enough easy problems to the set. The set we analyze is interesting because it is natural and because it contains hard problems.

As we will show, the summands in equation 7 are approximately Gaussian. We asymptotically sum the series by finding the position of the peak, expanding the deviation from a Gaussian in a power series, and summing the power series times the Gaussian using the Euler summation formula [8]. The main steps of this procedure are described in the remainder of this section.

Using $t = v^{\alpha}$, formula 7 can be rewritten as

$$A_s(v,v^{\alpha}) = 1 + \sum_{0<j\le v} \exp(j\ \ell n\ 2 + v^{\alpha}\ \ell n(1 - (\tfrac{j-1}{2v})^s)) . \qquad (15)$$

Let k be the value of j that maximizes the summand, and let $x = (k-1)/2v$. The value of x can be found by setting the derivative of the summand to zero (if the maximum is not at an endpoint), giving

$$x^{s-1} = \frac{2v^{1-\alpha}}{s}(1 - x^s)\,\ell n\, 2 \ . \tag{16}$$

Using either successive approximations or power series methods [9] on (16) gives

$$x = \left(\frac{2\,\ell n\, 2}{s}\right)^{\frac{1}{s-1}} v^{\frac{1-\alpha}{s-1}} \sum_{j\geq 0} (s-1)^j f_j(s) \left(\frac{2\,\ell n\, 2}{s}\right)^{\frac{js}{s-1}} v^{\frac{js(1-\alpha)}{s-1}} \ . \tag{17}$$

The coefficients are given by the relations

$$f_0(s) = 1 \ , \quad g_0(s) = 1 \ , \quad g_1(s) = -1 \ ,$$

$$f_k(s) = \sum_{1\leq j\leq k} \left[\frac{sj}{(s-1)k} - 1\right] g_j(s)\, f_{k-j}(s)(s-1)^j \ , \text{ and} \tag{18}$$

$$g_k(s) = \sum_{1\leq j\leq k-1} \left[\frac{(s+1)j}{k-1} - 1\right] f_j(s)\, g_{k-j}(s)(s-1)^{-j} \ .$$

The $g_j(s)$ are coefficients in a power series expansion for $1 - x^s$. Values through $f_{10}(s)$ are given in Table 1. The power series converges only for $\alpha > 1$. (For $\alpha < 1$, we have $k > v$.) Only the first few values of $f_j(s)$ are needed unless α is near one.

The value of the maximum term, which we will need later, is given by

$$\begin{aligned} &\exp\left(2vx\,\ell n\, 2 + v^{\alpha}\,\ell n\,(1 - x^s)\right) \\ &= \exp\left(\sum_{j\geq 1} (s-1)^{2-j} e_j(s) \left(\frac{2\,\ell n\, 2}{s}\right)^{\frac{js}{s-1}} v^{\frac{js}{s-1}(1-\alpha)+\alpha}\right) \ , \end{aligned} \tag{19}$$

where

$$(s-1)^{2-j}\, e_j(s) = s\, f_{j-1}(s) - \sum_{1\le k\le j} \frac{1}{k}\, h_{j-k,k}(s) ,$$

$$h_{0k}(s) = 1 \text{ , and} \tag{20}$$

$$h_{ik}(s) = -\sum_{1\le j\le i} \left(\frac{(k+1)j}{i} - 1\right) g_{j+1}(s)\, h_{i-j,k}(s) .$$

The h_{ik} are coefficients in the power series expansion of x^{ks}. Values of $e_j(s)$ through $j = 10$ are given in Table 2. Tables 1 and 2 were calculated using REDUCE programs. The initial entries were checked against previous hand calculations.

Replacing the j in equation (15) by $j+k$ and expanding the natural log in a power series gives

$$\begin{aligned} & A_s(v, v^\alpha) \\ &= 1 + \sum_{-k\le j\le v-k} \exp\left[(j+k)\,\ell n\, 2 + v^\alpha\, \ell n\left(1 - \left(\frac{j+k-1}{2v}\right)^s\right)\right] \\ &= 1 + \sum_{-k\le j\le v-k} 2\, \exp\left[(j+2vx)\,\ell n\, 2 - v^\alpha \sum_n \left(\sum_{i\ge 1} \frac{1}{i}\binom{is}{n} x^{is-n}\right)\left(\frac{j}{2v}\right)^n\right] . \end{aligned} \tag{21}$$

The factor independent of j is

$$\begin{aligned} & 2\, \exp\left(2\, vx\, \ell n\, 2 - v^\alpha \sum_{i\ge 1} \frac{1}{i} x^{is}\right) \\ &= 2\, \exp\left(2\, vx\, \ell n\, 2 + v^\alpha\, \ell n(1 - x^s)\right) . \end{aligned} \tag{22}$$

This is the value of the maximum term; it can be moved outside the sum over j. The terms in the exponent that are proportional to j are

$$\ell n\, 2 - v^{\alpha} \sum_{i\geq 1} \frac{s x^{is-1}}{2v}$$

$$= \ell n\, 2 - \frac{s v^{\alpha}}{2v} \frac{x^{s-1}}{1-x^{s}} \quad . \tag{23}$$

This is zero by equation 16. Using 22 and 23 and separating the j^2 term from the rest gives

$$A_s(v,v^{\alpha}) = 1 + 2 \exp(2vx\, \ell n\, 2 + v^{\alpha} \ell n\,(1 - x^{s})) \sum_{-k<j\leq v-k} \exp(-aj^2) \exp\Big(\sum_{n\geq 3} t_n j^n\Big) \quad , \tag{24}$$

where

$$a = v^{\alpha} \sum_{i\geq 1} s \frac{is-1}{8v^2} x^{is-2} \quad \text{and}$$

$$t_n = -v^{\alpha} \sum_{i\geq 1} \frac{1}{i} \binom{is}{n} x^{is-n} \left(\frac{1}{2v}\right)^{n} \quad \text{for} \quad n \geq 3 \ .$$

Now

$$\exp\left(\sum_{n\geq 3} t_n j^n\right) = 1 + \sum_{i\geq 3} b_i j^i \quad , \tag{25}$$

where

$$b_i = \sum_{R} \prod_{n\geq 3} \frac{t_n^{u_n}}{u_n!} \quad , \qquad R = \Big\{u_3 \geq 0 \, , u_4 \geq 0 \, , \ldots \Big| \sum_{n\geq 3} n u_n = i\Big\} \ . \tag{26}$$

For example, $b_3 = t_3$, $b_4 = t_4$, $b_5 = t_5$, and $b_6 = t_6 + \frac{t_3^2}{2}$.

This reduces the problem to evaluating sums of the form

$$\sum_{-k\leq j\leq v-k} b_n j^n \exp(-aj^2) \ .$$

Using $f_n(y) = b_n y^n \exp(-ay^2)$ in the Euler summation formula [8] gives

$$\begin{aligned}
&\sum_{-k\le j\le v-k} b_n j^n \exp(-aj^2) \\
&= \int_{-k+1}^{v-k+1} f_n(y)\,dy + \sum_{1\le p\le m} \frac{B_p}{p!}\left(f_n^{(p-1)}(v-k+1) - f_n^{(p-1)}(-k+1)\right) \\
&\quad + \frac{(-1)^{m+1}}{m!} \int_{-k+1}^{v-k+1} B_m(\{y\})\, f_n^{(m)}(y)\,dy \ ,
\end{aligned} \tag{27}$$

where parenthesized superscripts indicate derivatives, the B_i are Bernoulli numbers and polynomials, and $\{\ \}$ is the sawtooth function. The error term is $O(v\, f_n^{(m)}(z))$, where z is the value of y that maximizes $f_n^{(m)}(y)$. Now $f_n^{(m)}(y)$ has the form $a^{(m-n)/2} R_{m+n}(a^{\frac{1}{2}} y)$, where $R_n(y)$ is e^{-y^2} times an n-th degree polynomial. The coefficients of the polynomial are numbers and do not change with a, so the maximum of $R_{m+n}(a^{\frac{1}{2}} y)$ is independent of a. Therefore the error term is $O\left(a^{\frac{m-n}{2}} v\right)$. Since $a = O(v^{\alpha-2} x^{s-2}) = O\left(v^{\frac{\alpha-s}{s-1}}\right)$, the error term is $O\left(v^{1+\frac{(m-n)(\alpha-s)}{2(s-1)}}\right)$. For $\alpha < s$ and m sufficiently large, the error term becomes small faster than any term we retain, so we will be able to neglect it if we can show that there is no problem in making m large.

To do this we show that the terms from equation 27 in the sum over p can be neglected. Consider $f_n^{(p-1)}(-k+1)$. Its asymptotic behavior

as v becomes large depends on its exponent, which is $a(-k+1)^2 = O\left(v^{\frac{\alpha-s}{(s-1)}} v^{2+2\frac{1-\alpha}{s-1}}\right) = O\left(v^{\frac{s-\alpha}{(s-1)}}\right)$. Therefore $f^{(p-1)}(-k+1)$ becomes exponentially small for $\alpha < s$. If $\alpha > 1$, then $f^{(p-1)}(v-k+1)$ also becomes exponentially small as v increases. Since the final answer is polynomial with fractional powers, the exponentially small terms in the sum over p can be neglected, so m can be made as large as desired.

This leaves $\int_{-k+1}^{v-k+1} f_n(y)\,dy$ to be evaluated. For $1 < \alpha < s$, this integral differs only by exponentially small terms from

$$\int_{-\infty}^{\infty} f_n(y)\,dy = \begin{cases} \dfrac{\pi^{\frac{1}{2}} n!}{(\frac{n}{2})!\, 2^n}\, a^{-\frac{n+1}{2}}\, b_n & \text{for } n \text{ even,} \\ 0 & \text{for } n \text{ odd.} \end{cases} \tag{28}$$

This gives, using equations 24 and 25,

$$A_s(v,v^{\alpha}) = 1 + 2\pi^{\frac{1}{2}} \left[a^{-\frac{1}{2}} + \sum_{\substack{i\geq 3 \\ i \text{ even}}} \frac{i!}{(\frac{i}{2})!\, 2^i}\, a^{-\frac{i+1}{2}}\, b_i \right] \tag{29}$$

$$\exp(2vx\,\ell n\,2 + v^{\alpha}\,\ell n(1-x^s)) \ .$$

Expanding the factor of equation 29 that is in square brackets in a power series gives, using equations 17, 24 (for a and t_n), and 25,

$$\sum_{\substack{p\geq 0 \\ q\geq 0}} \left(\frac{2\,\ell n\,2}{s}\right)^{\frac{(2q-2p-1)s+2}{2(s-1)}} w_{pq}\, v^{(\frac{1-2p}{2})(\frac{s-\alpha}{s-1})+\frac{qs(1-\alpha)}{s-1}} , \tag{30}$$

where the first few values for w_{pq} are given in Table 3. In the power series expansion w_{0q} is obtained from $a^{-\frac{1}{2}}$ and w_{pq} for $p \geq 1$ is obtained from all factors containing

$$a^{-\left(\frac{\sum_{r\geq 0} r u_r + 1}{2}\right)} \prod_n t_n^{u_n} , \tag{31}$$

where $u_1 = 0$, $u_2 = 0$, $\sum_{r\geq 0} (r-2)u_r = 2p$, and $u_r \geq 0$ for all r. For example, w_{1q} is obtained from t_4 and t_3^2, and w_{2q} is obtained from t_6, $t_5 t_3$, t_4^2, and t_3^4.

Thus we obtain the following formula:

$$\begin{aligned} & A_s(v, v^\alpha) \\ &= 1 + 2\pi^{\frac{1}{2}} \sum_{\substack{p\geq 0 \\ q\geq 0}} \left(\frac{2\ell n\, 2}{s}\right)^{\frac{(2q-2p-1)s+2}{2(s-1)}} w_{pq}\, v^{\left(\frac{1-2p}{2}\right)\left(\frac{s-\alpha}{s-1}\right)+\frac{qs(1-\alpha)}{s-1}} \\ & \qquad \exp\left[\sum_{j\geq 1} (s-1)^{2-j} e_j(s) \left(\frac{2\ell n\, 2}{s}\right)^{\frac{js}{s-1}} v^{\frac{js}{s-1}(1-\alpha)+\alpha}\right] . \end{aligned} \tag{32}$$

The leading term of this formula is

$$\begin{aligned} & \left[4\left(\frac{2\pi}{s(s-1)}\right)^{\frac{1}{2}} \left(\frac{2\ell n\, 2}{s}\right)^{\frac{-s+2}{2(s-1)}} v^{\frac{s-\alpha}{2(s-1)}} + O\left(v^{-\frac{s-\alpha}{2(s-1)}} + v^{\frac{s-\alpha}{2(s-1)}+\frac{s(1-\alpha)}{s-1}}\right)\right] \\ & \exp\left[(s-1)\left(\frac{2\ell n\, 2}{s}\right)^{\frac{s}{s-1}} v^{\alpha+\frac{s}{(s-1)}(1-\alpha)} + O\left(v^{\alpha+\frac{2s}{s-1}(1-\alpha)}\right)\right] . \end{aligned} \tag{33}$$

When $s = 3$ and $\alpha = \frac{3}{2}$ (a case for which the problem set is NP complete), this gives

$$A_3(v,v^{3/2}) = 1 + e^{0.6282482661\, v^{3/4}}[4.725676293\, v^{3/8} - 0.9010567850\, v^{-3/8} + 0.3756198156\, v^{-9/8} + O(v^{-15/8})] . \qquad (34)$$

In work of this type checking is important to avoid errors. Formulas 7-10 were checked for small values of v, t, and s by comparison with programs that generated all predicates in a class and counted the number of nodes in the backtrack trees. Special cases for formula 6 were checked in the same way. Formulas 32 and 34 were checked using numerical comparison with formula 7. In Formula 30, $s = 3$, 4, 5, and 6 were checked with α varying from 1 to s in 20 steps. For $s = 3$ and 4, the formula checked. For $s = 5$ and 6, overflow errors prevented setting v large enough (a few thousand) for a definitive test. The successful tests check for all errors in the values given in Table 3, except in the terms in w_{20} that are multiplied by $\binom{s}{5}$ or $\binom{s}{6}$.

6. Conclusions.

Formulas 5, 6, and 14 are general formulas that can be used to predict the average time required for backtracking over many classes of

randomly selected predicates. Formulas 5 and 6 can be evaluated exactly in time proportional to v, the number of variables. The largest term in Formula 5 determines the value to within a factor of $v^{\frac{1}{2}}$. For such exponential problems this accuracy is often adequate.

The asymptotic result for random conjunctive normal form predicates, given in equation 32, shows that backtracking can save substantial time over exhaustive search on the average. Although the average time for backtracking is exponential in v, the dependence of the exponent on v is sublinear. For random problems backtracking does best on problems with low direct interdependence (small s) and on problems with a lot of restrictions (large α).

It is important to consider whether studies of random conjunctive normal form predicates lead to valid conclusions about typical backtracking problems. Certainly in our experience the typical backtracking problem is not in conjunctive normal form. Nevertheless, the random conjunctive normal form predicates do have many properties that we regard as typical. The constraints are initially not very effective, but they become more so as one goes down the search tree, so that the number of nodes per level shows a rapid increase up to a rounded peak followed by a rapid decrease. There is some correlation between adjacent branches of the search tree, but it is not very significant. On the other hand, real problems often have solutions, while random conjunctive normal form predicates with enough terms to mimic what we consider to be a typical problem almost never have solutions. The existence of solutions, however, does not have much effect on the size of the search tree. Our

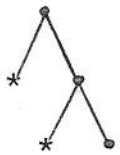
model, which is qualitatively correct on the important aspects of the problem, can be expected to give qualitatively correct results.

Goldberg [2] analyzed the average time for a variant of the Putnam-Davis procedure on a class of conjunctive normal form predicates. The time he obtained was polynomial, in contrast to the exponential time with sublinear exponent for backtracking. As was discussed in the introduction, Goldberg's variant of the Putnam-Davis procedure is similar to backtracking, but saves a great deal of time on predicates that have many solutions by stopping search at nodes where all descendants lead to solutions. While this shortcut can save a huge amount of time on predicates with many solutions, its effect on predicates with few solutions (such as the ones we considered in the most detail) is insignificant.

We have analyzed only the most straightforward backtracking algorithm. There are variations on backtracking [5, 6] that are careful about which variable to introduce at each step in the search. The investigations of Bitner and Reingold [5] as well as our own numerical studies with random conjunctive normal form predicates show that these variations can be much more efficient than traditional backtracking. All these backtracking methods maintain the advantage of treating the predicate as a black box: the algorithms are controlled only by the results of evaluating the predicate for selected values of the variables. We hope to analyze these methods in the future.

1. $P = \{x_1, \neg x_1\}$

2. $P = \{x_1, x_2\}$



3. $P = \{x_1, \neg x_2\}$

4. $P = \{\neg x_1, x_2\}$

5. $P = \{\neg x_1, \neg x_2\}$

6. $P = \{x_2, \neg x_2\}$

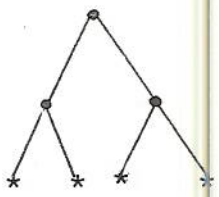


Figure 1

The backtrack trees for six simple predicates. The false branch is to the left. Each node where a term is false is marked with an asterisk.

$$f_0 = 1 \qquad f_1 = -1 \qquad f_2 = \frac{1}{2}(s+2)$$

$$f_3 = -\frac{1}{3}(s^2 + 4s + 3) \qquad f_4 = \frac{1}{24}(6s^3 + 37s^2 + 58s + 24)$$

$$f_5 = -\frac{1}{10}(2s^4 + 17s^3 + 42s^2 + 37s + 10)$$

$$f_6 = \frac{1}{720}(120s^5 + 1318s^4 + 4553s^3 + 6388s^2 + 3708s + 720)$$

$$f_7 = -\frac{1}{315}(45s^6 + 612s^5 + 2761s^4 + 5456s^3 + 5071s^2 + 2124s + 315)$$

$$f_8 = \frac{1}{4480}(560s^7 + 9148s^6 + 51540s^5 + 133769s^4 + 175804s^3 + 118236s^2 + 37904s + 4480)$$

$$f_9 = -\frac{1}{4536}(504s^8 + 9666s^7 + 65881s^6 + 214554s^5 + 371550s^4 + 354106s^3 + 182529s^2 + 46674s + 4536)$$

$$f_{10} = \frac{1}{3628800}(362880s^9 + 8026416s^8 + 64621692s^7 + 255644668s^6 + 557061609s^5 + 700870638s^4 + 512539012s^3 + 210852936s^2 + 44339040s + 3628800)$$

Table 1

Coefficients for Formula 17

$$e_1(s) = 1 \qquad e_2(s) = -\frac{1}{2} \qquad e_3(s) = \frac{1}{6}(s+2)$$

$$e_4(s) = -\frac{1}{12}(s^2+4s+3) \qquad e_5(s) = \frac{1}{120}(6s^3+37s^2+58s+24)$$

$$e_6(s) = -\frac{1}{60}(2s^4+17s^3+42s^2+37s+10)$$

$$e_7(s) = \frac{1}{5040}(120s^5+1318s^4+4553s^3+6388s^2+3708s+720)$$

$$e_8(s) = -\frac{1}{2520}(45s^6+612s^5+2761s^4+5456s^3+5071s^2+2124s+315)$$

$$e_9(s) = \frac{1}{40320}(560s^7+9148s^6+51540s^5+133769s^4+175804s^3+118236s^2+37904s+4480)$$

$$e_{10}(s) = -\frac{1}{45360}(504s^8+9666s^7+65881s^6+214554s^5+371550s^4+354706s^3+182529s^2+46674s+4536)$$

Table 2

Coefficients for Formula 19

$$w_{00} = 2\left(\frac{2}{s(s-1)}\right)^{1/2} \qquad w_{01} = -\left(\frac{s+1}{s-1}\right)\left(\frac{2}{s(s-1)}\right)^{1/2} \qquad w_{02} = \frac{3s^2+8s+3}{4(s-1)^2}\left(\frac{2}{2(s-1)}\right)^{1/2}$$

$$w_{10} = -\frac{4!}{2!\,2^3}\binom{s}{4}\left(\frac{2}{s(s-1)}\right)^{5/2} + \frac{6!}{3!\,2^6}\binom{s}{3}^2\left(\frac{2}{s(s-1)}\right)^{7/2}$$

$$w_{11} = \frac{4!}{2!\,2^3}\left[-\frac{1}{2}\binom{2s}{4} + \frac{6s+1}{s-1}\binom{s}{4}\right]\left(\frac{2}{s(s-1)}\right)^{5/2} +$$

$$+ \frac{6!}{3!\,2^6}\left[\binom{s}{3}\binom{2s}{3} - \frac{11s-5}{2(s-1)}\binom{s}{3}^3\right]\left(\frac{2}{s(s-1)}\right)^{7/2}$$

$$w_{20} = -\frac{6!}{3!\,2^5}\binom{s}{6}\left(\frac{2}{s(s-1)}\right)^{7/2} + \frac{8!}{4!\,2^7}\left[\binom{s}{4}^2 + 2\binom{s}{3}\binom{s}{5}\right]\left(\frac{2}{s(s-1)}\right)^{9/2} +$$

$$+ \frac{12!}{4!\,6!\,2^{11}}\binom{s}{3}^4\left(\frac{2}{s(s-1)}\right)^{13/2}$$

Table 3

Coefficients for Formula 32

References

1. D.E. Knuth, Estimating the efficiency of backtracking programs, Math. Comput., 29 (1975), pp. 121-136.

2. Allen Goldberg, Average case complexity of the satisfiability problem, Proceedings Fourth Workshop on Automated Deduction (1979), pp. 1-6.

3. Martin Davis and Hilary Putnam, A computing procedure for quantification theory, J. ACM, 7 (1960), pp. 201-215.

4. Paul W. Purdom, Tree size by partial backtracking, SIAM J. Comput., 7 (1978), pp. 481-491.

5. J.R. Bitner and E.M. Reingold, Backtrack programming techniques, Comm. ACM, 18 (1975), pp. 651-655.

6. Paul Purdom, Edward Robertson, and Cynthia Brown, Multi-level dynamic search arrangement, Indiana University Computer Science Dept., Technical Report No. 77 (1979).

7. Stephen A. Cook, The complexity of theorem-proving procedures, Proc. Third ACM Symp. on Theory of Computing (1971), pp. 151-158.

8. Donald E. Knuth, The art of computer programming, vol. 1 , Addison-Wesley, Reading, Massachusetts (1975), p. 110.

9. Donald E. Knuth, The art of computer programming, vol. 2 , Addison-Wesley, Reading, Massachusetts (1969), pp. 444-450.