ON THE COMPLEXITY OF THE EVALUATION OF TRANSIENT EXTENSIONS OF BOOLEAN FUNCTIONS

JANUSZ BRZOZOWSKI, BAIYU LI

David R. Cheriton School of Computer Science,
University of Waterloo, Waterloo, ON, Canada N2L 3G1
{*brzozo,b5li*}*@uwaterloo.ca*

YULI YE

Department of Computer Science
University of Toronto, Toronto, ON, Canada M5S 3G4
y3ye@cs.toronto.edu

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

Transient algebra is a multi-valued algebra for hazard detection in gate circuits. Sequences of alternating 0's and 1's, called transients, represent signal values, and gates are modeled by extensions of boolean functions to transients. Formulas for computing the output transient of a gate from the input transients are known for NOT, AND, OR and XOR gates and their complements, but, in general, even the problem of deciding whether the length of the output transient exceeds a given bound is NP-complete. We propose a method of evaluating extensions of general boolean functions. We study a class of functions for which, instead of evaluating the extensions on a given set of transients, it is possible to get the same values by using transients derived from the given ones, but having length at most 3. We prove that all functions of three variables, as well as certain other functions, have this property, and can be efficiently evaluated.

Keywords: algebra, boolean function, circuit, complexity, evaluation, gate, hazard, multi-valued, transient, transient extension

1991 Mathematics Subject Classification: 68Q17, 68Q19, 68R01

1. Introduction

In 2003 Brzozowski and Ésik [2] proposed an infinite algebra as a basis for a theory of hazards in gate circuits. The fundamental concept in this theory is that of a "transient", which is a nonempty alternating sequence of 0's and 1's representing a series of signal values. Boolean functions that are normally used to model gates are extended to transients. Given a boolean function $f(x_1, \ldots, x_n)$, and n transients $\mathbf{x_1}, \ldots, \mathbf{x_n}$, the extension $\mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_n})$ of f to transients is defined as the longest possible transient that can be obtained by considering all possible orders of changes of the input variables. While this definition is straightforward, it involves the construction of an n-dimensional directed graph in which all possible orders of input

changes are displayed. The size of this graph is exponential in the length of the input transients, and so this method is inefficient.

There exist simple formulas [2] for common boolean functions like NOT, AND, OR and XOR, and such formulas have been extended to functions obtained from the set {OR, XOR} by complementing any number of inputs, and/or the output [3]. However, function composition does not preserve extensions [2], and the evaluation problem remains open for general boolean functions. We study extensions of general functions, and propose ways of evaluating them. In particular, we introduce a method in which an arbitrary vector $\mathbf{x}$ of transients is replaced by a vector $\tilde{\mathbf{x}}$ of "characteristic transients" which are of length at most 3. We show that evaluating $\mathbf{f}(\mathbf{x})$ can be reduced to evaluating $\mathbf{f}(\tilde{\mathbf{x}})$ for some functions. This makes it possible to efficiently evaluate all the functions of three variables, and certain other functions.

The remainder of the paper is structured as follows. Transients, vectors of transients, and extensions of boolean functions are defined in Section 2. The evaluation of functions in a certain class $\mathcal{G}$ is considered in Section 3, where we define the "cost" of a transient vector $\mathbf{x}$ to be the difference between the number of changes in $\mathbf{x}$ and the number of changes in $\mathbf{f}(\mathbf{x})$. The concept of cost is extended to paths in digraphs and walks in boolean cubes in Section 4. Characteristic vectors are defined in Section 5. In Section 6 we prove that all 3-variable functions can be efficiently evaluated using characteristic vectors, and that there exists a 5-variable function that cannot be so evaluated. Section 7 concludes the paper.

2. Transients, Vectors, and Extensions of Functions

The cardinality of a set S is $|S|$. For $n \geq 1$, let $[n] = \{1, \ldots, n\}$. If A is an alphabet, then A^* (A^+) denotes the free monoid (free semigroup) generated by A. The length of a word $w \in A^*$ is $l(w)$, and the first and last letters of $w \in A^+$ are $\alpha(w)$ and $\omega(w)$, respectively. For boolean operations, we use x' for complement, xy for AND, $x + y$ for OR, and $x \oplus y$ for XOR (exclusive OR).

Let $B = \{0, 1\}$; a *binary word* is any word in B^*. A *transient* is a binary word in B^+ of alternating 0's and 1's; thus the set $\mathbf{T}$ of all transients is $0(10)^* + (01)^*01 + (10)^*10 + (10)^*1$, in regular-expression notation. Transients are denoted by boldface letters. A transient can be obtained from any nonempty binary word by *contraction*, *i.e.*, elimination of all duplicates immediately following a symbol; thus contraction is a function from B^+ to $\mathbf{T}$. We denote the contraction of a word w by $\overleftrightarrow{w}$. For example, $\overleftrightarrow{001000} = 010$. For $\mathbf{s}, \mathbf{t} \in \mathbf{T}$, $\mathbf{s} \circ \mathbf{t}$ is concatenation followed by contraction, *i.e.*, $\mathbf{s} \circ \mathbf{t} = \overleftrightarrow{\mathbf{st}}$, where $\mathbf{s}, \mathbf{t} \in \mathbf{T}$. The $\circ$ operation is associative.

If $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$ is a transient, $t_i \in B$ for $i \in [m]$, then $\Delta(\mathbf{t}) = \mathbf{l}(\mathbf{t}) - \mathbf{1} = \mathbf{m} - \mathbf{1}$ is the *number of changes* in $\mathbf{t}$. A transient $\mathbf{t}$ is completely determined by its beginning $\alpha(\mathbf{t})$ and the number of changes $\Delta(\mathbf{t})$; thus we have the representation of $\mathbf{t}$ which we indicate by angle brackets: $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m} = \langle \alpha(\mathbf{t}); \boldsymbol{\Delta}(\mathbf{t}) \rangle = \langle \mathbf{t_1}; \mathbf{m} - \mathbf{1} \rangle$. The number of 0's in a transient $\mathbf{t}$ is $z(\mathbf{t})$, and the number of 1's ("units") is $u(\mathbf{t})$.

A *prefix* of a transient $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$ is any transient $\mathbf{u} = \mathbf{t_1} \cdots \mathbf{t_i}$, where $i \in [m]$.

A *suffix* of a transient is defined similarly. The empty word is not a prefix or suffix, because it is not a transient. However, $\mathbf{t}$ *is* a prefix and suffix of itself. If $\mathbf{u}$ is a prefix of $\mathbf{t}$ and $l(\mathbf{u}) < \mathbf{l}(\mathbf{t})$, then there exists a transient $\mathbf{v}$, a suffix of $\mathbf{t}$, such that $\mathbf{t} = \mathbf{uv}$. A transient $\mathbf{s}$ is the *successor* of a transient $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$ if and only if $\mathbf{s} = \mathbf{t_1} \cdots \mathbf{t_m t_{m+1}}$, where $t_i \in B$, for $i \in [m+1]$, *e.g.*, 010 is the successor of 01.

A *transient vector,* or simply a *vector,* is a tuple $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$. By convention, if $\mathbf{x}$ is a vector, then $\mathbf{x_i}$ is a component of $\mathbf{x}$. The $\circ$ operation is extended to transient vectors component-wise. The *length* of a vector $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ is $l(\mathbf{x}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{l}(\mathbf{x_i})$. The *number of changes* of $\mathbf{x}$ is $\Delta(\mathbf{x}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\Delta}(\mathbf{x_i}) = \mathbf{l}(\mathbf{x}) - \mathbf{n}$. We also define vectors $\alpha(\mathbf{x}) = (\alpha(\mathbf{x_1}), \ldots, \alpha(\mathbf{x_n}))$, and $\omega(\mathbf{x}) = (\omega(\mathbf{x_1}), \ldots, \omega(\mathbf{x_n}))$. A vector is completely determined by its beginning $\alpha(\mathbf{x})$ and the number of changes $\Delta(\mathbf{x_i})$ of each component of $\mathbf{x}$; thus we have the representation $\mathbf{x} = \langle \alpha(\mathbf{x}); \mathbf{\Delta}(\mathbf{x_1}), \ldots, \mathbf{\Delta}(\mathbf{x_n}) \rangle$. A vector $\mathbf{u} = (\mathbf{u_1}, \ldots, \mathbf{u_n})$ is a *prefix* (*suffix*) of vector $\mathbf{v} = (\mathbf{v_1}, \ldots, \mathbf{v_n})$ if $\mathbf{u_i}$ is a prefix (suffix) of $\mathbf{v_i}$ for all $i \in [n]$. A vector $\mathbf{u} = (\mathbf{u_1}, \ldots, \mathbf{u_n})$ is a *successor* of $\mathbf{v} = (\mathbf{v_1}, \ldots, \mathbf{v_n})$ if $\mathbf{u_i}$ is the successor of $\mathbf{v_i}$ for some $i \in [n]$ and $\mathbf{u_j} = \mathbf{v_j}$, for all $j \neq i$.

Our terminology on graphs is from [1]. If $f : B^n \to B$ is a boolean function and $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$ is a vector, we construct the *transient digraph* $D = D_f(\mathbf{x}) = (\mathbf{V}, \mathbf{E}, \psi, \lambda)$ of f for $\mathbf{x}$, where (V, E, ψ) is a digraph, V (the set of *vertices*) is the set of all prefixes of $\mathbf{x}$, E (the set of *arcs*) is $E = \{e = (\mathbf{u}, \mathbf{v}) \mid \mathbf{v}$ is a successor of $\mathbf{u}\}$, ψ (the *incidence function* assigning to each arc of D an ordered pair of vertices of D) is $\psi(e) = \psi((\mathbf{u}, \mathbf{v})) = (\mathbf{u}, \mathbf{v})$, and $\lambda : V \to B$ is the *output function* assigning the value $f(\omega(\mathbf{v}))$ to every $\mathbf{v} \in \mathbf{V}$. Each directed path $P = \mathbf{v_1}, \ldots, \mathbf{v_m}$ in D from $\mathbf{v_1} = \alpha(\mathbf{x})$ to $\mathbf{v_m} = \mathbf{x}$ has length $m = \sum_{i=1}^{n} \Delta(\mathbf{x_i})$. We extend λ to paths: $\lambda(P) = \lambda(\mathbf{v^1}) \cdots \lambda(\mathbf{v^m})$. Paths are always from $\alpha(\mathbf{x})$ to $\mathbf{x}$.

Definition 1. *Let $f(x) : B^n \to B$ be a boolean function. The* transient extension *(or simply* extension*) of f is a function $\mathbf{f}(\mathbf{x}) : \mathbf{T^n} \to \mathbf{T}$, such that for any $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$, $\mathbf{f}(\mathbf{x}) = \overleftrightarrow{\lambda(P)}$, where P is a path in $D_f(\mathbf{x})$ and $\overleftrightarrow{\lambda(P)}$ is of maximal length; we call such a path P* optimal.

Example 2. *In Fig. 1 we show the digraph $D_f(010, 1010)$ for $f = x_1 + x_2'$, where changes in x_1 are horizontal, and in x_2, vertical. The initial vertex is $\alpha(010, 1010) = (0, 1)$. If the inputs are changed in the order x_2, x_2, x_1, x_2, x_1 (path $\mathbf{v^1}, \mathbf{v^2}, \ldots, \mathbf{v^6}$), then the binary word defined by λ is* 010111*, and its contraction is* 0101*. The longest output is $(01)^3$ (using the order x_1, x_1, x_2, x_2, x_2), and so $\mathbf{f}(\mathbf{010}, \mathbf{1010}) = (\mathbf{01})^{\mathbf{3}}$.*

For a vector $\mathbf{x} \in \mathbf{T^n}$, let $\varphi(\mathbf{x})$ be the number of paths in $D_f(\mathbf{x})$, let $m = \Delta(\mathbf{x})$, and let $m_i = \Delta(\mathbf{x_i})$ for $i \in [n]$. Then $\varphi(\mathbf{x})$ is the multinomial coefficient:

$$\varphi(\mathbf{x}) = \binom{\mathbf{m}}{\mathbf{m_1}, \ldots, \mathbf{m_n}} = \frac{\mathbf{m!}}{\mathbf{m_1!} \cdots \mathbf{m_n!}}; \tag{1}$$

The maximal value of $\varphi(\mathbf{x})$ has the following approximation [6]:

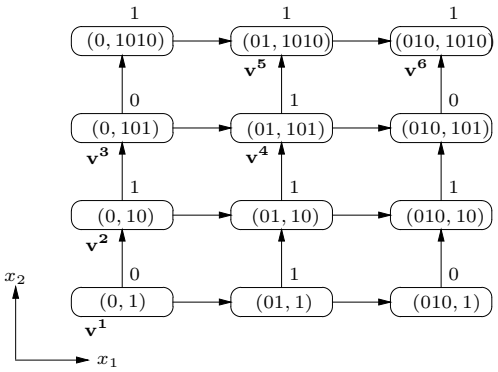


Fig. 1. Digraph $D_f(010, 1010)$ for $f = x_1 + x_2'$.

$$\varphi(\mathbf{x}) \approx (\mathbf{2}\pi\mathbf{m})^{\frac{\mathbf{1}-\mathbf{n}}{\mathbf{2}}}\mathbf{n}^{\mathbf{m}+\frac{\mathbf{n}}{\mathbf{2}}}. \tag{2}$$

We usually consider n to be small and fixed; then $\varphi(\mathbf{x})$ is exponential in m. So the obvious way to evaluate $\mathbf{f}(\mathbf{x})$ is not feasible because of the large number of paths.

3. Functions in the Class $\mathcal{G}$

In contrast to the general case above, there are simple formulas for NOT, XOR, OR and AND [2]: If $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$, then

$$\mathbf{t}' = (\mathbf{t_1} \cdots \mathbf{t_m})' = \mathbf{t}_\mathbf{1}' \cdots \mathbf{t}_\mathbf{m}'. \tag{3}$$

If $f(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$ is XOR, then, for all $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$,

$$\alpha(\mathbf{f}(\mathbf{x})) = \alpha(\mathbf{x_1}) \oplus \cdots \oplus \alpha(\mathbf{x_n}), \quad \omega(\mathbf{f}(\mathbf{x})) = \omega(\mathbf{x_1}) \oplus \cdots \oplus \omega(\mathbf{x_n}), \tag{4}$$

$$l(\mathbf{f}(\mathbf{x})) = 1 + \sum_{i=1}^{n} (l(\mathbf{x_i}) - \mathbf{1}). \tag{5}$$

If $f(x_1, \ldots, x_n) = x_1 + \cdots + x_n$ is OR, then, for all $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$,

$$\alpha(\mathbf{f}(\mathbf{x})) = \alpha(\mathbf{x_1}) + \cdots + \alpha(\mathbf{x_n}), \quad \omega(\mathbf{f}(\mathbf{x})) = \omega(\mathbf{x_1}) + \cdots + \omega(\mathbf{x_n}), \tag{6}$$

$$z(\mathbf{f}(\mathbf{x})) = \begin{cases} 0, & \text{if } \exists i \in [n]\ \mathbf{x_i} = \mathbf{1}; \\ 1 + \sum_{i=1}^{n} (z(\mathbf{x_i}) - \mathbf{1}), & \text{otherwise.} \end{cases} \tag{7}$$

If $f(x_1, \ldots, x_n) = x_1 \cdots x_n$ is AND, then, for all $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n}) \in \mathbf{T^n}$,

$$\alpha(\mathbf{f}(\mathbf{x})) = \alpha(\mathbf{x_1}) \cdots \alpha(\mathbf{x_n}), \quad \omega(\mathbf{f}(\mathbf{x})) = \omega(\mathbf{x_1}) \cdots \omega(\mathbf{x_n}), \tag{8}$$

$$u(\mathbf{f}(\mathbf{x})) = \begin{cases} 0, & \text{if } \exists i \in [n]\ \mathbf{x_i} = \mathbf{0}; \\ 1 + \sum_{i=1}^{n} (u(\mathbf{x_i}) - \mathbf{1}), & \text{otherwise.} \end{cases} \tag{9}$$

Using these formulas we can evaluate transient extensions of NOT, XOR, OR and AND in the time linear in the length of the input vector. For example, to evaluate OR for $\mathbf{x} \in \mathbf{T^n}$, we compute $\alpha(\mathbf{f}(\mathbf{x}))$, $\omega(\mathbf{f}(\mathbf{x}))$, and the number of 0's in $\mathbf{x}$.

The class $\mathcal{G}$ of boolean functions was defined in [3] as follows:

Definition 3. *Let $\mathcal{G} = \mathcal{H} \cup \tilde{\mathcal{H}}$, where $\mathcal{H} = \{$OR, XOR$\}$, $\tilde{\mathcal{H}}$ is the set of functions obtained by complementing any number of inputs and/or the output of functions from $\mathcal{H}$, and* OR *and* XOR *have an arbitrary number of inputs (including one).*

Note that a 1-input OR or XOR function is the identity function, and that $\mathcal{G}$ includes all the boolean functions of two variables, except the constants 0 and 1, as well as AND, NOR, NAND and XNOR functions with any numbers of inputs.

It was proved in [3] that functions in $\mathcal{G}$ can be evaluated by complementing the input transients of any complemented arguments, and by complementing the output transient, if the function itself is complemented. Consequently, we have

Proposition 4. *Functions in $\mathcal{G}$ can be evaluated in the time linear in the length of the input vector.*

Example 5. *If $f(x_1, x_2) = (x_1 + x_2')'$, then $\mathbf{f(010, 10)} = (\mathbf{010} + (\mathbf{10})')' = (\mathbf{010} + \mathbf{01})' = (\mathbf{0101})' = \mathbf{1010}$. However, in general, function composition does not preserve extensions* [2]*. For example, by (4) and (5), $01 \oplus 101 = 1010$, but if we express $\mathbf{s} \oplus \mathbf{t}$ as $\mathbf{st'} + \mathbf{s't}$, we get 101010. So we need to consider functions that are not in $\mathcal{G}$.*

As we have seen, evaluating a transient extension from the transient digraph is not efficient. In [2] it is shown that even the problem of estimating the length of $\mathbf{f}(\mathbf{x})$ is NP-complete. However, the concept of "cost" that we are about to define makes the calculation feasible for some functions.

Definition 6. *Let $f : B^n \to B$ be a boolean function, and $\mathbf{f} : \mathbf{T^n} \to \mathbf{T}$, its extension. Let $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_2}) = \langle \alpha(\mathbf{x}); \mathbf{\Delta}(\mathbf{x_1}), \ldots, \mathbf{\Delta}(\mathbf{x_n}) \rangle$ be a transient vector. The* cost of $\mathbf{x}$ for f *is $c_f(\mathbf{x}) = \mathbf{\Delta}(\mathbf{x}) - \mathbf{\Delta}(\mathbf{f}(\mathbf{x})) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\Delta}(\mathbf{x_i}) - \mathbf{\Delta}(\mathbf{f}(\mathbf{x}))$.*

The following upper bound for $l(\mathbf{f}(\mathbf{x}))$ is given in [2]: $l(\mathbf{f}(\mathbf{x})) \leq \mathbf{1} + \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{l}(\mathbf{x_i}) - \mathbf{1}) = \mathbf{1} + \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\Delta}(\mathbf{x_i}) = \mathbf{1} + \mathbf{\Delta}(\mathbf{x})$. Thus $\Delta(\mathbf{f}(\mathbf{x})) = \mathbf{l}(\mathbf{f}(\mathbf{x})) - \mathbf{1} \leq \mathbf{\Delta}(\mathbf{x})$, and $c_f(\mathbf{x})$ is a non-negative integer. If we know $\mathbf{x}$ and its cost $c_f(\mathbf{x})$, then we can easily evaluate $\mathbf{f}(\mathbf{x})$ as follows: $\mathbf{f}(\mathbf{x}) = \langle \alpha(\mathbf{f}(\mathbf{x})); \mathbf{\Delta}(\mathbf{x}) - \mathbf{c_f}(\mathbf{x}) \rangle$.

A binary vector $(x_1, \ldots, x_n)$ with $x_i = 0$, for all $i \in [n]$ is denoted 0^n. We define *non-negative subtraction* $m \ominus n$ of integer n from integer m as $m \ominus n = m - n$ if $m \geq n$, and $m \ominus n = 0$, otherwise. A transient $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$ is *proper* if its length is at least 2, *i.e.*, if it contains at least one change. A vector is *proper* if all of its components are proper.

Theorem 7. *If $\mathbf{x}$ is proper, then*

$$c_{xor}(\mathbf{x}) = \mathbf{0}, \tag{10}$$

$$c_{or}(\mathbf{x}) = (\mathbf{u}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1}), \tag{11}$$

$$c_{and}(\mathbf{x}) = (\mathbf{z}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{z}(\omega(\mathbf{x})) \ominus \mathbf{1}). \tag{12}$$

Proof. If f is XOR, we have $c_{xor} = \Delta(\mathbf{x}) - \mathbf{\Delta}(\mathbf{f(x)}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\Delta}(\mathbf{x_i}) - \mathbf{\Delta}(\mathbf{f(x)}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{l(x_i)} - \mathbf{1}) - (\mathbf{l(x)} - \mathbf{1}) = \mathbf{0}$, where we have used Equation (5).

If f is OR, we consider three cases:

(1) $u(\alpha(\mathbf{x})) = \mathbf{0}$. We have $u(\alpha(\mathbf{x})) \ominus \mathbf{1} = \mathbf{0}$. Since $\alpha(\mathbf{x_i}) = \mathbf{0}$ for all $i \in [n]$, we have $\alpha(f(\mathbf{x})) = \mathbf{0}$. Let $S = \{i \mid \omega(\mathbf{x_i}) = \mathbf{1}\}$, $T = \{i \mid \omega(\mathbf{x_i}) = \mathbf{0}\}$. Then

$$\begin{aligned}\Delta(\mathbf{x}) &= \textstyle\sum_{i=1}^{n}(l(\mathbf{x_i}) - \mathbf{1}) = \sum_{\mathbf{i}\in\mathbf{S}}(\mathbf{l(x_i)} - \mathbf{1}) + \sum_{\mathbf{i}\in\mathbf{T}}(\mathbf{l(x_i)} - \mathbf{1}) \\ &= \textstyle\sum_{i\in S}(2z(\mathbf{x_i}) - \mathbf{1}) + \sum_{\mathbf{i}\in\mathbf{T}}(\mathbf{2z(x_i)} - \mathbf{2}) \\ &= \textstyle\sum_{i=1}^{n}(2z(\mathbf{x_i}) - \mathbf{2}) + |\mathbf{S}| = \mathbf{2(z(f(x))} - \mathbf{1}) + \mathbf{u}(\omega(\mathbf{x})),\end{aligned}$$

where the last equality uses Equation (7). If $u(\omega(\mathbf{x})) = \mathbf{0}$, then $\omega(\mathbf{x_i}) = \mathbf{0}$ for all $i \in [n]$, and $\omega(f(\mathbf{x})) = \mathbf{0}$. Thus we have $l(f(\mathbf{x})) = \mathbf{2z(f(x))} - \mathbf{1}$, and $\Delta(\mathbf{x}) = \mathbf{l(f(x))} - \mathbf{1} = \mathbf{\Delta(f(x))}$. Then $c_{or}(\mathbf{x}) = \mathbf{\Delta(x)} - \mathbf{\Delta(f(x))} = \mathbf{0} = (\mathbf{u}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1})$. Otherwise, $u(\omega(\mathbf{x})) \geq \mathbf{1}$. Then $\omega(f(\mathbf{x})) = \mathbf{1}$, and $l(f(\mathbf{x})) = \mathbf{2z(f(x))}$. Thus $\Delta(\mathbf{x}) = \mathbf{2(z(f(x))} - \mathbf{1}) + \mathbf{u}(\omega(\mathbf{x})) = \mathbf{l(f(x))} - \mathbf{1} + \mathbf{u}(\omega(\mathbf{x})) - \mathbf{1}$ $= \mathbf{l(f(x))} - \mathbf{1} + \mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1} = \mathbf{\Delta(f(x))} + \mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1}$.
Hence, $c_{or}(\mathbf{x}) = \mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1} = (\mathbf{u}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1})$.

(2) $u(\omega(\mathbf{x})) = \mathbf{0}$. This case is symmetric to Case 1.

(3) $u(\alpha(\mathbf{x})) \neq \mathbf{0}$, and $u(\omega(\mathbf{x})) \neq \mathbf{0}$. Since $\mathbf{x}$ is proper, then for all $i \in [n]$, we have $l(\mathbf{x_i}) \geq \mathbf{2}$, and there is at least one 0 in $\mathbf{x_i}$. Let $\mathbf{x_i} = \mathbf{v_i} \circ \mathbf{u_i}$, where $\omega(\mathbf{v_i}) = \alpha(\mathbf{u_i}) = \mathbf{0}$, $\mathbf{v} = (\mathbf{v_1}, \ldots, \mathbf{v_n})$, and $\mathbf{u} = (\mathbf{u_1}, \ldots, \mathbf{u_n})$. Then $\mathbf{x} = \mathbf{v} \circ \mathbf{u}$, and $\omega(\mathbf{v}) = \alpha(\mathbf{u}) = (\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0^n}$. By Cases 1 and 2, $c_{or}(\mathbf{v}) = \mathbf{u}(\alpha(\mathbf{x})) \ominus \mathbf{1}$, and $\mathbf{c_{or}}(\mathbf{u}) = \mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1}$. Therefore, $c_{or}(\mathbf{x}) = \mathbf{c_{or}}(\mathbf{v}) + \mathbf{c_{or}}(\mathbf{u}) = (\mathbf{u}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{u}(\omega(\mathbf{x})) \ominus \mathbf{1})$.

By AND/OR duality, we have $c_{and}(\mathbf{x}) = (\mathbf{z}(\alpha(\mathbf{x})) \ominus \mathbf{1}) + (\mathbf{z}(\omega(\mathbf{x})) \ominus \mathbf{1})$. □

4. Costs of Paths in Digraphs and Walks in Cubes

For $n \geq 1$, the *boolean n-cube* is a graph $C^n = (V, E, \psi)$, where $V = B^n$ (*vertices*), $E = \{e = (v^i, v^j) \mid v^i, v^j \in V$ and v^i and v^j differ in exactly one coordinate$\}$ (*edges*), and $\psi(e) = \psi((v^i, v^j)) = (v^i, v^j)$ (*incidence function*). For a boolean function $f : B^n \to B$, $n \geq 1$, the *cube* C_f^n of f is the n-cube where $f(v)$ is assigned to each vertex $v \in V = B^n$. If n is understood, we denote C_f^n by C_f.

In a function cube C_f, an edge $e = (v^i, v^j) \in E$ is *live* if $f(v^i) \neq f(v^j)$; otherwise it is *dead.* A *live graph* of f is that subgraph L_f of C_f that consists of all the live edges and their incident vertices.

Instead of considering paths in a digraph $D_f(\mathbf{x})$, we will examine walks in the cube C_f. The size of a digraph $D_f(\mathbf{x})$ increases as the length of $\mathbf{x}$ increases, and the length of any path in $D_f(\mathbf{x})$ increases accordingly. However, the size of a cube C_f is independent of any vector $\mathbf{x}$, if the dimension of $\mathbf{x}$ is fixed.

Example 8. *The cube of $f = x_1 + x_2'$ is shown in Fig. 2 (a), where a vertex is white if $f(v^i) = 0$, and black otherwise, and 00 stands for $(0, 0)$, etc. The live edges of f are shown by thick lines in Fig. 2 (b). The live graph of f is shown in Fig. 2 (c).*

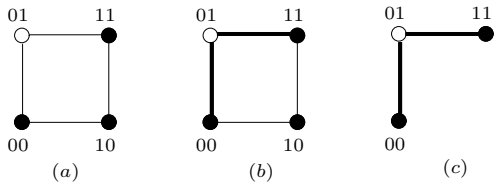


Fig. 2. Graphs for $f = x_1 + x_2'$: (a) C_f; (b) live edges; (c) L_f.

We extend the concept of cost to paths in digraphs and walks in cubes. For a path $P = \mathbf{v^1}, \ldots, \mathbf{v^m}$ in $D_f(\mathbf{x})$, let $c(P) = |E_=(P)|$, where $E_=(P) = \{(\mathbf{v^i}, \mathbf{v^{i+1}}) \mid \lambda(\mathbf{v^i}) = \lambda(\mathbf{v^{i+1}}),\ \mathbf{i = 1}, \ldots, \mathbf{m-1}\}$. Thus $c(P)$ is the number of arcs in P whose endpoints have the same λ values. For any walk $W = w^1, w^2, \cdots, w^m$ in C_f, let $c(W) = |E_=(W)|$, where $E_=(W) = \{(w^j, w^{j+1}) \mid f(w^j) = f(w^{j+1}),\ j = 1, \ldots, m-1\}$. Thus $c(W)$ is the number of edges in W whose endpoints have the same f values, that is, the number of edges in W which are not in the live graph L_f.

Definition 9. *Let $f : B^n \to B$ be a boolean function, and $\mathbf{f}$, its extension. Let $\mathbf{x} \in \mathbf{T^n}$ be a vector, and $P = \mathbf{v^1}, \ldots, \mathbf{v^r}$, a path in $D_f(\mathbf{x})$ from $\mathbf{v^1} = \alpha(\mathbf{x})$ to $\mathbf{v^r} = \mathbf{x}$. Let $W(P)$ be the sequence $W(P) = \omega(\mathbf{v^1}), \ldots, \omega(\mathbf{v^r})$. Conversely, let $W = w^1, \ldots, w^r$ be any walk in C_f, where $w^i \in B^n$, for $i = 1, \ldots, r$. Define $\mathbf{x^i} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^i}$, for $i = 1, \ldots, r$ and let $P(W)$ be the sequence $P(W) = \mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^r} = \mathbf{w^1}, \mathbf{w^1} \circ \mathbf{w^2}, \ldots, \mathbf{w^1} \circ \cdots \circ \mathbf{w^r}$.*

Theorem 10. *If P is a path in $D_f(\mathbf{x})$ then $W(P)$ is a walk in C_f and $c(P) = c(W(P))$. If $W = w^1, \ldots, w^r$ is a walk in C_f, let $\mathbf{x} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^r}$. Then $P(W)$ is a path in $D_f(\mathbf{x})$ and $c(W) = c(P(W))$. Moreover, if P is a path in $D_f(\mathbf{x})$, then $P(W(P)) = P$, and if W is a walk in C_f, then $W(P(W)) = W$.*

Proof. If $W(P) = \omega(\mathbf{v^1}), \ldots, \omega(\mathbf{v^r})$, then $\omega(\mathbf{v^i})$ is a vertex in C_f. Since every pair $(\mathbf{v^i}, \mathbf{v^{i+1}})$, $i = 1, \ldots, r-1$, is an arc in $D_f(\mathbf{x})$, $\mathbf{v^{i+1}}$ is a successor of $\mathbf{v^i}$. Thus $\omega(\mathbf{v^{i+1}})$ and $\omega(\mathbf{v^i})$ differ in exactly one coordinate, $(\omega(\mathbf{v^i}), \omega(\mathbf{v^{i+1}}))$ is an edge in C_f, and $W(P)$ is indeed a walk in C_f. By definition, $\lambda(\mathbf{v^i}) = \mathbf{f}(\omega(\mathbf{v^i}))$ for $i = 1, \ldots, r$. The endpoints of an arc $(\mathbf{v^i}, \mathbf{v^{i+1}})$ in $D_f(\mathbf{x})$ have the same λ value if and only if the endpoints of the edge $(\omega(\mathbf{v^i}), \omega(\mathbf{v^{i+1}}))$ have the same f value. So $c(P) = c(W(P))$.

Let $P(W) = \mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^r} = \mathbf{w^1}, \mathbf{w^1} \circ \mathbf{w^2}, \ldots, \mathbf{w^1} \circ \cdots \circ \mathbf{w^r}$. Clearly $w^1 = \mathbf{x^1} = \alpha(\mathbf{x})$ is a vertex in $D_f(\mathbf{x})$. Suppose $\mathbf{x^i} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^i} = (\mathbf{x_1^i}, \ldots, \mathbf{x_n^i})$ is a vertex in $D_f(\mathbf{x})$. Let $\mathbf{x^{i+1}} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^i} \circ \mathbf{w^{i+1}} = (\mathbf{x_1^{i+1}}, \ldots, \mathbf{x_n^{i+1}})$. Since (w^i, w^{i+1}) is an edge in C_f, w^i and w^{i+1} differ in exactly one coordinate, say the k-th. Then, $\mathbf{x_j^{i+1}} = \mathbf{x_j^i} \circ \mathbf{w_j^{i+1}} = \mathbf{x_j^i}$ if $j \neq k$, and $\mathbf{x_k^{i+1}}$ is a successor of $\mathbf{x_k^i}$. Hence $\mathbf{x^{i+1}}$ is a successor of $\mathbf{x^i}$; thus, by induction on r, $(\mathbf{x^i}, \mathbf{x^{i+1}})$ is an arc in $D_f(\mathbf{x})$. Therefore, $P(W)$ is a path in $D_f(\mathbf{x})$ from w^1 to $\mathbf{x}$. Since $\lambda(\mathbf{x^i}) = \mathbf{f}(\omega(\mathbf{x^i})) = \mathbf{f}(\mathbf{w^i})$, for $i = 1, \ldots, r$, the endpoints of (w^i, w^{i+1}) have the same f value if and only if the endpoints of $(\mathbf{x^i}, \mathbf{x^{i+1}})$ have the same λ value. Therefore, $c(W) = c(P(W))$.

Now suppose $P = \mathbf{v^1}, \ldots, \mathbf{v^r}$. By Definition 9, $W(P) = \omega(\mathbf{v^1}), \ldots, \omega(\mathbf{v^r})$. Let $\mathbf{x^i} = \omega(\mathbf{v^1}) \circ \cdots \circ \omega(\mathbf{v^i})$, for $i = 1, \ldots, r$; then $\mathbf{x^1} = \omega(\mathbf{v^1}) = \mathbf{v^1}$. Suppose $\mathbf{v^i} = \mathbf{x^i}$ for some $1 \leq i < r$. Then, since $(\mathbf{v^i}, \mathbf{v^{i+1}})$ is an arc in $D_f(\mathbf{x})$, $\mathbf{v^{i+1}}$ is a successor of $\mathbf{v^i}$; so $\mathbf{v^{i+1}} = \mathbf{v^i} \circ \omega(\mathbf{v^{i+1}}) = \mathbf{x^i} \circ \omega(\mathbf{v^{i+1}}) = \mathbf{x^{i+1}}$. Thus, by induction on r, $\mathbf{v^i} = \mathbf{x^i}$ for $i = 1, \ldots, r$. Therefore, $P(W(P)) = P$.

Finally, suppose $W = w^1, \ldots, w^r$. By Definition 9, $P(W) = \mathbf{x^1}, \ldots, \mathbf{x^r}$, where $\mathbf{x^i} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^i}$ for $i = 1, \ldots, r$. Since $\omega(\mathbf{x^i}) = \mathbf{w^i}$ for $i = 1, \ldots, r$, we have $W(P(W)) = \omega(\mathbf{x^1}), \ldots, \omega(\mathbf{x^r}) = \mathbf{w^1}, \ldots, \mathbf{w^r} = \mathbf{W}$. □

A walk W is *optimal* if the path $P(W)$ is optimal.

Example 11. *In Fig. 1, $P = \mathbf{v^1}, \ldots, \mathbf{v^6}$ is a path from $\mathbf{v^1} = \alpha(\mathbf{x})$ to $\mathbf{v^6} = \mathbf{x}$, and $c(P) = |\{(\mathbf{v^4}, \mathbf{v^5}), (\mathbf{v^5}, \mathbf{v^6})\}| = \mathbf{2}$. Let $w^i = \omega(\mathbf{v^i})$, for $i = 1, \ldots, 6$; then $W(P) = w^1, \ldots, w^6$ is a walk in Fig. 2 (a). Note that (w^4, w^5) and (w^5, w^6) are not in the live graph L_f; thus $c(W(P)) = 2 = c(P)$.*

In the rest of the paper we consider walks in the n-cube C_f. A walk $W = w^1, \ldots, w^r$ is a *walk for a vector* $\mathbf{x}$ if $\mathbf{x} = \mathbf{w^1} \circ \cdots \circ \mathbf{w^r}$. To evaluate $\mathbf{f}(\mathbf{x})$ we find a walk $W = w^1, \ldots, w^r$ for $\mathbf{x}$ with minimal cost, and then $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{w^1}) \circ \cdots \circ \mathbf{f}(\mathbf{w^r})$. This approach takes the advantage of the fact that the size of the n-cube C_f is independent of the length $l(\mathbf{x})$ of the input vector $\mathbf{x}$.

5. Characteristic Vectors

Now we are interested only in proper vectors. A vector is *minimal* if each component has length 2 or 3. If a transient $\mathbf{t} = \mathbf{t_1} \cdots \mathbf{t_m}$ is proper, the *characteristic transient* of $\mathbf{t}$ is $\tilde{\mathbf{t}}$, where $\tilde{\mathbf{t}} = t_1 t_2$ if m is even, and $\tilde{\mathbf{t}} = t_1 t_2 t_3$ if m is odd. Note that $\tilde{\mathbf{t}}$ is a prefix of $\mathbf{t}$, $\alpha(\tilde{\mathbf{t}}) = \alpha(\mathbf{t})$, $\omega(\tilde{\mathbf{t}}) = \omega(\mathbf{t})$, and $\Delta(\mathbf{t}) \equiv \mathbf{\Delta}(\tilde{\mathbf{t}})$, where $\equiv$ is equivalence modulo 2. If $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ is a proper vector, then $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n)$ is the *characteristic vector* of $\mathbf{x}$. Also, $\tilde{\mathbf{x}}$ is a prefix of $\mathbf{x}$, $\alpha(\tilde{\mathbf{x}}) = \alpha(\mathbf{x})$, $\omega(\tilde{\mathbf{x}}) = \omega(\mathbf{x})$, and $\Delta(\mathbf{x_i}) \equiv \mathbf{\Delta}(\tilde{\mathbf{x}_i})$, for $i \in [n]$. The characteristic vector of any vector is minimal, and every minimal vector is the characteristic vector of some vector. Any vector $\mathbf{x}$ which has $\tilde{\mathbf{x}}$ as its characteristic vector is an *extension* of $\tilde{\mathbf{x}}$.

A function $f : B^n \to B$ *depends* on its k-th argument if there exist $x_i \in B$, such that $f(x_1, \ldots, x_{k-1}, 0, x_{k+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{k-1}, 1, x_{k+1}, \ldots, x_n)$. In this section, we only consider functions that depend on all of their arguments. If $f : B^n \to B$ depends on x_k, then there exists at least one live edge $e = (w^i, w^j)$ in the cube C_f of f, where w^i and w^j differ only in x_k.

Definition 12. *A boolean function $f : B^n \to B$ that depends on all of its variables is* convenient *if, for every proper vector $\mathbf{x}$, the cost $c_f(\mathbf{x})$ of $\mathbf{x}$ is equal to the cost $c_f(\tilde{\mathbf{x}})$ of its characteristic vector $\tilde{\mathbf{x}}$.*

For any vector $\mathbf{x} \in \mathbf{T^n}$, let $\tilde{\varphi}(\mathbf{x}) = \varphi(\tilde{\mathbf{x}})$ be the number of walks for $\tilde{\mathbf{x}}$. When $\Delta(\tilde{\mathbf{x}}_1) = \cdots = \Delta(\tilde{\mathbf{x}}_n) = 2$, the maximal value of $\tilde{\varphi}(\mathbf{x})$ is obtained from Equation (2)

by setting $m = 2n$, and is approximately $\tilde{\varphi}(\mathbf{x}) \approx (\mathbf{4}\pi)^{\frac{\mathbf{1-n}}{\mathbf{2}}}\mathbf{n}^{\mathbf{2n}+\frac{\mathbf{1}}{\mathbf{2}}}$, which is independent of the length m of $\mathbf{x}$; hence the evaluation of a particular function $\mathbf{f}$ can be much more efficient if f is convenient. We first search for an optimal walk for $\tilde{\mathbf{x}}$, and get its cost c; we then compute the boolean value $f(\alpha(\mathbf{x}))$ and construct a transient of length $l(\mathbf{x}) - \mathbf{c}$ beginning with $f(\alpha(\mathbf{x}))$. With fixed n, this can be done in time linear in the length of $\mathbf{x}$.

The next claim follows immediately from Theorem 7.

Corollary 13. *All functions in $\mathcal{G}$ are convenient.*

To prove that a function f is convenient we must verify that, for every vector $\mathbf{x}$, the cost of $\mathbf{x}$ for f is the same as the cost of $\tilde{\mathbf{x}}$ for f. Equivalently, we need to show that, for every minimal vector $\tilde{\mathbf{x}}$, the cost for f of any extension $\mathbf{x}$ of $\tilde{\mathbf{x}}$ is the same as the cost of $\tilde{\mathbf{x}}$ for f.

An edge e in a cube corresponds to a unique coordinate x_i which has complementary values in the two vertices of that edge; we say that e is *an edge in coordinate* x_i. An edge is *incident* to a walk W if it shares at least one vertex with W. A walk W is *complete* if, for every coordinate, there is a live edge in that coordinate incident to W. A vertex v in a cube C_f of a boolean function f is a *focus* if every edge incident to v is live. Any walk through a focus is complete.

Proposition 14. *Let $f : B^n \to B$ be a boolean function, and let C_f be its cube. Let $\mathbf{x}$ be a transient vector and $\tilde{\mathbf{x}}$, its characteristic vector. If an optimal walk W in C_f for $\tilde{\mathbf{x}}$ is complete, then the cost of any optimal walk for $\mathbf{x}$ is $c(W)$.*

Proof. Let W be an optimal walk for $\tilde{\mathbf{x}}$. Since the difference between the number of changes in $\mathbf{x_i}$ and in $\tilde{\mathbf{x}}_i$ is even for any i, the additional changes in $\mathbf{x_i}$ can be inserted after a vertex incident to the live edge in that coordinate is reached. □

6. 3-Variable Functions

Suppose $f : B^n \to B$ and $g : B^n \to B$ are boolean functions. If $g(y_1, \ldots, y_n)$ can be obtained from $f(x_1, \ldots, x_n)$ by renaming the variables and complementing some number of inputs and/or the output, then we write $f \sim g$, where $\sim$ is an equivalence relation [4, 5], and we say that f and g are in the same *symmetry class*. For example, we can start with $x + y$, rename y as z to get $x + z$, complement z to get $x + z'$, and complement this result to get $x'z$. Thus $x'z \sim x + y$. If we know how to evaluate the transient extension of f, then we can also evaluate the transient extensions of all the functions that are in the same symmetry class as f [3]. Hence we consider only one representative function of each symmetry class.

For $n = 3$, there are 256 functions, which can be reduced to 14 symmetry classes [4, 5]. Four classes, represented by 0, x, $x + y$, and $x \oplus y$, contain degenerate functions; these classes account for 38 functions which can all be evaluated using the formulas in Section 2. The remaining 218 functions can be reduced to 2 symmetry

classes (18 functions) in $\mathcal{G}$ and 8 classes (200 functions) represented by the functions shown in Fig. 3, where the circled vertices can be ignored for now.

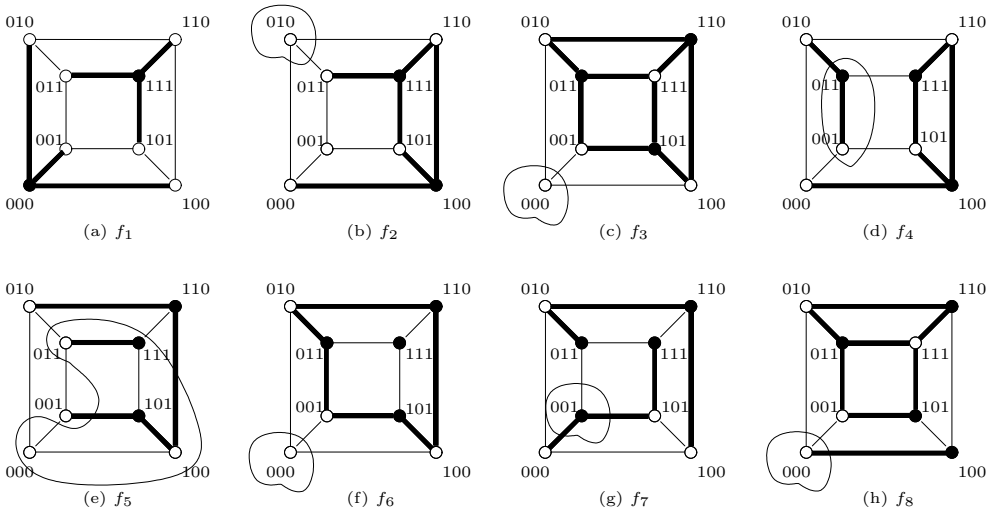


Fig. 3. Representatives of the eight symmetry classes of 3-variable functions.

The main result in this section is the following:

Theorem 15. *All 3-variable functions are convenient.*

To prove Theorem 15, it is sufficient to examine the eight functions of Fig. 3. The next four results are useful in proving that certain walks are optimal.

Lemma 16. *Let W_1 and W_2 be two walks from vertex u to vertex v. Then the difference between the cost of W_1 and that of W_2 is a multiple of 2.*

Proof. Consider a walk W from u to v. If $u_i \neq v_i$ ($u_i = v_i$), then variable x_i must change an odd (even) number of times in W. If there are k components that differ in u and v, then the total number of changes must be $k + 2l$ for some $l \geq 0$. So the lengths of all walks from u to v have the same parity.

Now consider any two walks W_1 and W_2 from u to v, and let walk W_2' from v to u be W_2 reversed. Combining W_1 and W_2' we get a walk W from u to itself. Since the lengths of W_2 and W_2' are the same, and the lengths of W_1 and W_2 have the same parity, the length of W is even.

Consider the sequences $f^0, f^1, \ldots, f^{r-1}, f^r$ and $g^0, g^1, \ldots, g^{p-1}, g^p$ of function values of the vertices in walks W_1 and W_2, respectively. The vertices of W have the sequence $f^0, \ldots, f^{r-1}, f^r = g^p, g^{p-1}, \ldots, f^0 = g^0$. The live edges in W are those edges whose endpoints have different function values, $f^i \neq f^{i+1}$ or $g^{j+1} \neq g^j$. Since

$f^0 = g^0$, there must be even number of such pairs (f^i, f^{i+1}) or (g^{j+1}, g^j), that is, there is an even number of live edges in W. Since the length of W is even, there is an even number of dead edges in W, which are the dead edges in W_1 and W_2.

Note that the cost of W_1 (W_2) is the number of dead edges in W_1 (W_2). Therefore $c(W_1) + c(W_2)$ is even, and so $c(W_1) - c(W_2)$ is also even. So the lemma holds. □

Corollary 17. *Let $c \geq 0$ be any integer. If there is no walk from u to v of cost less than or equal to $c-1$, and there is a walk W of cost $c+1$, then W is optimal. In particular, every walk of cost 1 from u to v is optimal.*

Lemma 18. *For a 3-variable function and a minimal vector $\mathbf{x}$, if $\alpha(\mathbf{x})$ is within distance 1 from a focus, then there is a complete optimal walk for $\mathbf{x}$.*

Proof. Figure 4 shows a 3-variable cube. We use the convention that a vertex (b_1, b_2, b_3), with $b_i \in B$, is represented by the integer $4b_1 + 2b_2 + b_3$. We enumerate

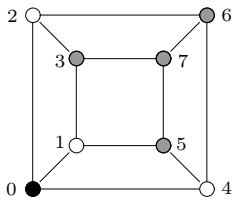


Fig. 4. Starting vertex at distance 1 of a focus.

all optimal walks starting from a given vertex. Since every walk from a focus is complete, we need to consider only start vertices that are distance 1 from a focus. Because of the symmetry of the cube, we can assume without loss of generality that the starting vertex is 1, and 0 is a focus. Note that every minimal vector $\mathbf{x}$ of three variables can be represented as $\langle (x_1, x_2, x_3), \Delta_1, \Delta_2, \Delta_3 \rangle$, where $x_i = \alpha(\mathbf{x_i})$ and $\Delta_i = \Delta(\mathbf{x_i})$, $\Delta_i \in \{1, 2\}$. Every walk of cost 0 is optimal, and if a walk has cost at most 1, then we know by Corollary 17 that it is optimal. Also, if a walk has cost at most 2 and there is no walk of cost 0, then it is optimal. There are eight cases:

(1) If $\Delta_1 = 1$, $\Delta_2 = 1$, $\Delta_3 = 1$, then $1, 0, 4, 6$ has cost at most 1.
(2) If $\Delta_1 = 1$, $\Delta_2 = 1$, $\Delta_3 = 2$, then $1, 0, 4, 5, 7$ has cost at most 2, and no walk has cost 0.
(3) If $\Delta_1 = 1$, $\Delta_2 = 2$, $\Delta_3 = 1$, then $1, 0, 2, 0, 4$ has cost 0.
(4) If $\Delta_1 = 1$, $\Delta_2 = 2$, $\Delta_3 = 2$, then $1, 0, 2, 0, 1, 5$ has cost at most 1.
(5) If $\Delta_1 = 2$, $\Delta_2 = 1$, $\Delta_3 = 1$, then $1, 0, 4, 0, 2$ has cost 0.
(6) If $\Delta_1 = 2$, $\Delta_2 = 1$, $\Delta_3 = 2$, then $1, 0, 4, 0, 2, 3$ has cost at most 1.
(7) $\Delta_1 = 2$, $\Delta_2 = 2$, $\Delta_3 = 1$, then $1, 0, 2, 0, 4, 0$ has cost 0.
(8) If $\Delta_1 = 2$, $\Delta_2 = 2$, $\Delta_3 = 2$, then $1, 0, 2, 0, 4, 0, 1$ has cost 0.

Therefore, for all cases there is an optimal walk which is complete. □

For any 3-variable function f, a walk $v_0v_1v_2v_3v_4v_5$ on C_f is *alternating* if each subwalk $v_iv_{i+1}v_{i+2}v_{i+3}$, $i = 0, 1, 2$, contains a change in every coordinate.

Lemma 19. *Let* $\mathbf{x}$ *be a minimal vector for a 3-variable function, let* $U = v_0v_1v_2v_3$ *be any walk such that* $v_0 = \alpha(\mathbf{x})$*, and let* $W = Uv_4v_5$*. If* W *is alternating,* $c(U) = 0$*, and* $c(W) \leq 1$*, then there exists a complete optimal walk* V *for* $\mathbf{x}$*.*

Proof. Suppose $\mathbf{x} = \langle \mathbf{v_0}; \mathbf{\Delta_1}, \mathbf{\Delta_2}, \mathbf{\Delta_3} \rangle$. If $\Delta_1 = \Delta_2 = \Delta_3 = 1$, let $V = U$. Then V has all three changes since it is alternating, and so it is a walk for $\mathbf{x}$. Since V has cost 0, it is optimal. If $\Delta_1 = \Delta_2 = \Delta_3 = 2$, let $V = Uv_2v_1v_0$. Then V is a walk for $\mathbf{x}$, has cost 0, and so is optimal. If exactly one of Δ_1, Δ_2 and Δ_3 is 2, use $V = Uv$, where (v_3, v) is an edge in the coordinate that has two changes. The cost of this walk is at most 1, and so it is optimal by Corollary 17. If exactly two of Δ_1, Δ_2 and Δ_3 are 2, assume without loss of generality that $\Delta_1 = 1$ and $\Delta_2 = \Delta_3 = 2$. There are two subcases: a) If (v_2, v_3) is an edge in the first coordinate, then $V = W$ has the correct changes and cost at most 1. By Corollary 17, V is optimal. b) If (v_2, v_3) is in the second (third) coordinate, then take $V = Uv_2v$, where (v_2, v) is an edge in the third (second) coordinate. The cost of V is at most 1, and so it is optimal by Corollary 17. Since V begins with U in all cases, V is a complete walk for $\mathbf{x}$. □

We are now ready to give the **proof of Theorem 15:**

Proof. For each of the eight functions in Fig. 3, we enumerate all minimal vectors $\mathbf{x} = \langle \alpha(\mathbf{x}); \mathbf{\Delta_1}, \mathbf{\Delta_2}, \mathbf{\Delta_3} \rangle$, and prove that there is a complete optimal walk for each $\mathbf{x}$. The vertices that need to be considered are circled in the figure. Since there are eight functions and each of them has eight possible starting vertices $\alpha(\mathbf{x})$ and eight change vectors $(\Delta_1, \Delta_2, \Delta_3)$, there are 512 cases to analyze. We reduce this number significantly by using Corollary 17, Lemmas 18 and 19, and symmetry.

To simplify the notation, we denote walks by words, rather than sequences. We give only sketches of proofs for f_4, f_6, f_7 and f_8, and complete proofs for the remaining four functions. The eight functions are treated as follows:

1. For f_1, every vertex is within distance 1 from a focus; by Lemma 18, no vertex needs to be considered.

2. For $f_2 = x_1(x_2 \oplus x_3)$, only vertices 1 and 2 are not within distance 1 of a focus. Since f_2 is symmetric in x_2 and x_3, we consider only one of 1 and 2, say 2. We list the complete walks in pairs $(\Delta_1\Delta_2\Delta_3, W)$, where W is a complete optimal walk for $\langle 2; \Delta_1, \Delta_2, \Delta_3 \rangle$. There are no minimal walks of cost 0. The walks of cost 1 are: $(111, 2645)$, $(112, 26454)$, $(121, 26467)$, $(122, 264676)$, $(212, 267640)$, and $(221, 264673)$. Walks $(211, 26451)$ and $(222, 2646762)$ are of cost 2; by Corollary 17, they are optimal. Each walk is complete for it goes through a focus.

3. For f_3, only 0 is not within distance 1 of a focus, so we consider it. Since every outgoing edge is dead, there are no walks of cost 0. The optimal walks

of cost 1 are: $(111, 0467)$, $(112, 04676)$, $(121, 04645)$, $(122, 046454)$, $(211, 04513)$, $(212, 045462)$, and $(221, 046451)$. Walk $(222, 0157640)$ has cost 2. By Corollary 17, it is optimal as there are no walks of cost 0. Each of these walks is complete because it goes through a focus, 3, 5, 6, or 7.

4. For $f_4 = x_2x_3 + x_1x_2'x_3'$, $1, 2, 3, 7$ are not within distance 1 of a focus. Since f_4 is symmetric in x_2 and x_3, we consider only 1 and not 2. For 7, there is a walk $W = 754023$, which satisfies the conditions of Lemma 19. So there is a complete optimal walk for any minimal vector starting at 7, and we consider only 1 and 3.

5. The function $f_5 = x_1(x_2 + x_3)$ has no focus. Since it is symmetric in x_2 and x_3, we consider only 1 (and not 2) and 5 (and not 6), say. Walk 154623 satisfies the conditions of Lemma 19, taking care of 1. So we consider 0, 3, 4, 5, and 7. For 0, there is no minimal walk of cost 0. Walks $(112, 01546)$, $(121, 02645)$, $(122, 045464)$, $(212, 015462)$, $(221, 046451)$ are of cost 1. Walks $(111, 0457)$, $(211, 04573)$ and $(222, 0464510)$ are of cost 2. For 3, there is no minimal walk of cost 0. Walks $(111, 3754)$, $(112, 37645)$, $(121, 37646)$, $(212, 326451)$, $(221, 375462)$ are of cost 1. Walks $(122, 376457)$, $(211, 37540)$ and $(222, 3764573)$ are of cost 2. For 4, walks $(112, 15767)$, $(121, 15754)$, $(212, 151323)$, $(221, 157510)$, $(222, 1575101)$ are of cost 0. Walks $(111, 1576)$, $(211, 15132)$ and $(122, 157545)$ are of cost 1. For 5, walks $(111, 5462)$, $(211, 54626)$, $(122, 546451)$, $(221, 546264)$, $(222, 5462645)$ are of cost 0. Walks $(112, 54623)$, $(121, 54620)$ and $(212, 546267)$ are of cost 1. For 7 there is no minimal walk of cost 0. Walks $(112, 76451)$, $(121, 75462)$, $(211, 73264)$, $(212, 742645)$, $(221, 762646)$ are of cost 1. Walks $(111, 7540)$, $(122, 754673)$ and $(222, 7626457)$ are of cost 2. One verifies that all of these walks are complete.

6. For $f_6 = x_1x_2 + x_2x_3 + x_3x_1$, there is no focus. Because f is symmetric in all three variables, it suffices to consider 1 (and not 2 and 4), 6 (and not 3 and 5), 0 and 7. Moreover, if we complement the function, the live and dead edges are preserved. Hence 1 and 6 are symmetric in the cube as are 0 and 7, and we consider only 0 and 1. For 1, walk 132645 meets the conditions of Lemma 19; so we look only at 0.

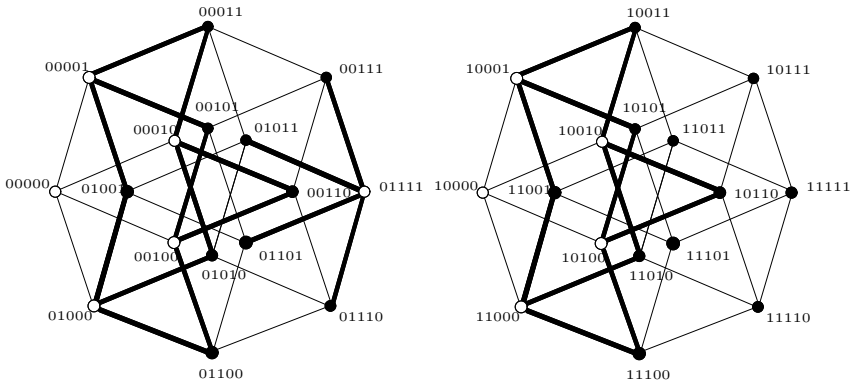
7. For f_7, vertices 0, 3, 4, and 7 are symmetric in the live graph, as are 1, 2, 5, and 6. The alternating walk 015762 takes care of 0, leaving only 1 to consider.

8. For f_8, only 0 and 4 are not within distance 1 of a focus, and they are symmetric with respect to live edges. So we examine only 0. □

We now show a function which is not convenient. Let $S_i(x_1, x_2, x_3, x_4)$ be the symmetric function of four variables that is 1 if and only if precisely i of its variables are 1. For example, $S_3(x_1, x_2, x_3, x_4) = x_1x_2x_3x_4' + x_1x_2x_3'x_4 + x_1x_2'x_3x_4 + x_1'x_2x_3x_4$. Also, let $S_{2,3}(x_1, x_2, x_3, x_4) = S_2(x_1, x_2, x_3, x_4) + S_3(x_1, x_2, x_3, x_4)$.

Proposition 20. *$f = S_{2,3}(x_1, x_2, x_3, x_4) + x_0x_1x_2x_3x_4$ is inconvenient.*

Proof. The cube C_f is shown in Fig. 5. The left subcube C_0 shows f when $x_0 = 0$, and the right subcube C_1, when $x_0 = 1$; the edges between the two subcubes are



The subcube C_0 of f with $x_0 = 0$ The subcube C_1 of f with $x_0 = 1$

Fig. 5. The cube of an inconvenient function of five variables.

not drawn. For convenience, for each vertex with label $(x_0, x_1, x_2, x_3, x_4)$, we use the value $16x_0 + 8x_1 + 4x_2 + 2x_3 + x_4$ to represent it; for example, the vertex with label $(0, 1, 1, 0, 1)$ is 13. Note that f has the property that, for any pair of vertices which differ only in x_0, the values of f are the same, except for the pair $(15, 31)$.

Consider the transient extension $\mathbf{f}$ of f with input: $\mathbf{x} = (\mathbf{0101}, \mathbf{010}, \mathbf{010}, \mathbf{010}, \mathbf{010})$. The initial vertex is 0, and the final one is 16. If transient $\mathbf{x_0} = \mathbf{0101}$ is to produce a change in the value of $\mathbf{f}$, every corresponding walk has to move from subcube C_0 to subcube C_1 via the live edge $(15, 31)$. This is the only live edge between the two subcubes. To produce the longest transient, there are the following possibilities:

1. The optimal walk uses the live edge $(15, 31)$ three times. Then each variable in $\{x_1, \ldots, x_4\}$ must change exactly twice. By inspection of C_0, we see that we have to change each of $x_1, \ldots, x_4$ once to reach 15. Eventually, in C_1, we have to change each of $x_1, \ldots, x_4$ once again to reach 16. Since the function is symmetric in $x_1, \ldots, x_4$, the order in which we change these variables is immaterial. We see that f changes twice in the walk from 0 to 15, and once in the walk from 31 to 16. To show this we add the three changes caused by x_0, for a total of six. The walk must have the form $0, \ldots, 15, 31, 15, 31 \ldots, 16$.

2. The optimal walk uses the live edge $(15, 31)$ twice. To achieve the largest number of changes, the walk has to go from 0 to 15; this results in two changes in f; and then two more changes caused by x_0. The walk is now in 15, and it remains to change each of $x_1, \ldots, x_4$ to 0 and x_0 to 1. The walk may stay in C_0 for a while, then change x_0, and then remain in C_1 to reach 16. So the walk has the form $0, \ldots, 15, 31, 15, \ldots, i, j \ldots, 16$, where i is in C_0 and j is in C_1. Now consider the walk from the second 15. One verifies that, no matter where the transition to C_1 takes place, we get two changes in f, for a total of six.

3. The optimal walk uses the live edge $(15, 31)$ once. Then the combined number of changes from C_0 and C_1 is at most three. Adding the one from x_0, we get a total of four. This contradicts the assumption that the walk is optimal.

4. The optimal walk does not use the live edge $(15, 31)$. Since all the variable changes result in the same function changes in both subcubes, this is equivalent to finding the optimal walk in C_0. One verifies that the maximal number of changes is also 6.

The characteristic vector of the input $\mathbf{x}$ is $\tilde{\mathbf{x}} = (01, 010, 010, 010, 010)$. By Case 4 of our analysis, $\tilde{\mathbf{x}}$ can also produce six changes in f. Since $l(\tilde{\mathbf{x}}) = 9$, the cost of $\tilde{\mathbf{x}}$ is 3. However, $l(\mathbf{x}) = \mathbf{11}$, and the cost of $\mathbf{x}$ is 5. Thus f is not convenient. □

7. Conclusions

The evaluation of extensions of boolean functions is simplified if we use walks in boolean cubes instead of paths in digraphs. The evaluation of extensions of convenient functions can be done with characteristic vectors in polynomial time. All 3-variable functions are convenient, and there exist inconvenient 5-variable functions. It remains open whether there is an inconvenient 4-variable function. The problem of characterizing convenient functions is also open.

Acknowledgments: This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871 and a Postgraduate Scholarship, and by a Graduate Award from the Department of Computer Science, University of Toronto.

References

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications* (American Elsevier, New York, 1976).

[2] J. Brzozowski and Z. Ésik, Hazard algebras, *Formal Methods in System Design* **23** (3) (2003) 223–256.

[3] J. Brzozowski and Y. Ye, Gate circuits with feedback in finite multivalued algebras of transients, *J. of Mult.-Valued Logic & Soft Computing* **16** (1–2) (2010) 155–176.

[4] S. H. Caldwell, *Switching Circuits and Logical Design* (Wiley, New York, 1958).

[5] M. A. Harrison, *Introduction to Switching and Automata Theory* (McGraw-Hill, New York, 1965).

[6] L. B. Richmond and J. Shallit, Counting abelian squares, *Electronic J. Combinatorics* **16** (1) (2008) #R72.