Research and Application of False-Assignments

Hong-mei PEI, Xuan-hai LI, Mei-li ZHANG and Jie-lin SHANG

Department of Basic, Dalian Naval Academy Liaoning, Dalian, China

*Corresponding author

Keywords: Complex, False-assignments, Boolean function, Elusive, Decision tree complexity.

Abstract. This article makes a further research on Boolean function’s false-assignment, by making a combination of topological and algebraic method, we get the way of how to use false-assignment to judge the function’s elusive.

Introduction

It is well known that a Boolean function is a function whose variable values and function value are all in \{0,1\}, besides, it is a very common function, which now have achieved many research results [1-3]. \(x \lor y\) and \(x \land y\) are the two main operations of Boolean function. We call \(\lor\) as disjunction, and we call \(\land\) as conjunction. Another often used operation of Boolean function is negation operation, which is marked as. Anyone of Boolean function can be expressed by three kinds of operations, which are disjunction, conjunction and negation operation. Other operations can be expressed by disjunction, conjunction and negation. For example, Boolean function \(f(x, y)\) is:

\[f(x, y) = (\overline{x} \land y) \lor (x \land \overline{y})\]  

For simplicity, we mark \(x \lor y\) and \(x \land y\) as \(x+y\) and \(xy\), thus the Boolean function that listed above can be expressed as

\[f(x, y) = \overline{x}y + x\overline{y}\]  

An assignment for a Boolean function is a mapping from its variables to \(\{0,1\}\), each variable gets one value from the assignment. For \(n\) variables function \(f(x_1, x_2, \cdots, x_n)\), the number of its assignment is \(2^n\). An assignments that makes \(f(x_1, x_2, \cdots, x_n)=1\) we call it a true-assignment, and an assignment that makes \(f(x_1, x_2, \cdots, x_n)=0\) we call it a false-assignment.

The same with general function, a Boolean function’s monotonicity can be also defined. If the two groups of variable assignment meet the conditions \(x_1 \leq y_1, \quad x_2 \leq y_2, \quad \cdots, \quad x_n \leq y_n\), then it can be expressed as \((x_1, x_2, \cdots, x_n) \leq (y_1, y_2, \cdots, y_n)\). Make an assignment to any two groups of variable, if

\[(x_1, x_2, \cdots, x_n) \leq (y_1, y_2, \cdots, y_n),\]  

then

\[f(x_1, x_2, \cdots, x_n) \leq f(y_1, y_2, \cdots, y_n),\]  

thus we can say that the Boolean function \(f\) is increasing; On the contrary, if any of the

\[(x_1, x_2, \cdots, x_n) \leq (y_1, y_2, \cdots, y_n)\]  

can be

\[f(x_1, x_2, \cdots, x_n) \geq f(y_1, y_2, \cdots, y_n)\]
then we can say that the Boolean function \( f \) is decreasing. A Boolean function is monotone if it is either increasing or decreasing, and it is nontrivial if it is not a constant function.

Besides the function expression, Boolean function can also be expressed by using binary tree which have special labels. 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. On every path that from root to leaf (called root leaf path for short), a variable appears no more than once, and, the two edges that from every inner vertex to its children also be labeled by 0 and 1 respectively. Give a group of variable assignments, and the corresponding function value can be calculated as below rules: start from root, and investigate the vertex. If the vertex is variable \( x_i \), while \( 0 = x_i \), then go down along the side labeled 0; While \( 1 = x_i \), then go down along the side labeled 1 until to a leaf, the label of the leaf is function value.

The maximum length of the root leaf path that on the decision tree is called the decision tree’s depth. For Boolean function \( f(x_1, x_2, \ldots, x_n) \) whose variable number is \( n \), it has limited numbers of decision trees, the number of them is \( 2^2 \cdot 2^1 \cdot 2^2 \cdot \ldots \cdot 2^1 \) all together. The minimum value of all the decision trees’ depth is named as function \( f(x_1, x_2, \ldots, x_n) \)’s decision tree complexity, labeled as \( D(f) \). It is obvious that \( D(f) \leq n \). If \( D(f) = n \), then function \( f \) is elusive.

People guess that every nontrivial monotone weakly symmetric Boolean function is elusive. This is the well-known Rivest-Vuillemin 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 [4-10].

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 mostly discussed aimed at truth-assignments of Boolean function, while few people research on false-assignments. In the forepart, we discussed the false-assignment’s application on constructing Boolean function and judging the function’s elusive, based on this, this article will make a further discussion on false-assignments’ application on judging Boolean function’s elusiveness.

**Preliminary**

In geometry, an \( n \)-dimensional simplex is an \( n \)-dimensional non-degenerate polyhedron that has \( n+1 \) numbers of vertex. For example, spot is a 0-dimension simplex, segment is a 1-dimension simplex, and triangle is a 2-dimension simplex. A simplex’s face is the subsets of its vertex sets. In algebra, the simplex is always regarded as a finite set \( \{v_1, v_2, \ldots, v_n\} \) and all of the function that satisfies below conditions:

\[
g(v_1) + g(v_2) + \cdots + g(v_n) = 1, \tag{8}
\]

\[
g(v_1) \geq 0, g(v_2) \geq 0, \cdots, g(v_n) \geq 0. \tag{9}
\]

Based on such understanding, we named a finite set \( V \)’s subsets family \( \Delta \) as a simplicial complex, if

1. \( A \subset B, B \in \Delta \Rightarrow A \in \Delta \);
2. \( \Delta \) includes a limited numbers of finite set.

Simplicial complex is called complex for short. Each finite set is a face of the complex. Each face’s dimension is equal to its element number minus 1, in a word, \( A \in \Delta, \dim(A) = |A| - 1 \). 0-dimension is a set which has only one element, which is also called vertex. Every \( n \)-dimensional face is an \( n \)-dimensional simplex.

To complex, the Euler characteristic is an important invariant, while a simplex \( \Delta \)’s Euler characteristic is defined as

101
\[
\chi(\Delta) = \sum_{A \in \Delta} (-1)^{\dim(A)}
\]  

(10)

While complex \(\Delta\) only has an null set, we define that \(\chi(\Delta) = 0\), while \(\Delta\) is a null family, we define that \(\chi(\phi) = 1\).

For monotone Boolean function \(f\), the corresponding complex can also be structured. That is to say, assume \(f(x_1, x_2, \cdots, x_n)\) is a decreasing Boolean function, define that

\[
\Delta_f = \left\{ A \subseteq \{x_1, x_2, \cdots, x_n\} \mid f|_{x_1=0, x_i\in A} \equiv 1 \right\}
\]

(11)

**Main Results**

By discussing false-assignment’s application on constructing Boolean function and judging function’s elusive, we get the conclusion as follows:

*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, then 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.

**Definition 1**

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

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 \(f(x_1, x_2, \cdots, x_n) = 1\).

**Lemma 1** If \(q_f(-1) \neq 0\), then \(f(x_1, x_2, \cdots, x_n)\) is elusive.

Above monotone function’s complex is defined by using true-assignment, then we will construct complex by using false-assignment, and discuss its application on judging the function’s elusive.

**Definition 2** If \(f(x_1, x_2, \cdots, x_n)\) is a decreasing Boolean function, define that

\[
\Delta_f = \left\{ A \subseteq \{x_1, x_2, \cdots, x_n\} \mid f|_{x_1=0, x_i\in A} \equiv 0 \right\}
\]

(12)

If \(f(x_1, x_2, \cdots, x_n)\) is an increasing Boolean function, define that

\[
\Delta_f = \left\{ A \subseteq \{x_1, x_2, \cdots, x_n\} \mid f|_{x_i=1, x_i\in A} \equiv 0 \right\}
\]

(13)

**Theorem 1** \(\Delta_f\) is a complex.

**Proof** Assume that \(f(x_1, x_2, \cdots, x_n)\) is a decreasing Boolean function.

(1) Take \(A \in \Delta_f\) wantonly, assume that \(B \subseteq A\). Assume the corresponding variable value to \(A\) is \(\{x_1, x_2, \cdots, x_n\}_A\), the corresponding variable value to \(B\) is \(\{x_1, x_2, \cdots, x_n\}_B\). According to Definition 2, we can learn that,

\[
\{x_1, x_2, \cdots, x_n\}_A \leq \{x_1, x_2, \cdots, x_n\}_B
\]

(14)

As \(f(x_1, x_2, \cdots, x_n)\) is a decreasing Boolean function, so

\[
f(x_1, x_2, \cdots, x_n)_A \geq f(x_1, x_2, \cdots, x_n)_B
\]

(15)

Also because \(A \in \Delta_f\), in other words, \(f(x_1, x_2, \cdots, x_n)_A = 0\), so \(f(x_1, x_2, \cdots, x_n)_B = 0\). Thus \(B \in \Delta_f\).

(2) \(f(x_1, x_2, \cdots, x_n)\) is a function with \(n\) variables, it has \(2^n\) kinds of assignments, so we define that \(\Delta_f\) includes a limited number of finite set.
In a word, $\Delta_f$ is a complex.

**Theorem 2** If $\chi(\Delta_f) \neq 1$, then $f(x_1, x_2, \ldots, x_n)$ is elusive.

**Proof** Assume that $f(x_1, x_2, \ldots, x_n)$ is an increasing Boolean function. According to Definition 2, 
\[
\Delta_f = \left\{ A \subseteq \{x_1, x_2, \ldots, x_n\} \mid f\big|_{x_i = 0, i \in A} = 0 \right\}.
\]

(16)

Its Euler characteristic is
\[
\chi(\Delta_f) = \sum_{A \in \Delta_f} (-1)^{\dim(A)}
\]
\[
= \sum_{x \in \mathcal{P}^n} \overline{f}(x)(-1)^{\text{char}(x)}
\]
\[
= \sum_{x \in \mathcal{P}^n} \overline{f}(x)(-1)^{\text{char}(x)} + 1
\]
\[
= - \sum_{x \in \mathcal{P}^n} \overline{f}(x)(-1)^{\text{char}(x)} + 1
\]
\[
= -q_f(-1) + 1.
\]

(17)

According to Lemma 1, we can learn that if $q_f(-1) \neq 0$, then $f(x_1, x_2, \ldots, x_n)$ is elusive. If $\chi(\Delta_f) \neq 1$, then $f(x_1, x_2, \ldots, x_n)$ is an elusive function.

Theorem 2 is to judge the function’s elusive by calculating the function’s complex, it is obvious that this method is much more simple than Lemma 1. For example, for increasing Boolean function
\[
f(x_1, x_2, x_3, x_4) = x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1,
\]

According to Definition 2,
\[
\Delta_f = \left\{ A \subseteq \{x_1, x_2, \ldots, x_n\} \mid f\big|_{x_i = 0, i \in A} = 0 \right\} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \emptyset\}.
\]

(18)

All graph properties are Boolean functions, so Theorem 2 can also be used to judge graph property’s elusive. Table 1 shows the connectivity of 3 vertex graphs:

| $x_{12}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|---------|---|---|---|---|---|---|---|---|
| $x_{13}$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_{23}$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $f(x_{12}, x_{13}, x_{23})$ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

Table 1. Connectivity of 3 vertex graphs.

The graph’s connectivity is an increasing Boolean function, according to Definition 2, we can get
\[ \Delta_f = \left\{ A \subseteq \{x_1, x_2, \ldots, x_n\} \mid \sum_{x \in A} f(x) \equiv 0 \right\} = \{\{x_1\}, \{x_2\}, \{x_3\}, \emptyset\}. \]  

Its Euler characteristic is \( \chi(\Delta_f) = 3 \neq 1 \), so the connectivity of 3 vertex graphs is elusive.

Summary
This article makes a further discussion on Boolean function’s false-assignment, and thus gets the theory basis on how to judge the function’s elusive.

Acknowledgement
This research was financially supported by the Dalian Naval Academy for Basic Research.

References
[1] Xie Zheng-long, Xu Qing-lin, Shi Fang-xia, A refinement algorithm of decision tree generation. Journal of Xianyang Normal University, 2006, 21(6): 31-33.

[2] Wang Pei-gen, Boole function and Boole polynomial, Journal of Capital Normal University, 2006, 27(5): 15-18, 21.

[3] Chen Guo-zhang, He Pei-lian, Chen Min, Algorithm for Boolean function complementation stochastic replenishment time, Journal of Tianjin University, 2001, 34(4), 447-451.

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

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

[6] 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.

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

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

[9] 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.

[10] 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.