Further Results on False-Assignments

Pei Hong Mei*
Department of Basis
Dalian Naval Academy
Dalian, China
E-mail: hongmei_005@163.com
* Corresponding Author

Wang Xin
Department of Basis
Dalian Naval Academy
Dalian, China
E-mail: 63180583@qq.com

Shang Jie Lin
Department of Basis
Dalian Naval Academy
Dalian, China
E-mail: 343065787@qq.com

Abstract—In this article, we made a further studies on the false-assignments of Boolean Function, which made us not only get the theory evidence by using false-assignments to judge the function's elusiveness, but also verify the result's effectiveness by using specific function.

Keywords- Truth-Assignments; False-Assignments; Elusive; Boolean Function; Decision Tree Complexity

I. INTRODUCTION

It is well known that graph theory has extensive applications in many areas. An important thing concerning to application is to treat graph with computer. Usually, a graph can be installed into a computer by encoding the entries of the triangular part of its adjacency matrix. One problem arising naturally is: Can we detect whether a graph G has some specific property without decoding all the entries of the upper triangular part of its adjacency matrix? That is, given a graph property P on n vertices, must it take $n(n-1)/2$ queries in the worst case to determine whether a graph G is in P? People guess the answer is yes when P is nontrivial and monotone. This is the well-known Karp conjecture which is still open and becomes a well-known difficult problem in computational complexity theory. In last few years, many researchers have paid their efforts to decision tree complexity and got some results [1-10].

Boolean function is one of the main tools which used to research Karp conjecture and it did obtain splendid research achievements. However, currently the research achievements are discussed aimed at truth-assignments of Boolean function, while few people research on false-assignments. This article looks at problems from a unique perspective to explore false-assignments' application on constructing Boolean function expression and judging Boolean function's elusiveness.

II. PRELIMINARY

A tree is a connected graph without cycle. A rooted tree is a tree with a special vertex named root.

Definition 2.1[1] A decision tree of a Boolean function, is a rooted binary tree, whose non-leaf vertices are labeled by its variables, and leaves are labeled by 0 and 1.

Definition 2.2[1] The depth of a decision tree is the maximum length of all paths from root to leaves.

Definition 2.3[1] A Boolean function is a function whose variable values and function value all are in $\{0,1\}$.

Definition 2.4[1] The decision tree complexity of f is the minimum depth of all decision trees for computing f, denoting by $D(f)$.

Definition 2.5[1] A Boolean function f is said to be elusive if $D(f)=n$.

III. MAIN RESULTS

A. Constructing Boolean Function by Using False-Assignments

This article proposes a new algorithm named dual algorithm to figure out Boolean function, which is to build Boolean function by using false assignments.

Dual Algorithm for every false assignment, build Boolean sum as below method firstly: if the valuation of variable is 0, then will be put in Boolean sum. Otherwise, if the valuation of variable is 1, it will be put in Boolean sum. Multiply Boolean sum that built by all the false assignments by Boolean product, then we will obtain the Boolean function's expression that expressed by Boolean sum, Boolean product and the operation of negation.

Based on the assignment of variables and functions values listed in table 1, we will obtain the Boolean function by using the traditional algorithm [1] [2]:

$$f(x_1,x_2)=\bar{x}_1x_2+x_1\bar{x}_2 \quad (1)$$

Calculate by using above dual algorithm, we will obtain the below Boolean function:

$$f(x_1,x_2)=(x_1+x_2)(\bar{x}_1+\bar{x}_2) \quad (2)$$

TABLE I. FUNCTIONAL RELATIONSHIPS TABLE

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

However,

$$\begin{aligned} f(x_1, x_2) &= (x_1 + x_2)(\bar{x}_1 + \bar{x}_2) \\ &= x_1\bar{x}_1 + x_1\bar{x}_2 + x_2\bar{x}_1 + x_2\bar{x}_2 \\ &= x_1\bar{x}_2 + x_2\bar{x}_1 \end{aligned}$$

Thus we can find that for the Boolean function listed in Table 1, the Eq. 1 calculated by using the traditional algorithm is the same with the equation Eq. 2 that calculated by dual algorithm which we proposed in this article. Then we presume that calculating by dual algorithm will obtain the same result with that calculating by the traditional algorithm under any circumstances. Then we will prove this result we presumed.

Theorem 3.1 The Boolean function obtained by dual algorithm is correct.

Proof *to* prove the Boolean function expression that calculated by dual algorithm is correct, we only need to prove that for any assignment of variables, we will obtain the same Boolean function value with the results listed in the function relationships table.

Let be a Boolean function with *n* variables, which is obtained by multiply *k* Boolean sums by Boolean product using the dual algorithm. Since each Boolean sum is determined by the definition of the dual algorithm, the *k* Boolean sums include all the false assignment terms. Then, for any assignment of variables, we have

(1) If it is a false assignment, then the Boolean function has the corresponding Boolean sum with it, and the value is 0, thus the function value is 0. It is the same with given results.

(2) If it is a true assignment, then the function expression doesn't have the corresponding Boolean sum with it, thus all the Boolean sums' values in the Boolean function are 1. Otherwise, if there is a Boolean sum which value is 0, according to the definition of the function, this group of assignment is false assignment, which is at odds with our assumption. So the function value is 1, the same with given results.

Conclusions as a result, for any group of assignment of n variables, the values of Boolean function calculated by dual algorithm is always the same with given values, which proves that the dual algorithm is correct.

We have proved the correctness of the dual algorithm. But is it redundant that we propose the dual algorithm while there is already an algorithm exists? Does the dual algorithm itself have practical use? Then we will show how the dual algorithm is indispensable by comparing the dual algorithm with the traditional algorithm.

It is obviously that while calculating a Boolean function's expression, choose which algorithm to calculate can have a smaller calculation amount is mainly depended on which assignments' number are smaller in all the assignments. That is, while true assignments' number is smaller than that of false assignments', then using traditional algorithm is preferred. While true assignments' number is greater than false assignments' number, then using the dual algorithm that we proposed is preferred, in this occasion, the calculation amount is much smaller than using the traditional algorithm. So we can find that these two algorithms oppose each other also complement each other, thus the dual algorithm proposed in this article is indispensable.

B. *Judging Boolean Function's Elusiveness by Using False-Assignments*

In the preceding part of the text, we provided a new algorithm for constructing Boolean function. With function expression， we can use it to judge elusiveness of function. On the basis of documentation [1] ， we use false-assignments to judge the elusiveness of Boolean function and thus get the conclusion as below：

Theorem 3.2 If the false-assignments of a Boolean function $f(x_1, x_2, \cdots, x_n)$ is odd, then $f(x_1, x_2, \cdots, x_n)$ is elusive.

Proof Here we use mathematical induction to prove variable's number n ：

(1) when $n = 1$， there are only 4 function exist which are

$$f(x_1) = x_1 \,,\, f(x_1) = \bar{x}_1 \,,\, f(x_1) \equiv 0 \,,\, f(x_1) \equiv 1$$

respectively, among which both

$$f(x_1) = x_1 \text{ and } f(x_1) = \bar{x}_1$$

only have one false-assignment respectively. So we can easily know that they are elusive.

On the contrary, constant functions

$$f(x_1) \equiv 0 \text{ and } f(x_1) \equiv 1$$

are not elusive，which the numbers of false-assignments are both even.

Therefore the theorem establishes under the condition $n = 1$.

(2) when $n > 1$ investigate Boolean function $f(x_1, x_2, \cdots, x_n)$. Assume that T is the decision tree of $f(x_1, x_2, \cdots, x_n)$ which has the minimum depth and variable x_i as its root. Then the two subtrees T_0 and T_1 of the root can express to function

$$f\big|_{x_i=0} = f(x_1, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n)$$

and

$$f\big|_{x_i=1} = f(x_1, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n)$$

Respectively. Both of them are function of $n-1$ variables. According to the condition, the number of false-assignments of the Boolean function $f(x_1, x_2, \cdots, x_n)$ is odd, and it is also sum of $f\big|_{x_i=0}$'s false-assignments and $f\big|_{x_i=1}$'s false-assignments. So for the $n-1$ variables function $f\big|_{x_i=0}$ and $f\big|_{x_i=1}$, there must be one of them which has an odd number of false-assignments, from induction we assume that $D\left(f\big|_{x_i=0}\right)=n-1$ or $D\left(f\big|_{x_i=1}\right)=n-1$. Therefore the depth of subtree T_0 or T_1 is $n-1$, decision tree T 's depth is n, in other words, $D(f)=n$.

We have proved the correctness of the theorem 3.2. It is an effective way which uses the number of false-assignments to judge the elusiveness of function. For example function $f(x_1, x_2)=x_1 x_2$, of which the functional relationships table shown as table 2, it has 3 false-assignments, it is elusive.

TABLE II. FUNCTIONAL RELATIONSHIPS TABLE

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

While the number n of variables is a relatively large number, judging the elusiveness of the function by using theorem 3.2 will be a troublesome thing. Below we will extend theorem 3.2, give the following definition.

Definition 3. 1 $q_f(t)=\sum\limits_{x_1, x_2, \cdots, x_n \in\{0,1\}} \bar{f}(x_1, x_2, \cdots, x_n) t^{\|x\|}$,

of which $\|x\|$ expresses the number of variables that with value 0 in $x_1, x_2, \cdots, x_n$, and if

$$f(x_1, x_2, \cdots, x_n)=0,$$

Then

$$\bar{f}(x_1, x_2, \cdots, x_n)=1,$$

According to definition 3.1 we can know that $q_f(1)$ is the number of false-assignments, and according to theorem 3.2, we get the corollary shown as below.

Corollary 3.1 If $q_f(1)$ is odd, then $f(x_1, x_2, \cdots, x_n)$ is elusive.

Theorem 3.3 $(t+1)^{n-D(f)} \big| q_f(t)$

Proof To judge the tree's complexity $D(f)$ by using mathematical induction:

(1) If $D(f)=0$, Boolean function $f(x_1, x_2, \cdots, x_n)$ is a constant function that is

$$f(x_1, x_2, \cdots, x_n) \equiv 0$$

or

$$f(x_1, x_2, \cdots, x_n) \equiv 1 .$$

If $f(x_1, x_2, \cdots, x_n) \equiv 0$, then

$$q_f(t)=\sum_{x_1, x_2, \cdots, x_n \in\{0,1\}} \bar{f}(x_1, x_2, \cdots, x_n) t^{\|x\|}=(t+1)^n ;$$

thus the theorem conclusion established;

If $f(x_1, x_2, \cdots, x_n) \equiv 1$, then

$$q_f(t)=\sum_{x_1, x_2, \cdots, x_n \in\{0,1\}} \bar{f}(x_1, x_2, \cdots, x_n) t^{\|x\|}=0 ,$$

the theorem conclusion established too;

(2) If $D(f)>0$, assume that T is the decision tree of $f(x_1, x_2, \cdots, x_n)$ which has the minimum depth and variable x_i is its root node. Therefore the tree's two subtrees T_0 and T_1 can express the function

$$f\big|_{x_i=0}=f(x_1, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n)$$

and

$$f\big|_{x_i=1}=f(x_1, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n)$$

respectively. We denoted by $f_0=f\big|_{x_i=0}$ and $f_1=f\big|_{x_i=1}$, then

$$D(f_0) \leq D(f)-1$$

and

$$D(f_1) \leq D(f)-1 ,$$

Assuming from induction,

$$(t+1)^{(n-1)-D(f_0)} \big| q_{f_0}(t)$$

and

$$(t+1)^{(n-1)-D(f_1)} \big| q_{f_1}(t) ,$$

thus

$$(t+1)^{n-D(f)} \Big| q_{f_0}(t)$$

and

$$(t+1)^{n-D(f)} \Big| q_{f_1}(t)$$

According to Definition 1 we can learn that,

$$\begin{aligned} q_f(t) &= \sum_{x_1, x_2, \cdots, x_n \in \{0,1\}} \bar{f}(x_1, x_2, \cdots, x_n) t^{\|x\|} \\ &= \sum_{x_1, \cdots, x_{i-1}, x_{i+1} \cdots, x_n \in \{0,1\}} \bar{f}(x_1, \cdots x_{i-1}, 0, x_{i+1} \cdots, x_n) t^{\|x\|} \\ &\quad + \sum_{x_1, \cdots, x_{i-1}, x_{i+1} \cdots, x_n \in \{0,1\}} \bar{f}(x_1, \cdots x_{i-1}, 1, x_{i+1} \cdots, x_n) t^{\|x\|} \\ &= t q_{f_0}(t) + q_{f_1}(t). \end{aligned}$$

Thus $(t+1)^{n-D(f)} \Big| q_f(t)$.

According to theorem 3.2 and theorem 3.3 we can get the corollary shown as below.

Corollary 3.2 If $q_f(-1) \neq 0$, then $f(x_1, x_2, \cdots, x_n)$ is elusive.

According to definition 3.1, we can learn that $q_f(-1)$ is the difference of the false-assignments assigned 0 which has an even number and the false-assignments assigned 0 which has an uneven number. This is also an effective way to judge the elusiveness of a Boolean function. For example, the Boolean function of 3 variables

$$f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 + x_2 x_3,$$

of which the functional relationships table shown as table 3. Its

$$q_f(1) = 4,$$

hence we can't tell its elusiveness by theorem 3.2. Theorem 3.2 loses effectiveness. But its

$$q_f(-1) = 3 - 1 = 2 \neq 0,$$

so it is an elusive function on the basis of corollary 3.2. It seems that corollary 3.2 is a more effective result than theorem 3.2. In fact, if the false-assignments of a function is odd, that is $q_f(1)$ odd, then it $q_f(-1) \neq 0$. It means that corollary 3.2 is indeed a stronger result than theorem 3.2.

TABLE III. FUNCTIONAL RELATIONSHIPS TABLE

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

IV. CONCLUSIONS

This article made a discussion about Boolean function's false-assignments, and got the theory evidence of judging the function's elusiveness, the next step will continue to research the false-assignments' applying in judging the function's elusiveness.

ACKNOWLEDGMENT

This work was supported by Dalian Naval Academy for Basic Research.

REFERENCES

[1] Du Ding-zhu, Decision Tree Theory [M]. Hunan Education Press 1998.

[2] Li De-ying, Du Ding-zhu. Graph Properties Based on the Karp Conjecture[J]. Chinese Science Bulletin, 2000: 2129-2134.

[3] Wang Shuhe. Graph Theory[M]. Science Press, 1998.

[4] Best M R, van Emde Boas P, H.W.Lenstra et al. A sharpened version of the Aanderaa-Tosenberg Conjucture, Report ZW 30/74, Mathematisch Centrum Amsterdam 1974.

[5] A. L. Rosenberg, On the time require to recognize peoperties of graphs: A problem, SIGACT News, 5:4(1973) 15-16.

[6] R. L. Rivest, S. Vuillemin, A generalization and proof of the Aanderaa-Rosenberg conjecture, in Proc. SIGACT Conf, Albuquerque, 1975.

[7] R. L. Rivest, S. Vuillemin, On recognizing graph properties from adjacency matrices, Theor. Comp. Sci., 1976, 3:371~384.

[8] S.-X. Gao, X. Hu, W. Wu, Nontrivial monotone weakly symmetric Boolean functions of six variables are elusive, Theoret. Comput. Sci., 1999, 223:193-197.

[9] S.-X. Gao, X. Hu, W. Wu et al, Nontrivial monotone weakly symmetric Boolean functions of ten variables are elusive, J. Comples. 15(1999):526-536.

[10] S.-X. Gao, X. Hu, W. Wu et al, Nontrivial monotone weakly symmetric Boolean functions of twelve variables are elusive, Discrete Math., 253(2002):19-34.