数理解析研究所講究録
第1599巻 2008年 91-96

On Patterns of Threshold Circuits computing the PARITY function
(しきい値回路のパターン数について)

東北大学大学院・情報科学研究科　内沢　啓 (Kei Uchizawa)
東北大学大学院・情報科学研究科　瀧本　英二 (Eiji Takimoto)
Graduate School of Information Sciences
Tohoku University

Abstract

In the paper, we prove that any threshold circuit computing the PAITY function of n variables has at least $n+1$ patterns, where a *pattern* is defined to be the sequence of gate states that arise during computation for an assignment.

1 Introduction

Circuits consisting of threshold gates are called threshold circuits, and have been extensively studied for a few decades[1, 2, 3, 4, 5, 6]. Recently, we have introduced a new notion of *patterns* into threshold circuits [7]. For an input assignment $\boldsymbol{x}$, the pattern for $\boldsymbol{x}$ is defined to be the sequence of gate states that arise during computation for the assignment. In Ref. [7], we show that the number of patterns that arise in a threshold circuit is closely related to the size of the circuit,, where the size of a circuit is the number of gates contained in the circuit. In particular, in Ref.[7], by estimating the number of patterns that arise in threshold circuits, we prove that threshold circuits with some restrictions need an exponential number of gates to compute a particular Boolean function, i.e., the Inner-Product function. However, our argument fails to derive non-trivial lower bounds for many simpler functions, including the PARITY function.

In the paper, we consider threshold circuits computing the PARITY function, and derive a lower bound on the number of patterns of threshold circuits. More precisely, we prove that threshold circuits computing the PARITY function of n variables must have at least $n+1$ patterns.

Moreover, from the lower bound we derive a tight lower bound on the size of threshold circuits computing the PARITY function. Note that one can find the same lower bound derived by a different proof method in [6]. However, we derive the lower bound, since it is a good example to give a insight into the relation between patterns and the size of circuits.

2 Definitions

In the section, we first give definitions and several terms needed to describe our results.

For every input $\boldsymbol{z} = (z_1, z_2, \ldots, z_m) \in \{0,1\}^m$, a *threshold gate* g (with *weights* $w_1, w_2, \ldots, w_m$ and a *threshold* t) computes a linear threshold function given by

$$g(\boldsymbol{z}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} w_i z_i \geq t; \\ 0 & \text{otherwise.} \end{cases}$$

A *threshold circuit* C with n input variables is represented by a directed acyclic graph; the graph has exactly n nodes of in-degree 0, each associated with an input variable and called an *input node*; each of the other nodes represents a threshold gate. For an assignment $\boldsymbol{x} \in \{0,1\}^n$ to the n input variables, the output of all gates in C are computed in topological order of the nodes in the directed acyclic graph. For a gate g in C, we denote by $g[\boldsymbol{x}]$ the output of g for an input $\boldsymbol{x}$ to circuit C (although the actual input to gate g will in general consist of some variables from $\boldsymbol{x}$ and, in addition, or even exclusively, the outputs of some other gates in C).

The *size* of a threshold circuit C is the number of gates in C. Since we consider only a threshold circuit that computes a Boolean function, one may assume without loss of generality that the circuit has exactly one gate of out-degree 0, called the *top gate*. We say that a threshold circuit C *computes* a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if the output of the top gate for $\boldsymbol{x}$ equals to $f(\boldsymbol{x})$ for every input $\boldsymbol{x} \in \{0,1\}^n$.

Assume that a threshold circuit C consists of m threshold gates, $g_1, g_2, \ldots, g_m$ for some $m \geq 1$. For an input assignment $\boldsymbol{x}$ for C, we call the m-tuple of the gate outputs,

$$(g_1[\boldsymbol{x}], g_2[\boldsymbol{x}], \ldots, g_m[\boldsymbol{x}]),$$

the *pattern of C for x*, and we say that the pattern *arises for x in C*. Let $PAT(C)$ be the set of all patterns of C. That is,

$$PAT(C) = \{(g_1[\boldsymbol{x}], g_2[\boldsymbol{x}], \ldots, g_m[\boldsymbol{x}]) \mid \boldsymbol{x} \in \{0,1\}^n\}.$$

For every input assignment

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n,$$

the PARITY function of n variables is defined to be

$$\text{PARITY}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \text{ is odd;} \\ 0 & \text{if } \sum_{i=1}^n x_i \text{ is even.} \end{cases}$$

3 Main Result

In the section, we prove the theorem below.

Theorem 1. *Any threshold circuit C computing the PARITY function of n variables has at least $n+1$ patterns. That is,*

$$|PAT(C)| \geq n+1.$$

To give a proof of the theorem, we need the following two lemmas, Lemma 1 and Lemma 2. We prove Lemma 1 in the section, while we prove Lemma 2 in the next section.

Lemma 1. *Any threshold circuit C computing the PARITY function of two variables has at least three patterns. That is,*

$$|PAT(C)| \geq 3.$$

Proof. Let C be a threshold circuit computing the PARITY function of two variables, x_1 and x_2. Assume that the circuit C contains threshold gates $g_1, g_2, \ldots, g_m$ indexed in the topological order, where m is the size of C. That is, for every index i, the gate g_i receives inputs only from $g_1, g_2, \ldots, g_{i-1}$ as well as from input variables x_1 and x_2. That is, for each index i, $1 \leq i \leq m$, we have

$$g_i[x_1, x_2] = \\ g_i(x_1, x_2, g_1[x_1, x_2], \ldots, g_{i-1}[x_1, x_2])$$

We prove the lemma by contradiction. Assume that the circuit has just two patterns. Clearly, one of the two patterns must be for the case where $x_1 + x_2$ is even, and the other must be for the case where $x_1 + x_2$ is odd. Therefore, one of the two patterns arises for inputs $(x_1, x_2) = (0,0), (1,1)$, and the other does for $(x_1, x_2) = (0,1), (1,0)$. Then, let i be the least index such that $g_i[0,0] \neq g_i[0,1]$. Note that $g_i[1,1] = g_i[0,0]$ and $g_i[0,1] = g_i[1,0]$. Since the outputs of the gates g_j for $j < i$ are considered to be a constant , we can consider that the gate g_i computes a threshold function of x_1 and x_2. More precisely, the function f defined as

$$f(x_1, x_2) = g_i[x_1, x_2] = \\ g_i(x_1, x_2, g_1[x_1, x_2], \ldots, g_{i-1}[x_1, x_2])$$

is a threshold function. Since $g_i[0,0] = g_i[1,1] \neq g_i[0,1] = g_i[1,0]$, we have

$$f(0,0) = f(1,1) \neq f(0,1) = f(1,0),$$

which implies that f computes the PARITY function of two variables. This contradicts the fact that the PARITY function is not a threshold function. □

Lemma 2. *Let C be any threshold circuit computing the PARITY function of n variables. Let C_0 be the threshold circuit obtained by replacing the input node x_n of C with constant input 0. Then*

$$|PAT(C_0)| \leq |PAT(C)| - 1.$$

Using the two lemmas, we prove the theorem in the following.

Proof (of the theorem) We will prove by induction on n that

$$|PAT(C)| \geq n + 1 \tag{1}$$

for any threshold circuit C computing the PARITY function of n variables.

Obviously, Lemma 1 confirms the basis, i.e., the case where $n = 2$, of the induction.

Below we show the induction step. Let C be any threshold circuit computing the PARITY function of n variables. Let C_0 be the circuit obtained by replacing the input node x_n of C with constant input 0. Since C_0 computes the PARITY function of $n - 1$ variables, the induction hypothesis implies that

$$|PAT(C_0)| \geq n. \tag{2}$$

By Lemma 2 and Eq. (2), we have

$$|PAT(C)| \geq |PAT(C_0)| + 1 \geq n + 1,$$

which confirms Eq. (1). □

By Theorem 1, we can easily derive as below that any threshold circuit computing the PARITY function of n variables needs $\log(n+1)$ gates. Althogh one can find the same lower bound derived by a different proof method in [6], we put it as corolary to give a insight into the relation between patterns and the size of circuits.

Corollary 1. *Every threshold circuit computing the PARITY function of n variables has at least $\log(n+1)$ gates.*

Proof. By Theorem 1, any threshold circuit computing the PARITY function of n variables needs $n+1$ patterns. To realize $n+1$ patterns, the circuit needs $\log(n+1)$ gates. □

This lower bound is tight, since the PARITY function is computable by a threshold circuit of size $O(\log n)$ [6].

4 Proof of Lemma 2

In the rest of the paper, we prove Lemma 2.

Let C be a threshold circuit computing the PARITY function of n variables, and let m be the size of C. Assume that the circuit C contains threshold gates $g_1, g_2, \ldots, g_m$ indexed in topological order. Let C_0 be a circuit obtained by replacing the input node x_n of C with constant input 0. We prove that

$$|PAT(C_0)| \leq |PAT(C)| - 1. \tag{3}$$

We give a proof by contradiction. Assume that

$$|PAT(C_0)| = |PAT(C)|, \tag{4}$$

that is,

$$PAT(C_0) = PAT(C). \tag{5}$$

Similarly to the definition of C_0, let C_1 be a circuit obtained by replacing the input node x_n of C with constant input 1. By Eq. (5), we have

$$PAT(C_1) \subseteq PAT(C_0) = PAT(C). \tag{6}$$

Let

$$X_0 = \{(x_1, x_2, \ldots, x_n) \in \{0,1\}^n \mid x_i = 0\}.$$

Now we consider the following sequence of patterns, $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s$ of length $s = |PAT(C)|+1$. The sequence starts with any pattern $\boldsymbol{p}_1 \in PAT(C_1)$. Let $\boldsymbol{x}_1 \in X_0$ be an input for which the pattern $\boldsymbol{p}_1$ arises in C. Equation (6) guarantees that there must exists such input $\boldsymbol{x}_1$. For any positive integer j, $1 \leq j \leq s$, the $j+1$-th pattern $\boldsymbol{p}_{j+1} \in PAT(C)$ of the sequence is the one that arises for the input $\boldsymbol{x}'_j$ in which all bits but the n-th bit are the same as those in $\boldsymbol{x}_j$. Let $\boldsymbol{x}_{j+1} \in X_0$ be an input for which the pattern $\boldsymbol{p}_{j+1}$ arises in C. Equation. (6) also guarantees that there must exist such input $\boldsymbol{x}_{j+1}$.

Since the length s of the sequence is larger than $|PAT(C)|$, there must exist a pattern that appears twice in the sequence. Assume without loss of generality that the pattern $\boldsymbol{p}_1$ appears twice. That is,

$$\boldsymbol{p}_1 = \boldsymbol{p}_k \tag{7}$$

for some $k \geq 2$.

Using the sequence, we next define a sequence of gates. For each integer j, $1 \leq j \leq k$, we choose the j-th gate of the sequence as follows: the j-th gate has the least index among the gates g such that $g[\boldsymbol{x}_j] \neq g[\boldsymbol{x}_{j+1}]$. Let I_j be the index of the j-th gate of the sequence. Furthermore, let

$$t = \arg \min_{1 \leq t \leq k} I_t. \tag{8}$$

Now we look at the outputs of the gate g_{I_t} for inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$. Note that $g_{I_t}[\boldsymbol{x}_j]$ is the I_t-th bit of the t-th pattern $\boldsymbol{p}_t$. Assume without loss of generality that

$$g_{I_t}[\boldsymbol{x}_1] = 0. \tag{9}$$

By the definition of g_{I_t}, we have

$$g_{I_t}[\boldsymbol{x}_j] = 0$$

for every index j, $1 \leq j \leq t$, and

$$g_{I_t}[\boldsymbol{x}'_t] = g_{I_t}[\boldsymbol{x}_{t+1}] = 1$$

This implies that the gate g_{I_t} has a positive weight for the input variable x_i. Equation (8) together with the fact implies that

$$g_{I_t}[\boldsymbol{x}_j] = 1$$

for every index $j \geq t+1$. Therefore, we have

$$g_I[\boldsymbol{x}_k] = 1. \tag{10}$$

Equations (9) and (10) contradict Eq. (7).

References

[1] E. Allender, *Circuit complexity before the dawn of the new millennium*, Foundations of Software Technology and Theoretical Computer Science (1996), 1–18.

[2] M. Anthony, *Boolean functions and artificial neural networks*, CDAM Research Report LSE-CDAM-2003-01 (2003).

[3] A. Bertoni and B. Palano, *Structural complexity and neural networks*, Proceedings of the 13th Italian Workshop on Neural Nets-Revised Papers **2486** (2002), 190–215.

[4] M. Minsky and S. Papert, *Perceptrons: An introduction to computational geometry*, MIT Press, 1988.

[5] I. Parberry, *Circuit complexity and neural networks*, MIT Press, 1994.

[6] K. Y. Siu, V. Roychowdhury, and T. Kailath, *Discrete neural computation; a theoretical foundation*, Information and System Sciences Series, Prentice Hall, 1995.

[7] K. Uchizawa and E. Takimoto, *An exponential lower bound on the size of threshold circuits with small energy complexity*, Proceedings of the 22nd Annual IEEE Conference on Computational Complexity (2007), 169–178.