A Lower Bound on the Number of Variables to Represent Incompletely Specified Index Generation Functions

Tsutomu Sasao, Yuta Urano, Yukihiro Iguchi
Dept. of Computer Science
Meiji University
Kawasaki, Kanagawa 214-8571, Japan

Abstract—Given an incompletely specified index generation function, the number of variables to represent the function can often be reduced by properly assigning don’t care values. In this paper, we derive a lower bound on the number of variables necessary to represent a given incompletely specified index generation function. We also derive three properties of incompletely specified index generation functions. We confirm these properties by experiments using random index generation functions.

Keywords—incompletely specified function, functional decomposition, logic minimization, random function.

I. INTRODUCTION

Index generation functions have wide applications for pattern matching in the Internet [4][5][11]. In an incompletely specified function \( f \), the number of variables to represent \( f \) can often be reduced by properly assigning don’t care values [1][2][3][7-16]. With this property, we can represent incompletely specified index generation functions more compactly than original specifications.

In this paper, we consider lower bounds on the number of variables necessary to represent incompletely specified index generation functions. We show that a lower bound for \( n \) variable index generation functions with weight \( k \) can be obtained by numerical calculation.

The rest of the paper is organized as follows: Section II defines terminology; Section III derives a lower bound on the number of variables necessary to represent an incompletely specified index generation function with weight \( k \); Section IV considers statistical properties of uniformly distributed incompletely specified index generation functions; Section V shows experimental results; and finally, Section VI summarizes the paper and presents future problems.

II. INCOMPLETELY SPECIFIED INDEX GENERATION FUNCTION

Definition 2.1: Consider a set of \( k \) different vectors of \( n \) bits. These vectors are registered vectors. For each registered vector, assign an index, a unique integer from 1 to \( k \). A registered vector table maps an index to each registered vector.

Definition 2.2: An incompletely specified index generation function \( f \) represents a mapping \( D \rightarrow \{1, 2, \ldots, k\} \), where \( D \) denotes the set of the registered vectors, \( D \subseteq B^n \); \( B = \{0, 1\} \); \( |D| = k \); and \( |D| \) denotes the number of the elements in the set \( D \). An incompletely specified index generation function represents the corresponding index when the input vector matches a registered vector. Otherwise, the value of the function is undefined. \( k \) is called the weight of the index generation function.

Example 2.1: Consider the registered vectors shown in Table 2.1. It shows an incompletely specified index generation function with weight \( k = 4 \).

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2.1 Registered Vector Table.

Definition 2.3: A completely specified index generation function \( f \) represents a mapping \( B^n \rightarrow \{0, 1, 2, \ldots, k\} \). Let \( D \) be the set of registered vectors. When \( \bar{a} \in D \), the value of \( f(\bar{a}) \) is the same as that of the corresponding incompletely specified index generation function. When \( \bar{a} \in B^n - D \), \( f(\bar{a}) = 0 \).

A circuit for a completely specified index generation function can be easily implemented from a circuit for an incompletely specified index generation function [11]. Thus, the problem is to find an economical realization of a given incompletely specified index generation function.

In this paper, incompletely specified index generation functions are often called index generation function, for short. The number of variables needed to represent incompletely specified index generation functions can often be reduced [11].

Theorem 2.1: [10] Suppose that an incompletely specified index generation function is represented by a decomposition chart [6]. When the decomposition chart has at most one non-zero element in each column, the function can be represented with only column variables.

(Proof) For each column, set the values of don’t cares to the value of the care element, and the function depends on only the column variables. \( \square \)
variables is given by

$$\eta$$

Then, only $x_1$ and $x_2$ are column variables, while $x_3$ and $x_4$ are row variables. In the table, blank cells denote don’t cares. In the decomposition chart shown in Table 2.2, each column has at most one non-zero element. In this case, the incompletely specified index generation function can be represented with only $x_1$ and $x_2$:

$$f = 1 \cdot \bar{x}_1 \bar{x}_2 \vee 4 \bar{x}_1 x_2 \vee 3 \cdot x_1 x_2 \vee 2 \cdot x_1 \bar{x}_2$$

### III. Number of Variables Necessary to Represent Index Generation Functions

For a given $n$ variable index generation function $f$, if we can estimate the number of variables to represent $f$, then we can estimate the size of hardware to realize it. We assume that index generation functions are implemented by memories. Thus, the number of the variables is vitally important.

A lower bound on the number of variables has been obtained as $\lceil \log_2 k \rceil$ in [12], and an upper bound has been obtained as $2 \lceil \log_2 k \rceil - 3$ [11]. Unfortunately, when $k$ is large, the difference between these bounds is rather large.

**Definition 3.1**: Let $a$ and $b$ be integers such that $a \geq b$. Then, $a^P_b$ denotes the number of sequences of length $b$ of elements taken from a set of $a$ distinct elements. That is,

$$a^P_b = \frac{a!}{(a-b)!}.$$  

The probability $\eta(p, n, m, k)$ that $p$-valued input $n$ variable index generation functions with weight $k$ can be represented with only the first $m$ variables is derived in [14]. By setting $p = 2$, we have the following:

**Theorem 3.1**: Given an $n$ variable incompletely specified index generation function $f$ with weight $k$, the probability $\eta(n, m, k)$ that $f$ can be represented with only the first $m$ variables is given by

$$\eta(n, m, k) = \frac{2^m P_{k, m} \cdot 2^{n-m-k}}{2^m P_k} .$$

(Proof) The probability that a function can be represented with only the first $m$ variables is given by $\eta(n, m, k) = \frac{A}{B}$, where $A$ denotes the number of incompletely specified index generation functions with weight $k$ that can be represented with $x_1, x_2, \ldots, x_m$. $B$ denotes the total number of incompletely specified index generation functions with weight $k$.

1) $A$ denotes the number of the incompletely specified index generation functions with weight $k$, where each column has at most one non-zero element in the decomposition chart. First, enumerate the ways to specify the columns that have non-zero elements. This is equal to the number of ways to distribute $k$ distinct elements into $2^n$ distinct bins: $2^m P_k$. Second, enumerate the ways to specify the row for each element. This is equal to the number of ways to select one row out of $2^{n-m}$ rows, and there are $k$ elements. Thus, the total number of ways to select the rows is $(2^{n-m})^k = 2^{(n-m)k}$. Thus, we have $A = 2^m P_k \cdot 2^{(n-m)k}$.

2) $B$ denotes the total number of $n$ variable incompletely specified index generation functions with weight $k$. This is equal to the number of ways to distribute $k$ distinct elements into $2^n$ distinct bins: $2^m P_k$.

From these, we have theorem.

**Lemma 3.1**: [11] If $0 < x < 1$, then $1 - x$ can be approximated by $e^{-x}$, where $e$ denotes the base of the natural logarithm, and $\ll$ means much less than.

Assume that $k \ll 2^n$. From Lemma 3.1, we have the following:

$$\eta(n, m, k) \approx \frac{\prod_{i=1}^{k-1} \left(1 - \frac{i}{2^n}\right)}{\prod_{i=1}^{k-1} \left(1 - \frac{i}{2^n}\right)} = \exp\left[\frac{-k(k-1)}{2^{n+2}}\right].$$

From this, we have the following:

**Corollary 3.1**:

$$\eta(n, m, k) \approx \exp\left[\frac{k^2}{2^{n+1}}(1 - 2^{n-m})\right].$$

**Theorem 3.2**: Assume that $n$ is sufficiently large. Given an $n$ variable index generation function $f$ with weight $k$, the probability $PR$ that $f$ can be represented with only $m$ variables is

$$PR(n, m, k) = 1 - \left(1 - \eta(n, m, k)\right)^{(n)}.$$

(Proof) The probability that the function cannot be represented with only $x_1, x_2, \ldots, x_{n-1}$, and $x_m$ is $1 - \eta(n, m, k)$.

Note that the number of combinations to select $m$ variables

<table>
<thead>
<tr>
<th>Table 2.2 Decomposition Chart.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
out of \( n \) variables is \( \binom{n}{m} \). Thus, the probability that at least one combination can represent the function with \( m \) variables is given by

\[
PR(n, m, k) = 1 - \left( 1 - \eta(n, m, k) \right) \binom{n}{m}.
\]

In the above proof, we assumed that \( n \) and \( m \) are sufficiently large, and the logic functions can be treated statistically.

IV. STATISTICAL PROPERTIES OF INDEX GENERATION FUNCTIONS

A. When the Values of \( k \) Are Changed

Fig. 4.1 shows the relations among \( m, k \), and \( PR \) for \( n = 20 \). In Fig. 4.1, the probability \( PR \) that a function can be represented with \( m \) variables suddenly decreases with the increase of \( k \). This can be explained with Equations (3.1) and (3.2).

- When \( \frac{k^2}{2n+1} \to 0 \), we have
  \[
  \eta(n, m, k) \simeq \exp\left(\frac{k^2}{2n+1}(1 - 2^{n-m})\right) \simeq \exp\left(-\frac{k^2}{2m+1}\right) \to 1
  \]

  Thus, we have
  \[
  PR(n, m, k) \to 1 - (1 - 1)\binom{n}{m} = 1.
  \]

- When \( \frac{k^2}{2n+1} \to \infty \), we have
  \[
  \eta(n, m, k) \simeq \exp\left(-\frac{k^2}{2m+1}\right) \to 0
  \]

  Thus, we have
  \[
  PR(n, m, k) \to 1 - (1 - 0)\binom{n}{m} = 0.
  \]

B. When the Values of \( m \) Are Changed

Figures 4.2, 4.3, and 4.4 show the relations among \( n, m \), and \( PR \), for \( k = 63 \), \( k = 255 \), and \( k = 1023 \), respectively. For example, when \( k = 63 \) and \( n = 12 \), the probabilities are almost 0.0 for \( m = 7 \); about 0.2 for \( m = 8 \); and almost 1.0 for \( m = 9 \).

In Figures 4.2, 4.3, and 4.4, note that the difference of \( m \) are at most 2 when the values of \( PR \) are changed from 0.00 to 1.00.

In Fig. 4.4, the line for \( n = 12 \) is shown up to \( m = 12 \), since \( m \leq n \).

Definition 4.1: Let \( M_{50} \) be the minimum real number that satisfies the following relation in Equation (3.2):

\[
PR(n, M_{50}, k) = 0.5.
\]

Note that, the values of \( \binom{n}{m} \) are normally defined only when both \( n \) and \( m \) are integers. However, in this case, we extend
the function so that \( n \) and \( m \) can take any positive real values as follows:

\[
\binom{n}{m} = \frac{\Gamma(n + 1)}{\Gamma(n - m + 1) \cdot \Gamma(m + 1)},
\]

where \( \Gamma(n) \) is the gamma function\(^1\).

Then, we have the following:

**Property 4.1:** To represent most incompletely specified index generation functions, at least \( \lceil M_{50} \rceil \) variables are necessary, where \( \lfloor a \rfloor \) denotes the integer part of the positive real number \( a \).

**C. When the Values of \( n \) Are Changed.**

Equation (3.2) implies that, when the value of \( PR(n, m, k) \) is small, the representation of the functions with only \( m \) variables is unlikely. Since the value of \( \eta(n, m, k) \) is sufficiently small, \( 1 - \eta(n, m, k) \) can be approximated by \( e^{-\eta(n,m,k)} \). Thus, the condition that makes \( PR(n, m, k) = 0.5 \) in Equation (3.2) can be represented as

\[
PR(n, m, k) \approx 1 - \exp(-\eta(n, m, k)s(n, m)) = 0.5,
\]

where

\[
s(n, m) = \frac{\Gamma(n + 1)}{\Gamma(n - m + 1) \cdot \Gamma(m + 1)}.
\]

In other words, we have the following:

\[
\eta(n, m, k)s(n, m) = \log_e 2 \approx 0.6931. \quad (4.1)
\]

Given the values of \( n \) and \( k \), the numerical values of \( M_{50} \) can be easily calculated by Equations (3.1) and (4.1) using a computer. Fig. 4.5 shows the relation among \( n, m, \) and \( k \) when \( PR(n, M_{50}, k) = 0.5 \). Fig. 4.5 shows that, for example, 14 variable index generation functions with weight \( k = 127 \) can be represented by \( m = 10 \) variables, with probability 0.5. Here, we consider the region for \( n \leq \log_2(k + 1) \) − 4. In this region, the relation \( n \approx m \) holds.

From this, we have the following:

**Property 4.2:** For most \( n \) variable incompletely specified index generation functions with weight \( k \), the reduction of variables is difficult when \( n \leq L \), where \( L = 2\lceil \log_2(k + 1) \rceil - 4 \).

Also, from the observation before Definition 4.1, we have:

**Property 4.3:** When \( n \) is sufficiently large and \( k \ll 2^n \), most incompletely specified index generation functions with weight \( k \) can be represented by \( L - 1, L, \) or \( L + 1 \) variables, where \( L = 2\lceil \log_2(k + 1) \rceil - 4 \).

**V. Experimental Results**

To test the properties obtained in the previous section, we produced many random index generation functions, and obtained the numbers of variables to represent the functions.

**A. Test for Property 4.1**

To test Property 4.1, we produced 1,000 random index generation functions for various pairs of \((n, k)\). Table 5.1 shows the minimum values to represent \( n \) variable incompletely specified index generation functions with weight \( k \).

We used the algorithm in [9] to obtain exact minimum number of variables. In the table, \( Min \) denotes the minimum value, and \( Count \) denotes the number of functions that gives the minimum value. \( M_{50} \) denotes the value of \( m \) that makes \( PR(n, M_{50}, k) = 0.5 \).

In Table 5.1, except for the case of \( n = 24 \) and \( k = 8191 \), the relation \( Min \geq \lceil M_{50} \rceil \) holds. When \( n = 24 \) and \( k = 8191 \), the number of functions that require the minimum value 21 is only one out of 1,000. Thus, \( M_{50} \approx 22 \). In other words, Property 4.1 holds.

---

\(^1\)The gamma function is defined as

\[
\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx.
\]
functions, when linear transformations can be used.

B. Test for Property 4.2

To test Property 4.2, we produced 1,000 random index generation functions for various pairs of \((n, k)\). Table 5.2 shows the average numbers of variables necessary to represent \(n\) variable incompletely specified index generation functions with weight \(k\). The values in the table are the average of 1,000 randomly generated functions. The numbers written in boldface denote the average when \(n = L\) holds, where \(L = 2\lfloor\log_2(k+1)\rfloor - 4\).

Note that for the entries that are below the boldface numbers, virtually no variables could be removed. In other words, Property 4.2 holds.

C. Test for Property 4.3

To test Property 4.3, we produced 1,000 random index generation functions for \(n = 22, 24, 26, 28\) and 30. In Table 5.3, the columns headed by \(L - 1, L,\) and \(L + 1\) denote the number of functions that require \(L - 1, L,\) and \(L + 1\) variables, respectively, where \(L = 2\lfloor\log_2(k+1)\rfloor - 4\). Note that most functions can be represented with \(L - 1, L,\) or \(L + 1\) variables.

In the table, in some rows, the sums of the numbers in the three columns are less than the total number of sample functions. Such rows are denoted by boldface. For example, when \(n = 22\) and \(k = 2047\), only one function required \(L + 2 = 20\) variables. However, when \(n = 26, n = 28\) and \(n = 30\), Property 4.3 holds for all the samples.

VI. SUMMARY AND FUTURE PROBLEMS

In this paper, we derived lower bounds on the number of variables necessary to represent incompletely specified index generation functions. Also, given the values for \(n\) and \(k\), we derived a method to predict whether the number of variables can be reduced or not.

In this paper, we assumed that 0’s and 1’s are equally likely to occur in registered vector tables. However, in practical applications, distributions of 0’s and 1’s are not always equal. Future problems include the case where the distribution of 0’s and 1’s are different. Also, we should obtain lower bounds on the numbers of compound variables [13] to represent incompletely specified index generation functions, when linear transformations can be used.

ACKNOWLEDGMENTS

This work is partially supported by the Japan Society for the Promotion of Science (JSPS), Grant in Aid for Scientific Research, and by the Adaptable and Seamless Technology Transfer Program through target-driven R&D, JST. Finally, Prof. Jon T. Butler improved English presentation.

REFERENCES


### Table 5.1
**Minimum number of variables necessary to represent incompletely specified index generation functions.**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$n = 12$</th>
<th>$n = 16$</th>
<th>$n = 20$</th>
<th>$n = 24$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>9</td>
<td>4.081</td>
<td>4</td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>117</td>
<td>6.038</td>
<td>6</td>
</tr>
<tr>
<td>63</td>
<td>8</td>
<td>143</td>
<td>8.157</td>
<td>8</td>
</tr>
<tr>
<td>255</td>
<td>11</td>
<td>1</td>
<td>11.868</td>
<td>12</td>
</tr>
<tr>
<td>1023</td>
<td>12</td>
<td>1000</td>
<td>11.996</td>
<td>15</td>
</tr>
<tr>
<td>2047</td>
<td>12</td>
<td>1000</td>
<td>11.999</td>
<td>16</td>
</tr>
<tr>
<td>4095</td>
<td>12</td>
<td>1000</td>
<td>11.999</td>
<td>16</td>
</tr>
<tr>
<td>8191</td>
<td>-</td>
<td>-</td>
<td>16</td>
<td>1000</td>
</tr>
</tbody>
</table>

### Table 5.2
**Average number of variables necessary to represent incompletely specified index generation functions.**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$L$</th>
<th>$n = 12$</th>
<th>$n = 14$</th>
<th>$n = 16$</th>
<th>$n = 18$</th>
<th>$n = 20$</th>
<th>$n = 22$</th>
<th>$n = 24$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>8</td>
<td>8.965</td>
<td>8.623</td>
<td>8.257</td>
<td>8.053</td>
<td>8.007</td>
<td>8.000</td>
<td>8.000</td>
</tr>
<tr>
<td>127</td>
<td>10</td>
<td>11.074</td>
<td>10.775</td>
<td>10.304</td>
<td>10.043</td>
<td>10.000</td>
<td>9.986</td>
<td>9.963</td>
</tr>
<tr>
<td>1023</td>
<td>16</td>
<td>12.000</td>
<td>14.000</td>
<td>15.991</td>
<td>16.823</td>
<td>16.293</td>
<td>15.985</td>
<td>15.874</td>
</tr>
<tr>
<td>2047</td>
<td>18</td>
<td>12.000</td>
<td>14.000</td>
<td>16.000</td>
<td>17.997</td>
<td>18.758</td>
<td>18.197</td>
<td>17.965</td>
</tr>
<tr>
<td>4095</td>
<td>20</td>
<td>12.000</td>
<td>14.000</td>
<td>16.000</td>
<td>18.000</td>
<td>19.992</td>
<td>20.676</td>
<td>20.093</td>
</tr>
<tr>
<td>8191</td>
<td>22</td>
<td>-</td>
<td>14.000</td>
<td>16.000</td>
<td>18.000</td>
<td>20.000</td>
<td>21.992</td>
<td>22.591</td>
</tr>
</tbody>
</table>

### Table 5.3
**Number of index generation functions that require $L - 1$, $L$, or $L + 1$ variables.**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$L$</th>
<th>$L - 1$</th>
<th>$L + 1$</th>
<th>$L - 1$</th>
<th>$L + 1$</th>
<th>$L - 1$</th>
<th>$L + 1$</th>
<th>$L - 1$</th>
<th>$L + 1$</th>
<th>$L - 1$</th>
<th>$L + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>0</td>
<td>79</td>
<td>921</td>
<td>0</td>
<td>122</td>
<td>878</td>
<td>0</td>
<td>148</td>
<td>852</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>0</td>
<td>961</td>
<td>39</td>
<td>0</td>
<td>997</td>
<td>3</td>
<td>0</td>
<td>999</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>8</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>127</td>
<td>10</td>
<td>0</td>
<td>14</td>
<td>986</td>
<td>0</td>
<td>37</td>
<td>963</td>
<td>0</td>
<td>92</td>
<td>908</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>12</td>
<td>0</td>
<td>49</td>
<td>950</td>
<td>1</td>
<td>131</td>
<td>869</td>
<td>0</td>
<td>355</td>
<td>645</td>
<td>0</td>
</tr>
<tr>
<td>511</td>
<td>14</td>
<td>0</td>
<td>42</td>
<td>958</td>
<td>0</td>
<td>213</td>
<td>787</td>
<td>0</td>
<td>551</td>
<td>449</td>
<td>0</td>
</tr>
<tr>
<td>1023</td>
<td>16</td>
<td>0</td>
<td>21</td>
<td>973</td>
<td>6</td>
<td>126</td>
<td>874</td>
<td>0</td>
<td>470</td>
<td>330</td>
<td>0</td>
</tr>
<tr>
<td>2047</td>
<td>18</td>
<td>0</td>
<td>5</td>
<td>794</td>
<td>200</td>
<td>35</td>
<td>965</td>
<td>0</td>
<td>265</td>
<td>735</td>
<td>0</td>
</tr>
<tr>
<td>4095</td>
<td>20</td>
<td>0</td>
<td>369</td>
<td>586</td>
<td>8</td>
<td>891</td>
<td>101</td>
<td>0</td>
<td>61</td>
<td>939</td>
<td>0</td>
</tr>
<tr>
<td>8191</td>
<td>22</td>
<td>0</td>
<td>992</td>
<td>-</td>
<td>1</td>
<td>436</td>
<td>534</td>
<td>11</td>
<td>919</td>
<td>70</td>
<td>0</td>
</tr>
</tbody>
</table>