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 $D \rightarrow \{1, 2, \ldots, k\}$,

Table 2.1
Registered Vector Table.

x_1	x_2	x_3	x_4	f
0	0	0	1	1
1	0	1	1	2
1	1	0	0	3
0	1	1	1	4

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 $k = 4$. ■

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 $\vec{a} \in D$, the value of $f(\vec{a})$ is the same as that of the corresponding incompletely specified index generation function. When $\vec{a} \in B^n - D$, $f(\vec{a}) = 0$.

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. □

Table 2.2
DECOMPOSITION CHART.

		0	0	1	1	x_1
		0	1	1	0	x_2
0	0			3		
0	1	1				
1	1		4		2	
1	0					
x_3	x_4					

Example 2.2: Table 2.2 is the decomposition chart corresponding to the registered vector table in Table 2.1. 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, ${}_aP_b$ denotes the number of sequences of length b of elements taken from a set of a distinct elements. That is,

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

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

$$\begin{aligned} \eta(n,m,k) &= \frac{{}_{2^m}P_k \cdot 2^{(n-m)k}}{{}_{2^n}P_k} \\ &= \frac{\prod_{i=1}^{k-1}(1-\frac{i}{2^m})}{\prod_{i=1}^{k-1}(1-\frac{i}{2^n})}. \end{aligned}$$

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

1) $\mathcal{A}$ 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^m 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 $\mathcal{A} ={}_{2^m} P_k \cdot 2^{(n-m)k}$.
2) $\mathcal{B}$ 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^ndistinct bins:${}_{2^n}P_k$.

From these, we have theorem. □

Lemma 3.1: [11] If $0 < x \ll 1$, 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 $k \ll 2^m$. From Lemma 3.1, we have the following:

$$\begin{aligned} \eta(n,m,k) &\simeq \frac{\prod_{i=1}^{k-1}(\exp(-\frac{i}{2^m}))}{\prod_{i=1}^{k-1}(\exp(-\frac{i}{2^n}))} \\ &= \frac{\exp(-\sum_{i=1}^{k-1}(-\frac{i}{2^m}))}{\exp(-\sum_{i=1}^{k-1}(-\frac{i}{2^n}))} \\ &\simeq \frac{\exp(-\frac{k(k-1)}{2 \cdot 2^m})}{\exp(-\frac{k(k-1)}{2 \cdot 2^n})} \\ &\simeq \frac{\exp(-\frac{k^2}{2^{m+1}})}{\exp(-\frac{k^2}{2^{n+1}})}. \end{aligned}$$

From this, we have the following:

Corollary 3.1:

$$\eta(n,m,k) \simeq \exp(\frac{k^2}{2^{n+1}}(1-2^{n-m})). \quad (3.1)$$

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

$$PR(n,m,k) = 1 - (1-\eta(n,m,k))^{\binom{n}{m}}. \quad (3.2)$$

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

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

□

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

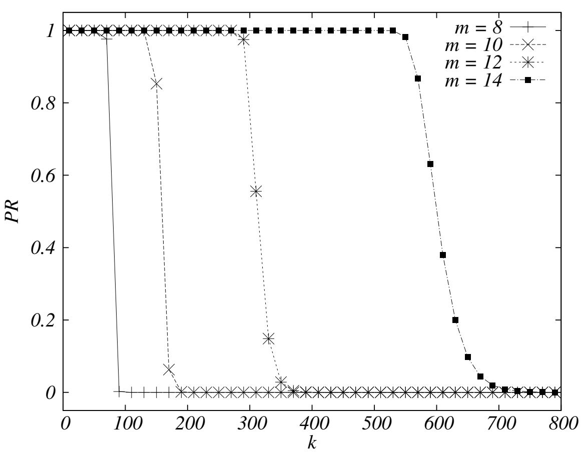

Figure 4.1. Relation among m, k, and PR for $n = 20$.

be represented with m variables suddenly decreases with the increase of k. This can be explained with Equations (3.1) and (3.2).

- When $\frac{k^2}{2^{m+1}} \to 0$, we have

$$\begin{aligned}\eta(n,m,k) &\simeq \exp\left(\frac{k^2}{2^{n+1}}(1-2^{n-m})\right)\\ &\simeq \exp\left(-\frac{k^2}{2^{m+1}}\right) \to 1\end{aligned}$$

  Thus, we have

$$PR(n,m,k) \to 1 - (1-1)^{\binom{n}{m}} = 1.$$

- When $\frac{k^2}{2^{m+1}} \to \infty$, we have

$$\eta(n,m,k) \simeq \exp\left(-\frac{k^2}{2^{m+1}}\right) \to 0$$

  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

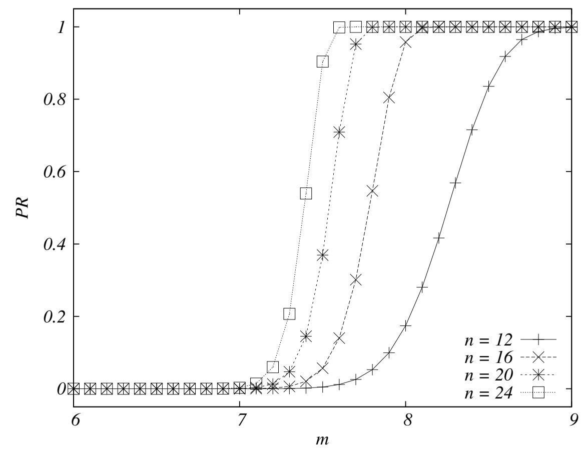

Figure 4.2. Relation among n, m, and PR for $k = 63$.

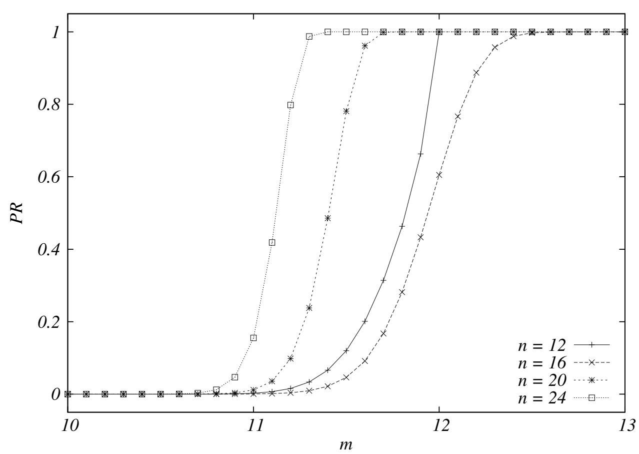

Figure 4.3. Relation among n, m, and PR for $k = 255$.

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

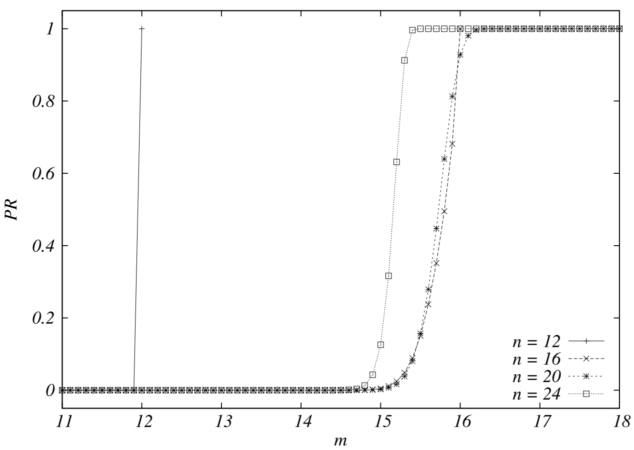

Figure 4.4. Relation among n, m, and PR for $k = 1023$.

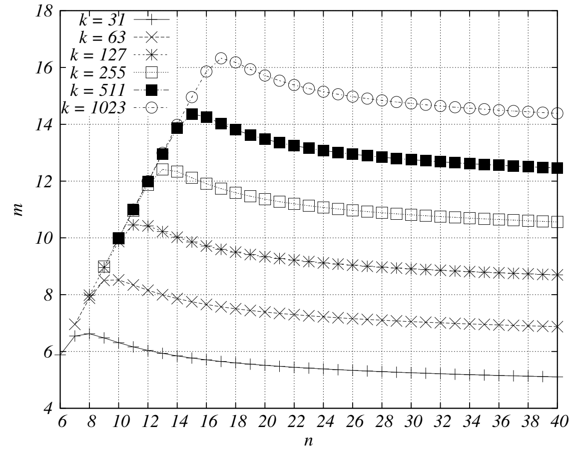

Figure 4.5. Relation among n, m, and k for $PR = 0.5$.

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 $\lfloor M_{50} \rfloor$ 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

$$PR(n, m, k) \simeq 1 - \exp(-\eta(n, m, k)s(n, m)) = 0.5,$$

where

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

In other words, we have the following:

$$\eta(n, m, k)s(n, m) = \log_e 2 \simeq 0.6931. \tag{4.1}$$

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 $n \leq 2\lceil \log_2(k+1) \rceil - 4$. In this region, the relation $n \simeq m$ holds.

[1]The gamma function is defined as

$$\Gamma(t) = \int_0^{\infty} x^{t-1} e^{-x} dx.$$

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* $\geq \lfloor M_{50} \rfloor$ 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, $M_{50} \simeq 22$. In other words, Property 4.1 holds.

Note that in the columns for $n = 12$, entries for $k = 8191$ are shown by '-'. This is due to the constraint $k \leq 2^n$.

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 $L = 2\lceil\log_2(k + 1)\rceil - 4$.

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 $L = 2\lceil\log_2(k+1)\rceil - 4$. 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

[1] F. M. Brown, *Boolean Reasoning: The Logic of Boolean Equations*, Kluwer Academic Publishers, Boston, 1990.

[2] C. Halatsis and N. Gaitanis, "Irredundant normal forms and minimal dependence sets of a Boolean function," *IEEE Trans. on Computers*, Vol. C-27, No. 11, pp. 1064-1068, Nov. 1978.

[3] Y. Kambayashi, "Logic design of programmable logic arrays," *IEEE Trans. on Computers*, Vol. C-28, No. 9, pp. 609-617, Sept. 1979.

[4] H. Nakahara, T. Sasao, and M. Matsuura, "A design method of a regular expression matching circuit based on decomposed automaton," *IEICE Transactions on Information and Systems*, Vol. E95-D, No.2, pp.364-373, Feb. 2012.

[5] H. Nakahara, T. Sasao, and M. Matsuura, "A virus scanning engine using an MPU and an IGU based on row-shift decomposition," *IEICE Transactions on Information and Systems*, Vol. E96-D, No. 8, pp. 1667-1675.

[6] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.

[7] T. Sasao, "On the number of dependent variables for incompletely specified multiple-valued functions," *30th International Symposium on Multiple-Valued Logic*, pp. 91-97, Portland, Oregon, U.S.A., May 23-25, 2000.

[8] T. Sasao, "Design methods for multiple-valued input address generators,"(invited paper) *International Symposium on Multiple-Valued Logic* (ISMVL-2006), Singapore, May 2006.

[9] T. Sasao, "On the number of variables to represent sparse logic functions," *ICCAD-2008*, San Jose, California, USA, Nov. 10-13, 2008, pp. 45-51.

[10] T. Sasao, "On the numbers of variables to represent multi-valued incompletely specified functions," *13th EUROMICRO Conference on Digital System Design* (DSD-2010), Lille, France, Sept. 1-3, 2010, pp. 420-423.

[11] T. Sasao, *Memory-Based Logic Synthesis*, Springer, 2011.

[12] T. Sasao,"Index generation functions: Recent developments,"(invited paper) *International Symposium on Multiple-Valued Logic* (ISMVL-2011), Tuusula, Finland, May 23-25, 2011.

[13] T. Sasao, "Linear decomposition of index generation functions," *17th Asia and South Pacific Design Automation Conference* (ASPDAC-2012), Jan. 30- Feb. 2, 2012, Sydney, Australia, pp. 781-788.

[14] T. Sasao, "Multiple-valued input index generation functions: Optimization by linear transformation,"*International Symposium on Multiple-Valued Logic* (ISMVL-2012), Victoria, Canada, May 14-16, 2012, pp. 185-190.

[15] T. Sasao, "An application of autocorrelation functions to find linear decompositions for incompletely specified index generation functions," *International Symposium on Multiple-Valued Logic* (ISMVL-2013), Toyama, Japan, May 21-24, 2013, pp. 96-102.

Table 5.1
MINIMUM NUMBER OF VARIABLES NECESSARY TO REPRESENT INCOMPLETELY SPECIFIED INDEX GENERATION FUNCTIONS.

	$n = 12$			$n = 16$			$n = 20$			$n = 24$		
k	Min	Count	M_{50}	Min	Count	M_{50}	Min	Count	M_{50}	Min	Count	M_{50}
15	4	9	4.081	4	21	3.863	4	53	3.728	4	122	3.634
31	6	117	6.038	6	523	5.705	6	885	5.514	6	997	5.386
63	8	143	8.157	8	743	7.655	8	993	7.389	8	1000	7.221
127	10	102	10.422	10	696	9.714	9	2	9.338	9	37	9.119
255	11	1	11.868	12	426	11.908	11	8	11.363	11	131	11.071
511	12	1000	11.980	14	206	14.253	13	11	13.478	13	213	13.078
1023	12	1000	11.996	15	9	15.856	15	4	15.718	15	126	15.149
2047	12	1000	11.999	16	1000	15.981	18	304	18.115	17	35	17.300
4095	12	1000	11.999	16	1000	15.996	19	8	19.841	19	8	19.571
8191	-	-	-	16	1000	15.999	20	1000	19.980	21	1	22.003

Table 5.2
AVERAGE NUMBER OF VARIABLES NECESSARY TO REPRESENT INCOMPLETELY SPECIFIED INDEX GENERATION FUNCTIONS.

k	L	$n = 12$	$n = 14$	$n = 16$	$n = 18$	$n = 20$	$n = 22$	$n = 24$
15	4	5.042	4.996	4.980	4.966	4.947	4.921	4.878
31	6	6.924	6.723	6.477	6.251	6.115	6.039	6.003
63	8	8.965	8.623	8.257	8.053	8.007	8.000	8.000
127	10	11.074	10.775	10.304	10.043	10.000	9.986	9.963
255	12	**11.999**	12.960	12.589	12.094	11.996	11.952	11.869
511	14	12.000	**13.999**	14.890	14.429	14.019	13.958	13.787
1023	16	12.000	14.000	**15.991**	16.823	16.293	15.985	15.874
2047	18	12.000	14.000	16.000	**17.997**	18.758	18.197	17.965
4095	20	12.000	14.000	16.000	18.000	**19.992**	20.676	20.093
8191	22	-	14.000	16.000	18.000	20.000	**21.992**	22.591

Table 5.3
NUMBER OF INDEX GENERATION FUNCTIONS THAT REQUIRE $L - 1$, L, OR $L + 1$ VARIABLES.

		$n = 22$			$n = 24$			$n = 26$			$n = 28$			$n = 30$		
k	L	$L-1$	L	$L+1$	$L-1$	L	$L+1$	$L-1$	L	$L+1$	$L-1$	L	$L+1$	$L-1$	L	$L+1$
15	4	0	79	921	0	122	878	0	148	852	0	214	786	0	264	736
31	6	0	961	39	0	997	3	0	999	1	0	1000	0	0	1000	0
63	8	0	1000	0	0	1000	0	5	995	0	9	991	0	12	988	0
127	10	14	986	0	37	963	0	92	908	0	195	805	0	402	598	0
255	12	49	950	1	131	869	0	355	645	0	710	290	0	930	70	0
511	14	42	958	0	213	787	0	551	449	0	899	101	0	998	2	0
1023	16	21	973	6	126	874	0	470	530	0	925	75	0	1000	0	0
2047	18	**5**	**794**	**200**	35	965	0	265	735	0	830	170	0	998	2	0
4095	20	**0**	**369**	**586**	8	891	101	61	939	0	485	515	0	980	20	0
8191	22	8	992	0	**1**	**436**	**534**	11	919	70	130	870	0	717	283	0

[16] D. A. Simovici, M. Zimand, and D. Pletea, "Several remarks on index generation functions," *International Symposium on Multiple-Valued Logic* (ISMVL-2012), Victoria, Canada, May 2012, pp. 179-184.