A 2.5n Lower Bound
on the Monotone Network Complexity of $T_3^n$*

Paul E. Dunne
Department of Computer Science, University of Warwick, Coventry CV47AL, Great Britain

Summary. The $k$-th threshold function, $T_k^n$, is defined as:

$$T_k^n(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i \geq k \\ 0 & \text{otherwise} \end{cases}$$

where $x_i \in \{0, 1\}$ and the summation is arithmetic. We prove that any monotone network computing $T_3^n(x_1, \ldots, x_n)$ contains at least $2.5n - 5.5$ gates.

1. Introduction

Let $X = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. It is well known that “almost all” $n$ argument Boolean functions have exponential network complexity, however to date, the best known lower bound on a specific function computed by an unrestricted network is only $3n$ Blum [1]. The difficulty of determining lower bounds on explicitly defined Boolean functions has led to the consideration of restricted forms of networks, the most widely studied of these being monotone networks. In this model, although a number of strong lower bounds have been obtained for networks computing sets of functions (e.g. Mehlhorn [4] and Wegener [7]) the best lower bounds on single output functions are also linear Tiekenheinrich [6].

The threshold functions are an important class of monotone Boolean functions. The $k$-th threshold function is defined as:

$$T_k^n(X) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i \geq k \\ 0 & \text{otherwise} \end{cases}$$

where the summation is arithmetic.

* This research was supported by the Science and Engineering Research Council of Great Britain, UK
We consider the monotone network complexity of $T_3^n(X)$ and prove that any such network computing $T_3^n(X)$ contains at least $2.5n - 5.5$ gates. This improves the previous lower bound of $2n - 3$ and implies similar lower bounds for all threshold functions $T_k^n$ with $k \geq 3$. The monotone network complexity of threshold functions has also been investigated by Bloniarz [2], who obtained a lower bound of $3n$ for the “majority” function, $T_{n/2}^n(X)$, subsequently improved to $3.5n$ [3].

The remainder of this paper is organised as follows. Below we give basic definitions and the notation used. In Sect. (2) we outline the proof technique and prove some preliminary lemmas. In Sect. (3) we prove the main result.

**Definition 1.** A monotone network $S$, is a directed acyclic graph with 2 distinguished sets of nodes: $X$ (the inputs of $S$) is the set of nodes with in-degree 0. These nodes are labelled with members of the set $\{x_1, \ldots, x_n\}$. $G$ is the remaining set of nodes, which have in-degree 2 (the gates of $S$). Gates are labelled by $\land$ or $\lor$ (Boolean conjunction and disjunction). Nodes having out-degree equal to 0 are called the outputs of $S$. The out-degree of an arbitrary node $u$ is called the fan-out of $u$.

**Definition 2.** If $S$ is a monotone network and $g$ is a node of $S$, $\text{RES}(g)$ is the Boolean function recursively defined by:

$$\text{RES}(g) = \begin{cases} x_i & \text{if } g \text{ is the input } x_i \text{ of } S \\ \text{RES}(g_1) \land \text{RES}(g_2) & \text{if } g \text{ is an } \land \text{ gate} \\ \text{RES}(g_1) \lor \text{RES}(g_2) & \text{if } g \text{ is an } \lor \text{ gate} \end{cases}$$

where $g_1$ and $g_2$ are the inputs of $g$ if $g$ is a gate. $S$ computes the monotone Boolean function $f(X)$ if and only if there exists some node $g$ of $S$ with $\text{RES}(g) = f(X)$.

Below, since we are concerned exclusively with networks computing single output functions, $f$, we shall assume that monotone networks have a unique output node, $t$, and that this is the only node of $S$ at which $f$ is computed. All optimal networks for $f$ meet these requirements.

**Definition 3.** $f(X) \leq g(X) \iff \forall x \in \{0, 1\}^n \quad f(x) = 1 \Rightarrow g(x) = 1$.

**Definition 4.** A monom $m$ is a function of the form:

$$m = x_{i1} \land x_{i2} \land \ldots \land x_{ip} \quad x_{ij} \in X.$$ 

A monom $m$ is an implicant of the monotone Boolean function $f$ if and only if $m \leq f$. $m$ is a prime implicant if:

1) $m$ is an implicant of $f$.

2) For all $m'$ such that $m < m'$, $m'$ is not an implicant of $f$.

**Definition 5.** Let $f(X)$ be a monotone Boolean function. The dual of $f$ (denoted $\bar{f}$) is the monotone function:

$$\bar{f}(X) = \neg f(\neg x_1, \neg x_2, \ldots, \neg x_n)$$