ELSEVIER

Information Processing Letters 61 (1997) 161-164

Information Processing Letters

# Monotone real circuits are more powerful than monotone boolean circuits

Arnold Rosenbloom [1]

*Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4*

Received 1 April 1996
Communicated by S.G. Akl

**Abstract**

Recently Pavel Pudlak (1995) and independently Armin Haken and Steve Cook (1995) gave exponential lower bounds for the size of monotone real circuits computing clique type functions. In both cases, the lower bound was established in the monotone boolean case and then extended to the real case. This left open the question of whether monotone boolean circuits are in general polynomially equivalent to monotone real circuits for boolean functions. By a simple construction, we show that monotone real circuits are exponentially more powerful than general boolean circuits by proving that linear-size log-depth fan in 2 monotone real circuits can compute any slice function. © 1997 Elsevier Science B.V.

*Keywords:* Computational complexity; Circuit complexity

## 1. Introduction

Monotone boolean circuits have been of interest for many years. It is in this powerful circuit setting that Razborov [8] first established an exponential lower bound for an explicit boolean function. [2] Berkowitz (see for example [10]) intimately linked Monotone circuit complexity with general boolean circuit complexity by showing that negations are powerless on slice functions. Consequently, monotone lower bounds on slice functions are of interest as they imply lower bounds on general boolean circuits. Monotone circuits once again became of interest when circuit complexity and propositional proof complexity were linked [6]. Bonet et al. [2] established that a bounded weight cutting plane proof of a clique related propositional tautology could serve as a framework for a polynomially larger monotone boolean circuit computing the clique function. Consequently, the known clique lower bounds [8,1] give rise to lower bounds on the lengths of proofs in the bounded weight cutting plane proof system.

Interest in real monotone circuits arose recently when Pudlak [7] pointed out that a general cutting plane lower bound for the clique related tautologies could be realized if more general monotone lower bounds were established for the clique function. To this end, Pudlak generalized Razborov's clique lower bound and proved the general cutting plane lower bound. At about the same time, Haken [4] introduced a new bottleneck counting technique and used it to obtain a simpler monotone boolean lower bound for a clique-like problem called the broken mosquito screen

[1] Email: arnold@cs.toronto.edu.

[2] Exponential lower bounds for non-explicit boolean functions had previously been established by Shannon [9].


PII S0020-0190(97)00007-0

problem. Cook [5] extended this to a monotone real lower bound for the same problem and so established another general cutting plane lower bound for broken mosquito screen tautologies.

In the remainder of this note we prove that monotone real circuits are a very powerful. We construct linear-size, log-depth monotone real circuits for any slice function and show that there are lots of slice functions which can not be computed by polysize boolean circuits.

## 2. Basic definitions

A *T-circuit* $c$ is an acyclic directed graph with labeled vertices (called gates) and a single sink vertex called the *output* of the circuit. We consider only fan-in 2 circuits, so the in-degree of any vertex in $c$ is at most 2. The *inputs* to $c$ are those vertices with in-degree 0 and are labeled by a member of $\{x_0, x_1, \ldots\}$. Non-input gates (those with in-degree $> 0$) are labeled with the function computed at that gate (functions listed in $T$). The *size of* $c$, sometimes written $|c|$, is the number of vertices in its associated graph. The *depth of* $c$ is the length of the longest input/output path. If the largest input label in $c$ is $x_n$ then $c$ computes a function on bit-strings $\langle b_0, b_1, \ldots, b_m\rangle$ with $m \geqslant n$ by assigning $b_i$ to $x_i$ and evaluating $c$ in the usual way. $\mathcal{C} = \{c_0, c_1, \ldots\}$ is a *T-circuit family* if each $c_i$ is a T-circuit and is defined on bit-strings of length $i$. $\mathcal{C}$ computes a function on inputs of length $n$ by writing $\mathcal{C}(b_0, \ldots, b_{n-1}) = c_{n-1}(b_0, \ldots, b_{n-1})$. The *size of* $\mathcal{C}$ is the function $|\mathcal{C}|(i) = |c_i|$, the *depth of* $\mathcal{C}$ is similarly defined. If $\mathcal{F} = \{f_0, \ldots\}$ is a family of boolean functions ($\mathcal{F}$ is defined on inputs of length $n$ as above) then its corresponding T-circuit family $\mathcal{C}$ consists of a smallest T-circuit $c_i$ computing $f_i$ for each $i$. The *T-circuit size of* $\mathcal{F}$ is then the size of its corresponding T-circuit family.

For *boolean circuits* non-input gates of in-degree 1 are labeled by $\neg$, while those of in-degree 2 are labeled by either $\wedge$ or $\vee$. It should be clear that any boolean function family has an associated boolean circuit family and so the size notion is well defined in the boolean case. *Real monotone circuits* have each non-input node labeled by a nondecreasing real function of its inputs. More formally, define $(x, y) \leqslant (a, b) \Leftrightarrow (x \leqslant a)$ and $(y \leqslant b)$. So $f(x, y) : \mathbb{R}^2 \to \mathbb{R}$ is a monotone real function if $(x, y) \leqslant (a, b) \Rightarrow f(x, y) \leqslant f(a, b)$.

For what follows, we will be considering inputs (bit strings) of length $n$. The $k$th *slice of* $\{0, 1\}^n$ (sometimes written $B_k$ when $n$ is obvious) is the set of length $n$ bit strings

$$\{x \in \{0, 1\}^n \mid x \text{ has exactly } k \text{ bits set to } 1\}.$$

Let $f$ be a boolean function defined for inputs of length $n$. We say that *$f$ is a slice function* if there is some $k$ $(0 \leqslant k \leqslant n)$ such that

$$f(B_i) = \begin{cases} \{0\} & \text{if } i < k, \\ \{1\} & \text{if } i > k. \end{cases}$$

Note that $f$ is unrestricted on the $k$th slice $B_k$. A family of boolean function $\{f_i \mid i = 0, 1, \ldots\}$ is a *slice family* if each $f_i$ is a slice function. Again, it should be clear that any slice family has an associated real monotone circuit family and so the size notion for the family is well defined in the real monotone case.

## 3. The result

**Theorem 1.** *Monotone real circuits can compute any slice family in log-depth, linear-size.*

**Proof.** Let $\mathcal{F} = \{f_i \mid i \in \mathbb{N}\}$ be any slice family. We show that each $f_n$ has small monotone real circuits. Fix $n$ and let $k \in \mathbb{N}$ be such that

$$f_n(x) = \begin{cases} \{0\} & \text{if } i < k, \\ \{1\} & \text{if } i > k. \end{cases}$$

Consider now the two monotone real functions $\text{Order}_+$ and $\text{Order}_-$ defined by

$$\text{Order}_+(x) = \sum_{i=0}^{n-1} (2^{n+1} + 2^i) x_i,$$

$$\text{Order}_-(x) = \sum_{i=0}^{n-1} (2^{n+1} - 2^i) x_i.$$

Each of $\text{Order}_+$ and $\text{Order}_-$ can be computed by linear size, log-depth monotone real circuits.

Consider the partial order induced on $\{0, 1\}^n$ by setting

$$x \leqslant y \Leftrightarrow (\text{Order}_+(x), \text{Order}_-(x)) \leqslant (\text{Order}_+(y), \text{Order}_-(y))$$

We verify only the first of the following two properties of the above ordering, the second is straightforward.

(1) If $x \neq y \in B_j$ then

$$\text{Order}_+(x) < \text{Order}_+(y) \Leftrightarrow \text{Order}_-(x) > \text{Order}_-(y)$$

and so $x \not\leqslant y$ and $x \not\geqslant y$

(2) If $x \in B_a$ and $y \in B_b$ with $a < b$ then

$$\text{Order}_+(x) < \text{Order}_+(y) \quad \text{and}$$
$$\text{Order}_-(x) < \text{Order}_-(y)$$

and so $x < y$.

Let $\overline{x} = \sum_{i=0}^{n-1} x_i 2^i$ be the number whose binary representation is $x$. If $x \neq y \in B_j$ then

$$\begin{aligned} &\text{Order}_+(x) < \text{Order}_+(y) \\ \Leftrightarrow\ & j2^{n+1} + \overline{x} < j2^{n+1} + \overline{y} \\ \Leftrightarrow\ & \overline{x} < \overline{y} \\ \Leftrightarrow\ & -\overline{x} > -\overline{y} \\ \Leftrightarrow\ & j2^{n+1} - \overline{x} > j2^{n+1} - \overline{y} \\ \Leftrightarrow\ & \text{Order}_-(x) > \text{Order}_-(y). \end{aligned}$$

The above properties say that the partial ordering induced on $\{0,1\}^n$ orders the $B_i$ as $B_0 < B_1 < \cdots < B_{n-1} < B_n$ and within any slice, elements are incomparable.

Finally, we can compute $f_n$ as follows. Consider the monotone real circuit with output gate $g$, its first input being $\text{Order}_+$ and its second input being $\text{Order}_-$. $g$ behaves as follows:

$$g(\text{Order}_+(x), \text{Order}_-(x)) = \begin{cases} 0 & \text{if } x \in B_a \text{ with } a < k, \\ 1 & \text{if } x \in B_a \text{ with } a > k, \\ f_n(x) & \text{if } x \in B_k. \end{cases}$$

Note that properties (1) and (2) above allow such a monotone real $g$ to be defined. □

**Lemma 2.** *Most boolean slice families have size at least* $2^{n/2}/10n$.

**Proof.** We follow the proof of Muller as outlined in [3], only we apply the technique to slice functions.

Without loss of generality, we can assume that any negations are at the inputs. If this is not the case, we can push the negations towards the inputs without increasing the circuit size. There are fewer than

$$[2(s+2n+2)^2]^s \tag{1}$$

boolean circuits of size $s$ over $\{\vee, \wedge, \neg\}$. To see this, note that each gate in the circuit can be either an $\vee$ or $\wedge$ (2 choices). Its first input can either be connected to another gate ($s$ choices), a literal ($2n$ choices), or a constant (2 choices). In total, there are $(s+2n+2)$ ways to connect the gates first input. Similarly, there are $(s+2n+2)$ ways of connecting the gates second input, so for one gate we have at most $[2(s+2n+2)^2]$ different choices. So there are at most (1) different possible circuits on $s$ gates.

Now for $s = \lceil 2^{n/2}/10n \rceil$ and sufficiently large $n$, the above bound is less than

$$2^{2^{n/2}/10}.$$

But there are exactly $2^{\binom{n}{n/2}} > 2^{2^{n/2}}$ different $n/2$ slice functions on inputs of length $n$. So at least

$$2^{\binom{n}{n/2}} - 2^{2^{n/2}/10}$$

slice functions require circuits of size greater than $2^{n/2}/10n$. □

Combining the above results we have the following theorem.

**Theorem 3.** *Monotone real circuits are exponentially more powerful than boolean circuits on slice families.*

## Acknowledgements

We thank Steve Cook for reading an earlier draft of this note and making suggestions which led to this improved version. Thanks Steve Cook, Vincent Gogan, Paul Kearney, Rich Paige, Francois Pitt and all my other friends at the department for numerous discussions and making research fun!

## References

[1] N. Alon and R.B. Boppana, The monotone complexity of boolean functions, *Combinatorica* 7 (1987) 1-22.

[2] M. Bonet, T. Pitassi and R. Raz, Lower bounds for cutting planes proofs with small coefficients, in: *Proc. 27th Ann. ACM Symp. Theor. Comput.* (1995) 575-584.

[3] R.B. Boppanna and M. Sipser, The complexity of finite functions, in: *Handbook of Theoretical Computer Science*, Vol. A (Elsevier, Amsterdam, 1990) 757-804.
[4] A. Haken, Counting bottlenecks to show monotone P ≠ NP, in: *Proc. 36th Ann. Symp. on Foundations of Computer Science*, 1995.
[5] A. Haken and S. Coo, An exponential lower bound for the size of monotone real circuits, Manuscript, 1995.
[6] J. Krajíček, Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic, 1995; also: *J. Symbolic Logic*, to appear.
[7] P. Pudlak, Lower bounds for resolution and cutting planes proofs and monotone computations, 1995; also: *J. Symbolic Logic*, to appear.
[8] A.A. Razborov, Lower bounds on the monotone complexity of some boolean functions, *Soviet Math. Dokl.* 31 (1985) 354-357.
[9] C.E. Shannon, The synthesis of two-terminal switching circuits, *Bell Systems Tech. J.*, 1949.
[10] I. Wegener, *The Complexity of Boolean Functions* (1987).