Size-Depth Tradeoffs for Boolean Formulae

Maria Luisa Bonet
Department of Mathematics
Univ. of Pennsylvania, Philadelphia

Samuel R. Buss*
Department of Mathematics
Univ. of California, San Diego

July 3, 2002

Abstract

We present a simplified proof that Brent/Spira restructuring of Boolean formulas can be improved to allow a Boolean formula of size n to be transformed into an equivalent log depth formula of size $O(n^\alpha)$ for arbitrary $\alpha > 1$.
KEYWORDS: *theory of computation, Boolean formulas, formula complexity, size-depth tradeoff*

1 Introduction

A Boolean formula is constructed from variables $x, y, \ldots$ and from Boolean functions (also called 'gate types') such as AND ($\wedge$), OR ($\vee$), NOT ($\neg$), PARITY ($\oplus$), etc. Equivalently, a Boolean formula is a Boolean circuit with fanout one. A *basis* B is a finite set of Boolean gate types, and a B-formula is a formula using only gate types from B. When deriving asymptotic size bounds on Boolean formulas, we always work with a fixed basis B and consider only B-formulas.

It has been known for some time (Spira [4] and Brent [1]) that a Boolean formula of size n can be transformed into an equivalent $O(\log n)$ depth formula. Examination of the methods of Brent and Spira shows that this transformation can yield a log depth formula of size $O(n^\alpha)$ with $\alpha = 2.1964$. We present a simple proof that for any $\alpha > 1$ and arbitrarily close to 1, a

*Supported in part by NSF grant DMS-9205181

Boolean formula of size n has an equivalent $O(\log n)$ depth formula of size $O(n^\alpha)$. We prove this for formulas over the basis AND, OR, NOT and also for formulas over the basis PARITY, AND and 1. Our methods also for work for other bases, e.g., B_2, the set of all binary gate types.

This improvement to arbitrary $\alpha > 1$ has already been obtained by Bshouty-Cleve-Eberly [2]. The advantage of our proof is that it is much simpler. The reasons that our proof is simpler are (1) we deal only with Boolean formulas while Bshouty-Cleve-Eberly deal with the more general arithmetic case, and (2) we use a much simpler method of choosing breakpoints in formulas based on a construction of Brent. Bshouty et.al. use a more complicated extension of Brent's construction. Our method of choosing breakpoints could also be used in the algebraic case allowing simpler proofs of the essential results of Bshouty,et.al.; however, we do not present this here.

2 Near-linear size, log depth transformations

We shall identify Boolean formulas with rooted trees in which each node is labelled with a Boolean gate type and each leaf is labelled with a variable name. The *depth* of a formula or a tree is the maximum number of nodes (Boolean gates) of arity ≥ 2 on any branch of the tree. Note that unary gates, such as negations, do not count towards the depth. The *leafsize* of a formula is the number of occurrences of variables in the formula, which is also equal to the number of leaves in the tree. We use *log* and *ln* to denote logarithms base two and e, respectively.

For expository purposes, we begin by giving the proof of a well-known theorem of Spira. For this, we need the following lemma about choosing breakpoints in a tree (of fanin ≤ 2) that split the tree roughly into half.

Lemma 1 *(Lewis-Stearns-Hartmanis [3]) If T is a tree with all nodes having arity at most 2, and if the leafsize of T is m where $m \geq 2$, then there is a subtree S of T with leafsize s, where*

$$\lceil \tfrac{1}{3}m \rceil \leq s \leq \lfloor \tfrac{2}{3}m \rfloor.$$

We let B_2 be the set of all binary Boolean gate types. Note that any B_2-formula must be a binary tree.

Theorem 2 *(Spira [4]) Let C be a B_2-formula of leafsize m. Then there is an equivalent $\{\wedge, \vee, \neg\}$-formula C' such that,*

$$depth(C') \leq 2 \cdot \log_{3/2} m \approx 3.419 \log m$$

and such that

$$leafsize(C') \leq m^{\alpha}$$

where α satisfies $\frac{1+2^{\alpha}}{3^{\alpha}} \leq \frac{1}{2}$ ($\alpha \geq 2.1964$ suffices).

The proof of this theorem is by induction on the leafsize of C. The induction step uses Lemma 1 to find a subformula D of C having leafsize s satisfying $\lceil \frac{1}{3}m \rceil \leq s \leq \lfloor \frac{2}{3}m \rfloor$.

We define C_0 and C_1 to be B_2-formulas obtained from C by the following process: first replace the subformula D by 0 and 1, respectively. Now eliminate the constants 0 and 1 by collapsing gates that contain a constant as input (this removes at least one Boolean gate and might reduce the leafsize). For a given truth assignment to the variables of C, if D has value 0 then C has value equal to the value of C_0, and if D has value 1, then C has the same value as C_1. Therefore C is equivalent to

$$E \quad := \quad (C_0 \wedge \neg D) \vee (C_1 \wedge D).$$

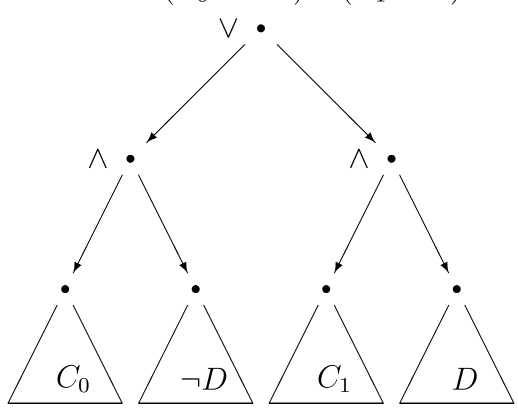

Figure 1

Now apply the induction hypothesis to C_0, C_1, D and $\neg D$ to get equivalent formulas C_0', C_1', D' and $(\neg D)'$ of logarithmic depth. Clearly C is equivalent to the formula

$$C' \quad := \quad (C_0' \wedge (\neg D)') \vee (C_1' \wedge D').$$

Also the leafsize of C' is equal to the sum of the leafsizes of C'_0, C'_1, D' and $(\neg D)'$. Its depth is two plus the maximum depth of these four formulas.

From this it straightforward to obtain the constants $2\log_{3/2} m$ and α: we leave this calculation to the reader, as we shall do a similar, but more complicated calculation below. □

It is possible to make an improvement to the constants in Theorem 2 if we assume that C is a $\{\wedge, \vee, \neg\}$-formula instead of a general B_2-formula. This improvement depends on the fact that only one occurrence of subformula D is picked as a breakpoint. Note that if D is a positively occurring subformula of C then C_0 tautologically implies C_1, and otherwise, if D is negatively occurring then C_1 tautologically implies C_0. In the first case, when C_0 tautologically implies C_1, we have that C is equivalent to both of the formulas (see Figure 2):

$$C_0 \vee (D \wedge C_1) \qquad \text{and} \qquad (C_0 \vee D) \wedge C_1.$$

In the second case, when C_1 tautologically implies C_0, we have that C is equivalent to both of the formulas:

$$C_1 \vee (\neg D \wedge C_0) \qquad \text{and} \qquad (C_1 \vee \neg D) \wedge C_0.$$

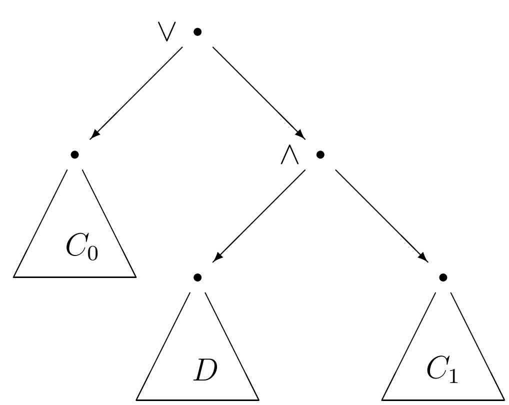

Figure 2

The point is that, unlike in the proof of Spira's theorem sketched above, it is unnecessary to include the subformula D twice in the formula E. This of course will improve the constant α and give a better bound on the leafsize of C'. However, even better (smaller) values for α can be obtained if we also change the choice of breakpoints so that instead of having D be approximately one half the size of C, we choose D to be some larger fraction of C. Intuitively, this will help because the larger piece (that is, D) will be used only once, whereas the smaller piece (that is, C_0 and C_1) will be used twice.

The new breakpoints will be based on the following simple lemma:

Lemma 3 *(Brent [1]) Let T be a tree with leafsize m, and $1 \leq s \leq m$. Then there is a subtree D such that D has leafsize $\geq s$ and such that its immediate subtrees have leafsize $< s$.*

Proof Any minimal subtree of T of leafsize $\geq s$ will suffice. □

Theorem 4 *(See also Bshouty-Cleve-Eberly [2]) Let C be a $\{\wedge, \vee, \neg\}$-formula of leafsize m. Then for all $k \geq 2$, there is an equivalent $\{\wedge, \vee, \neg\}$-formula C' such that*

$$\mathit{depth}(C') \leq (3k \ln 2) \cdot \log m \approx 2.07944 k \log m,$$

and such that

$$\mathit{leafsize}(C') \leq m^{\alpha},$$

where $\alpha = 1 + \frac{1}{1+\log(k-1)}$.

Proof By induction on the leafsize m. If $m = 1$, C computes either x_i or $\neg x_i$, and C already has the desired leafsize and depth.

Let us assume now that the theorem applies to leafsizes up to $m-1$, and prove the theorem for m. Brent's lemma provides a subformula D of leafsize $\geq \frac{k-1}{k}m$ and immediate subtrees D_L and D_R of leafsize $< \frac{k-1}{k}m$: let $*$ denote the gate type of D's root. Consider now the formulas C_0 and C_1 obtained from C by replacing the subformula D by the constants 0 and 1 respectively and then collapsing the gates that use them so that $\mathit{leafsize}(C_0), \mathit{leafsize}(C_1) \leq \frac{1}{k}m$. Now we use the induction hypothesis to obtain formulas C_0', C_1', D_L', and D_R' so that

$$\mathit{leafsize}(D_L'), \mathit{leafsize}(D_R') < \left(\frac{k-1}{k}m\right)^{\alpha}$$

$$\mathit{leafsize}(C'_0), \mathit{leafsize}(C'_1) \leq \left(\frac{m}{k}\right)^{\alpha}$$

and

$$\mathit{depth}(D'_L), \mathit{depth}(D'_R) < (3\ln 2)k \log\left(\tfrac{k-1}{k}m\right)$$

$$\mathit{depth}(C_0), \mathit{depth}(C_1) \leq (3\ln 2)k \log\left(\tfrac{m}{k}\right)$$

Depending on whether D occurs positively or negatively as a subformula of C, the formula C' is to be defined to be either $C'_0 \vee ((D'_L * D'_R) \wedge C'_1)$ or $C'_1 \vee (\neg(D'_L * D'_R) \wedge C'_0)$. In either case,

$$\begin{aligned}
\mathit{depth}(C') &= \max\{\mathit{depth}(D'_L)+3, \mathit{depth}(D'_R)+3, \mathit{depth}(C'_0)+2, \mathit{depth}(C'_1)+2\} \\
&< (3\ln 2)k\log(\tfrac{k-1}{k}m)+3 \\
&= (3\ln 2)k\log m + (3\ln 2)k\log(\tfrac{k-1}{k})+3 \\
&= (3\ln 2)k\log m + 3k\ln 2\log(\tfrac{k-1}{k})+3 \\
&= (3\ln 2)k\log m + 3k\ln(1-1/k)+3 \\
&< (3\ln 2)k\log m + 3(-1)+3 \\
&= (3\ln 2)k\log m
\end{aligned}$$

The last inequality holds because $\ln(1-1/k) < -1/k$.

For notational convenience, we now write $\|A\|$ for the leafsize of A. We can bound the leafsize of C' by:

$$\|C'\| \leq 2(m - \|D_L\| - \|D_R\|)^{\alpha} + \|D_L\|^{\alpha} + \|D_R\|^{\alpha}.$$

To study the worst case, let us set $b = \|D\| = \|D_L\| + \|D_R\|$. Thinking of b as a constant, we can bound $\|C'\|$ by

$$f(\|D_L\|) = 2(m-b)^{\alpha} + \|D_L\|^{\alpha} + (b - \|D_L\|)^{\alpha}.$$

The function f is concave up, so the above expression is maximized at the endpoints which are (1) $\|D_L\| = \frac{k-1}{k}m$ and $\|D_R\| = 0$ and (2) $\|D_L\| = 0$ and $\|D_R\| = \frac{k-1}{k}m$. In either case, $\|C'\|$ is bounded above by

$$2(m - \|D\|)^{\alpha} + \left(\tfrac{k-1}{k}m\right)^{\alpha} + \left(\|D\| - \tfrac{k-1}{k}m\right)^{\alpha}.$$

Again this is a concave up as a function of $\|D\|$, so the maximum values are at the endpoints (1) $\|D\| = m$ and (2) $\|D\| = \frac{k-1}{k}m$. In this case, the maximum is at $\|D\| = \frac{k-1}{k}m$. So the worst case happens when $\|C_0\|$ and $\|C_1\|$ are $\frac{m}{k}$, $\|D_L\| = \frac{k-1}{k}m$ and $\|D_R\| = 0$. Thus we have the bound

$$\mathit{leafsize}(C') \leq 2\left(\frac{m}{k}\right)^{\alpha} + \left(\frac{k-1}{k}m\right)^{\alpha}.$$

To finish the proof of Theorem 4 we must prove that for $\alpha = 1 + \frac{1}{\log(k-1)+1}$, we have $2(\frac{m}{k})^{\alpha} + (\frac{k-1}{k}m)^{\alpha} \leq m^{\alpha}$; this is of course equivalent to showing that $2(\frac{1}{k})^{\alpha} + (\frac{k-1}{k})^{\alpha} \leq 1$. It is easy to see that the lefthand side of the inequality is a decreasing function of α and to prove the inequality, it will suffice to let α_0 be the (unique) value, greater than 1, so that $2(\frac{1}{k})^{\alpha_0} + (\frac{k-1}{k})^{\alpha_0} = 1$ and prove that $\alpha_0 < 1 + \frac{1}{\log(k-1)+1}$. Multiplying the equation defining α_0 by k^{α_0}, we get that

$$k^{\alpha_0} - (k-1)^{\alpha_0} = 2. \tag{1}$$

Now it must be that that $\alpha_0 < 2$, since $k \geq 2$ and thus $k^2 - (k-1)^2 = 2k - 1 > 2$. Define $g_{\alpha_0}(k) = k^{\alpha_0}$. By the Mean Value Theorem, equation (1) implies that there exists x, $(k-1) < x < k$, such that

$$g'_{\alpha_0}(x) = \alpha_0 x^{\alpha_0 - 1} = 2.$$

Since g'_{α_0} is increasing, $g'_{\alpha_0}(k-1) = \alpha_0(k-1)^{\alpha_0-1} < 2$. Taking logarithms yields:

$$\begin{aligned} \log(\alpha_0(k-1)^{\alpha_0-1}) &< \log 2 = 1 \\ \log\alpha_0 + (\alpha_0 - 1)\log(k-1) &< 1 \end{aligned}$$

Since $\alpha_0 - 1 < \log\alpha_0$ for $1 < \alpha_0 < 2$, $(\alpha_0 - 1)(\log(k-1) + 1) < 1$. So,

$$\begin{aligned} \alpha_0 - 1 &< \frac{1}{\log(k-1)+1} \\ \alpha_0 &< 1 + \frac{1}{\log(k-1)+1} \end{aligned}$$

which completes the proof of Theorem 4. □

Theorem 5 *(See also Bshouty-Cleve-Eberly [2]) Let C be a $\{\oplus, \wedge, 1\}$-formula of leafsize m. Then for all $k \geq 2$, there is an equivalent $\{\oplus, \wedge, 1\}$-formula C' such that*

$$depth(C') \leq (3 \ln 2) \log m$$

and

$$leafsize(C') \leq m^{\alpha}$$

where $\alpha = 1 + \frac{1}{1+\log(k-1)}$.

Proof First notice that if D is a subtree of C, then C is equivalent to:

$$(D \wedge (C_0 \oplus C_1)) \oplus C_0.$$

This is because if D is 0 then C will have the same value as C_0, and if D is 1 then C will have the same value as C_1 which is equivalent to $(C_0 \oplus C_1) \oplus C_0$.

Consider now C_x which has a new variable x substituted for D. Consider the branch from x to the root of C as in Figure 3; the $A_1, \ldots, A_s$ are subformulas of C which are inputs to gates having x in their other input.

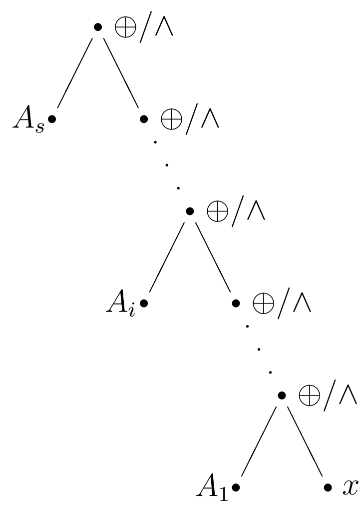

Figure 3

We claim that,

$$C_0 \oplus C_1 \equiv A_{i_1} \wedge \cdots \wedge A_{i_r}$$

where $\{A_{i_1}, \cdots, A_{i_r}\}$ is the subset of $\{A_1, \cdots, A_s\}$ that consists of the inputs to $\wedge$ gates. To prove this, first suppose that some A_{i_j} has value 0: then $C_0 \oplus C_1 = 0$ because the values of that conjunction in C_0 and in C_1 are equal, and therefore C_0 and C_1 have the same value. On the other hand, suppose all A_{i_j}'s have value 1: then the values of the $\wedge$-gates will depend on their other inputs, and thus the value of C_x will depend on the value of x, which implies that $C_0 \oplus C_1$ has value 1.

Let A be the formula $A_{i_1} \wedge \cdots \wedge A_{i_r}$. The leafsize of A is obviously less or equal than the leafsize of C_0. Now we can use the proof of Theorem 4 to prove Theorem 5: the only difference is that instead of using the fact that C is equivalent either to $C_0 \vee (D \wedge C_1)$ or to $C_1 \vee (\neg D \wedge C_0)$, we now use the fact that C is equivalent to $(D \wedge A) \oplus C_0$. The calculations of the bounds on leafsize and depth of C' are identical to those in the proof of Theorem 4. □

References

[1] R. P. Brent, *The parallel evaluation of general arithmetic expressions*, J. Assoc. Comput. Mach., 21 (1974), pp. 201–206.

[2] N. H. Bshouty, R. Cleve, and W. Eberly, *Size-depth tradeoffs for algebraic formulae*, in Proceedings of the 32th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 1991, pp. 334–341.

[3] P. L. II, R. Stearns, and J. Hartmanis, *Memory bounds for recognition of context-free and context-sensitive languages*, in Sixth Annual IEEE Symp. on Switching Circuit Theory and Logical Design, 1965, pp. 191–202.

[4] P. M. Spira, *On time hardware complexity tradeoffs for Boolean functions*, in Proceedings of the Fourth Hawaii International Symposium on System Sciences, 1971, pp. 525–527.