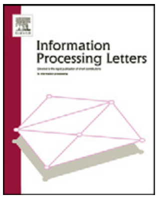
Information Processing Letters 111 (2011) 213–217

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl



The size and depth of layered Boolean circuits

Anna Gál [*,1], Jing-Tang Jang [2]

Dept. of Computer Science, University of Texas at Austin, Austin, TX 78712-1188, USA

ARTICLE INFO

Article history:
Received 14 September 2010
Received in revised form 25 November 2010
Accepted 26 November 2010
Available online 30 November 2010
Communicated by J. Torán

Keywords:
Computational complexity
Boolean circuits
Circuit size
Circuit depth
Pebble games

ABSTRACT

We consider the relationship between size and depth for layered Boolean circuits and synchronous circuits. We show that every layered Boolean circuit of size s can be simulated by a layered Boolean circuit of depth $O(\sqrt{s \log s})$. For synchronous circuits of size s, we obtain simulations of depth $O(\sqrt{s})$. The best known result so far was by Paterson and Valiant (1976) [17], and Dymond and Tompa (1985) [6], which holds for general Boolean circuits and states that $D(f) = O(C(f)/\log C(f))$, where $C(f)$ and $D(f)$ are the minimum size and depth, respectively, of Boolean circuits computing f. The proof of our main result uses an adaptive strategy based on the two-person pebble game introduced by Dymond and Tompa (1985) [6]. Improving any of our results by polylog factors would immediately improve the bounds for general circuits.

1. Introduction

In this paper, we study the relationship between the size and depth of fan-in 2 Boolean circuits over the basis $\{\vee, \wedge, \neg\}$. Given a Boolean circuit C, the size of C is the number of gates in C, and the depth of C is the length of the longest path from any input to the output. We will use the following notation for complexity classes. *DTIME*$(t(n))$ and *SPACE*$(s(n))$ are the classes of languages decidable by deterministic multi-tape Turing machines in time $O(t(n))$ and space $O(s(n))$, respectively. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, define $C(f)$ to be the smallest size of any circuit over $\{\vee, \wedge, \neg\}$ computing f, and define $D(f)$ to be the smallest depth of any circuit over $\{\vee, \wedge, \neg\}$ computing f. Note that $C(f)$ and $D(f)$ are not necessarily achieved by the same circuit.

Pippenger and Fischer [18] showed that for $t(n) \geqslant n$, *DTIME*$(t(n))$ can be simulated by logspace uniform families of circuits of size $O(t(n) \log t(n))$. Borodin [4] showed that for $s(n) \geqslant \log n$, languages computed by logspace uniform families of circuits of depth $s(n)$ are contained in *SPACE*$(s(n))$, and *SPACE*$(s(n))$ can be simulated by logspace uniform families of circuits of depth $O(s^2(n))$. Furthermore, circuit depth is related to parallel computation time [22]. These results show that the study of circuit size versus depth helps to investigate the relationship between sequential and parallel computation time, as well as time versus space in sequential computation. However, very little is known about the size versus depth question for general Boolean circuits. The best known result so far is the following theorem, which was first proved by Paterson and Valiant [17], and later proved by Dymond and Tompa [6] using another method.

Theorem A. *(See [17,6].) Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we have $D(f) = O(C(f)/\log C(f))$.*

On the other hand, it can be easily shown that $D(f) = \Omega(\log C(f))$. Theorem A leaves a huge gap ($\log C(f)$ versus $C(f)/\log C(f)$) for circuits of any size. McColl and Paterson [14] showed that every Boolean function depending on n variables has circuit depth at most $n + 1$. There is an even stronger result by Gaskov [8] showing that circuit depth is

[*] Corresponding author.
E-mail addresses: panni@cs.utexas.edu (A. Gál), keith@cs.utexas.edu (J.-T. Jang).

[1] Supported in part by NSF Grants CCF-0830756 and CCF-1018060.

[2] Supported in part by MCD fellowship from Dept. of Computer Science, UT Austin.

0020-0190/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2010.11.023

at most $n - \log\log n + 2 + o(1)$. This gives a much stronger bound on depth than Theorem A for functions that require circuits of large size. In particular, for $f : \{0,1\}^n \to \{0,1\}$ such that $C(f)$ is exponential in n, [14] and [8] give essentially tight bounds on depth. However, for functions that can be computed by subexponential-size circuits, there is still a large gap. Note that Theorem A gives a stronger result than [14] and [8] only when $C(f) = o(n\log n)$. Improving Theorem A would yield improvements over [14] and [8] for larger $C(f)$ as well.

Because of the connections mentioned above, there are other important consequences if Theorem A can be improved. Hopcroft, Paul, and Valiant [11] proved the following analogous theorem about sequential time and space, and Adleman and Loui [1] later gave an alternative proof.

Theorem B. *(See [11,1].) DTIME$(t(n)) \subseteq$ SPACE$(t(n)/\log t(n))$.*

By the results of [18] and [4] mentioned above, improving Theorem A by at least a polylog factor in a uniform setting immediately improves Theorem B.

For general Boolean circuits, the simulating depth $O(t(n)/\log t(n))$ in Theorem A is very close to the circuit size. On the other extreme, consider tree-like circuits, where every gate has fan-out at most 1. Spira [21] showed that given any tree-like Boolean circuit C of size $t(n)$, we can always simulate C by another tree-like Boolean circuit of depth $O(\log t(n))$. Note that tree-like circuits are commonly referred to as *formulas* in circuit complexity. We will use the term tree-like circuits to avoid any ambiguity. It is unlikely that Spira's result holds for general Boolean circuits, since that would imply $P = NC_1$. Still, it is possible that Theorem A can be improved. We indeed achieve improved simulations for special classes of Boolean circuits.

1.1. Our results

We consider the size versus depth problem for special classes of Boolean circuits. As far as we know, previously no better bounds were known for these classes than what follows from the bounds for general circuits [17,6]. We obtain significant improvements over these general bounds for layered circuits, synchronous circuits, and planar circuits as well as classes of circuits with small separators. Informally, a circuit is layered if its set of gates can be partitioned into subsets called layers, such that every wire in the circuit is between adjacent layers. A circuit is synchronous if for any gate g, every path from the inputs to g has the same length. Synchronous and planar circuits have been extensively studied before. Synchronous circuits were introduced by Harper [10]. Planar circuits were introduced by Lipton and Tarjan [13]. Layered circuits are a natural generalization of synchronous circuits, but as far as we know they have not been explicitly studied. Layered graphs have been studied by Paul, Tarjan, and Celoni [15] (they call these "level graphs" in their paper). Belaga [3] defined locally synchronous circuits, which is a subclass of layered circuits, with the extra condition that each input variable can appear at most once. The synchronous circuits form a proper subset of layered circuits. (See next section for more details.) Furthermore, Turán [23] showed that there exists a function f_n such that any synchronous circuit for f_n has size $\Omega(n\log n)$, but there exists a layered circuit for f_n with size $O(n)$. See Belaga [3] for the same gap for functions with multiple outputs. This distinguishes synchronous circuits and layered circuits with respect to their computational powers. Notice that every Boolean function can be computed by circuits from each of the classes we consider.

Our main result is for layered circuits.

Theorem 1. *Every layered Boolean circuit of size s can be simulated by a layered Boolean circuit of depth $O(\sqrt{s \log s})$ computing the same function.*

We can obtain slightly better bounds for synchronous circuits.

Theorem 2. *Every synchronous Boolean circuit of size s can be simulated by a synchronous Boolean circuit of depth $O(\sqrt{s})$ computing the same function.*

A circuit is planar if its underlying graph can be embedded in the plane without crossings of the wires [13]. In [7] we showed that planar circuits of size s can be simulated by planar circuits of depth $O(\sqrt{s})$. For planar circuits, we used the fact that every planar circuit of size s has a separator of size $O(\sqrt{s})$ [12]. Informally, the separator of a graph is a subset of the nodes whose removal yields two subgraphs of comparable sizes. This allowed us to use a simple divide-and-conquer strategy. Graphs with small separators include trees, planar graphs [12], graphs with bounded genus [9], and graphs with excluded minors [2]. In fact, we can get similar results for arbitrary classes of circuits with small separators.

On the other hand, not all synchronous circuits and layered circuits have small separators. See [20] for many examples. So we need strategies other than the divide-and-conquer approach. Our idea is to consider *cuts*, which separate the graph into two subgraphs that are not necessarily comparable in size. For synchronous circuits, our technique is to find a relatively small cut such that the function can be computed by the composition of two circuits of small depths. This gives a simple proof for synchronous circuits, but the same method cannot be applied to the more general layered circuits. For layered circuits, we develop an adaptive strategy in the two-person pebble game, such that the sizes of the cuts are taken into account during the game. Note that both [17] and [6] use the notion of separators in their proofs. Our results for synchronous circuits and layered circuits show that the minimum circuit depth does not necessarily grow with the separator size of the minimum-size circuit.

Finally we note that an arbitrary circuit of size s can be converted to either a planar or a synchronous circuit of size $O(s^2)$ [24]. Thus improving our results by polylog factors for any of the classes we considered would also yield improvements over the best known bounds for general circuits.