D. L. Ostapko
S. J. Hong

Generating Test Examples for Heuristic Boolean Minimization

Abstract: This article describes simple methods of generating many-variable test-case problems for heuristic logic minimization studies. Covering problems and coloring problems are converted into Boolean functions that are useful test cases for minimization.

Introduction

There are many heuristic minimization procedures [1-4] and, no doubt, many more are being developed. In the evaluation of these heuristic Boolean minimization algorithms, one of the important criteria is the degree of minimality obtained. The degree of minimality, however, can be determined only by minimizing examples for which a minimum solution is known. As the number of input variables increases, it becomes increasingly difficult to find examples for which this minimum is known. For some functions, such as symmetric functions, the minimum can be calculated. However, most minimum solutions are obtained by actually minimizing the function with an alternate algorithm. In the following, we present two methods of generating Boolean functions with a large number of variables for which a minimum solution is known or easily determined. The functions generated are derived from the problems of selecting a minimum cover for a covering table and for an incompatibility table. By construction, any minimum solution for the defined Boolean function yields a solution to the corresponding covering problem.

Minimum cover of a covering table

Classical two-level Boolean minimization methods [5] usually proceed through two steps. First, all prime implicants of a function f are generated and a table of minterms (or vertices) covered by prime implicant is formed. This table has one column for each minterm of the function and one row for each prime implicant. Table 1 shows an example of a covering table in which a 1 in row p_i and column m_j means that prime implicant p_i covers minterm m_j.

For each covering table, a function g that describes the possible groupings of the minterms of f can be defined. Each variable in the defined function g represents a minterm of f that must be covered. For example, the function g that corresponds to the five-variable function in Table 1 has 12 variables corresponding to the 12 minterms of f that must be covered. In general, the minterms of f that must be covered are given by the m columns of the covering table. The DON'T CARE minterms of the function f are not included in the table or as variables of g because they need not be covered and their effect on the generation of prime implicants has already been considered. Since the function g is to describe the possible groupings of the minterms of f, the conditions governing the minterms of f must be expressed as constraints on the variables of g. In order to define the constraints on the variables of g, the correspondence between the variables of g and the minterms of f must be given. This correspondence is established by considering a minterm of g, $X_m = x_1 \cdots x_m$, as the characteristic vector of a set of minterms of f. The order of the list of minterms of f will be taken as that given by the columns of the covering table. For example, in Table 1, $X_k = 100110000000$ is the minterm of g that corresponds to the three minterms of f, 00000, 00100, and 10000. Thus, for X_m, a minterm of g, the corresponding set of minterms of f is $F(X_m) = \{m_i | x_i = 1\}$; and for $F = \{m_i\}$, a set of minterms of f, the corresponding minterm of g is $X_m(F) = x_1 \cdots x_m$, where $x_i = 1$ if $m_i \in F$ and $x_i = 0$ otherwise. The function g is defined to operate on the characteristic vector of a set of minterms of f. The value of g evaluated at the characteristic vector is defined to be 0 if the corresponding set of minterms cannot

Table 1 Covering table for a five-variable function $f(Y_1 \cdots Y_5)$.

Prime implicants p_i	Minterms m_j 00000	00001	00010	00100	10000	10001	01010	10010	00111	01111	10111	11111
$X0X00$	1			1	1							
$XX001$		1				1						
$XX010$			1				1	1				
$X0X11$									1		1	
$XX111$									1	1	1	1
$X00XX$	1	1	1		1	1		1				

DON'T CARE *minterms* $(y_1 \cdots y_5)$: 00011 01001 10100 10011 11001 11010

be combined, 1 if the set contains a single minterm, and DON'T CARE if the set can be combined. It will be shown that the number of implicants in the minimum representation of g gives the number of implicants in the minimum representation of f.

The definition of g requires that the set of minterms $I = \{\delta_{i1}\delta_{i2} \cdots \delta_{im} | 1 \leq i \leq m; \delta_{ij} = 0, i \neq j; \delta_{ii} = 1\}$ be mapped to 1. If there were no further restrictions, the minimum representation of g would map all minterms to 1 which corresponds to covering the entire space with one prime implicant. However, the rows of the table in Table 1 give the maximal sets of minterms of f that can be combined. These sets of minterms and all sets contained in them correspond to minterms of g that may be mapped to 1. Since all of these minterms need not be mapped to 1, they provide many DON'T CARE conditions. Thus, if D is the space formed by replacing the 1's in the table by DON'T CARE's, the DON'T CARE space of g is given as $DC = D \cdot \bar{I}$. The DON'T CARE space can be quite large and will always include the minterm $0 \cdots 0$. The regions I, D, and DC for the function g defined by Table 1 are given in Table 2. The covering problem thus becomes the problem of finding a minimum representation for the function g which maps the set of minterms I to 1 and has a DON'T CARE space DC.

The covering function g is an expanded representation of the function f. However, before discussing the relation between the minimum realizations of the two functions, the nature of the minimum realization of g will be considered.

Proposition 1 The minterms of f which correspond to an implicant of g are contained in one prime implicant of f.

Proof Let m_1 and m_2 be two minterms contained in a single implicant of g. Assume that the 1's in m_1 and m_2 correspond to minterms of f that are not contained in any one prime implicant of f. It follows from the definition of an implicant that $(m_1 + m_2)$, the bit-wise OR of m_1 and m_2, is also in the implicant of g. However, if the 1's in m_1 and m_2 are never mapped to the same prime implicant of f, the 1's in $(m_1 + m_2)$ will never be contained in a prime implicant of f. But, since $(m_1 + m_2)$ is in an implicant of g, the 1's in $(m_1 + m_2)$ map to minterms of f that can be combined and must therefore map into a prime implicant of f. This contradiction proves the proposition. The preceding proposition is useful in proving the following proposition which describes the prime implicants of the new function g.

Proposition 2 If $P = p_1p_2 \cdots p_m$ is a prime implicant of g, then $p_i \in \{0, X\}$, $1 \leq i \leq m$, where X means DON'T CARE.

Proof Assume $P = p_1p_2 \cdots p_m$ is a prime implicant of g and $p_j = 1$. Let $Q = \{i | p_i \neq 0\}$. By Proposition 1, the 1's in P correspond to minterms of f that are contained in one prime implicant of f. Thus, there exists an implicant $E = e_1e_2 \cdots e_m$ in D such that $e_i = X$, $i \in Q$. Because $D = DC + I$, every minterm in E is either mapped to 1 or is DON'T CARE, which means that none is required to be mapped to 0. Therefore, the minterms in $P' = p_1p_2 \cdots \bar{p}_j \cdots p_m$ are either mapped to 1 or are DON'T CARE, and P can be enlarged to $P'' = p_1p_2 \cdots p_{j-1}Xp_{j+1} \cdots p_m$. This contradicts the hypothesis that P is a prime implicant and proves the proposition.

The above proposition allows us to establish a mapping from the prime implicants of g into implicants of f. Each prime implicant of g will map into the implicant of f that contains all those minterms that correspond to X's in the prime implicant of g. In addition, it is possible to define a mapping from the prime implicatns of f into implicants of g that contain only 0's and X's. Each variable in the implicant of g will be a 0 or an X if the corresponding minterm is absent or present respectively in the prime implicant of f. The nature of these mappings is further described by the following proposition.

Proposition 3 There is a one-to-one correspondence between the prime implicants of f and the prime implicants of g.

Proof Let p be a prime implicant of f. Assume that the minterms of p map to an implicant of g that is not prime. Then, there must be another minterm in g that can be added to the set. By Proposition 1, this larger set must map into a prime implicant of f which contradicts the hypothesis that p is a prime implicant. Conversely if the minterms G of P a prime implicant of g do not form a prime implicant of f, they must by Proposition 1 be contained in a prime implicant. The set G can therefore be expanded which contradicts the hypothesis that P is prime, and thus the proposition follows.

Since a minimum cover of a function can always be expanded into a prime implicant cover, it follows from Proposition 3 that minimum covers of f and g should have the same number of implicants. Thus if the prime implicants of a Boolean function f are generated and covered by one algorithm, the Boolean function g can then be minimized by another algorithm. Although the second function has the same number of prime implicants, the number of variables and DON'T CARE conditions can be vastly different. Thus examples constructed in this manner should be useful in determining if a heuristic minimization program uses DON'T CARE conditions effectively. Table 3 shows minimum realizations for the five-variable function f and the 12-variable expanded function g. Using Table 1 and the mapping described above, the correspondence between the two solutions is easily seen.

Although the discussion above dealt with a prime implicant covering table, the method can be applied to a larger class of problems. The essential information provided by the table is the compatibility of the items to be covered. Thus any covering table in which the columns correspond to the items to be covered and the rows correspond to all collections that can be covered simultaneously may be used as a test example for a minimization technique provided that a minimum cover is separately calculated by some other method.

Table 2 Definition of g, the 12-variable expanded function that corresponds to the five-variable function, f.

I $(X_1 \cdots X_{12})$
100000000000
010000000000
001000000000
000100000000
000010000000
000001000000
000000100000
000000010000
000000001000
000000000100
000000000010
000000000001

a) The ON set of g.

D $(X_1 \cdots X_{12})$
X00XX0000000
0X000X000000
00X000XX0000
00000000X0X0
00000000XXXX
XXX0XX0X0000

b) The implicants of g that correspond to the prime implicants of f.

DC $(X_1 \cdots X_{12})$
000000001001
000000000110
000000001010
0000000011X0
00000000X101
000110000000
001000100000
00000000XX11
1001X0000000
00X000110000
010001000000
100001000000
001000010000
100000010000
001010000000
100010000000
011000000000
11000X000000
X1X000010000
X1X010000000
1X10X00X0000
XX1001000000
XXX011000000
XXX010010000
000000000000
XXX0X1010000

c) The DON'T CARE space of g.

Table 3 The minimum representations of the five-variable function f and the 12-variable expanded function g.

$(Y_1 \cdots Y_5)$	$(X_1 \cdots X_{12})$
X0X00	X00XX0000000
XX010	00X000XX0000
XX111	00000000XXXX
X00XX	XXX0XX0X0000
a) Minimum prime implicant representation of f.	b) Minimum prime implicant representation of g.

Minimum cover of an incompatibility table

The four-color problem has received considerable attention in the literature [6, 7]. In simple terms, this problem is stated as follows: Find the minimum number of colors necessary to color a map so as to avoid identical colors in lineally contiguous areas. This problem has come to be known as the four-color problem since it has been proven possible with five colors, and there are counter examples for three colors. To date, no map has been found which cannot be colored with four colors.

For a given map we shall construct a Boolean function h that has as many variables as there are regions in the map. The function will be specified by defining those minterms that the function must map to 1 and those minlike-terms that the function must map to 0. The function h may map to 1 any minterm whose 1's correspond to regions that can be colored with the same color. The important property of the function is that any represen-

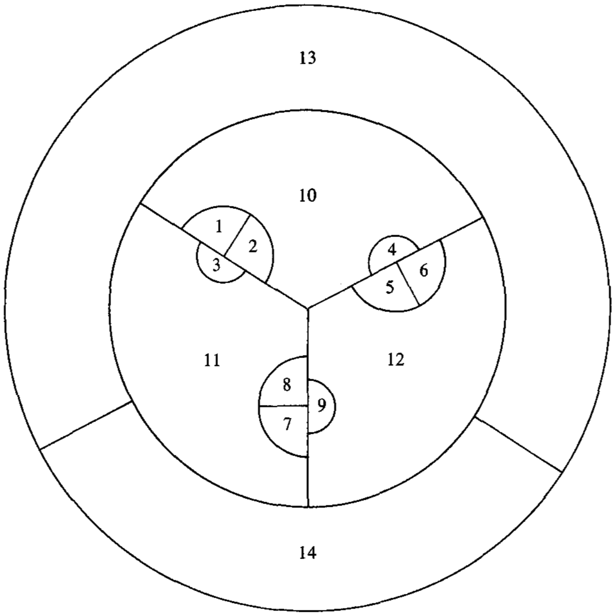


Figure 1 A 14-region map.

Table 4 The coloring function, h, for the map in Fig. 1.

I	H
10000000000000	1 1 XXXXXXXXXXXX
01000000000000	1 X1 XXXXXXXXXXX
00100000000000	1 XXXXXXXX1 XXXX
00010000000000	1 XXXXXXXXX1 XXX
00001000000000	X1 1 XXXXXXXXXXX
00000100000000	X1 XXXXXXX1 XXXX
00000010000000	X1 XXXXXXXX1 XXX
00000001000000	XX1 XXXXXXX1 XXX
00000000100000	XXX1 1 XXXXXXXXX
00000000010000	XXX1 X1 XXXXXXXX
00000000001000	XXX1 XXXXX1 XXXX
00000000000100	XXXX1 1 XXXXXXXX
00000000000010	XXXX1 XXXX1 XXXX
00000000000001	XXXX1 XXXXXX1 XX
	XXXXX1 XXX1 XXXX
a) The ON set of h.	XXXXX1 XXXXX1 XX
	XXXXXX1 1 XXXXXX
	XXXXXX1 X1 XXXXX
	XXXXXX1 XXX1 XXX
	XXXXXX1 XXXX1 XX
	XXXXXXX1 1 XXXXX
	XXXXXXX1 XX1 XXX
	XXXXXXX1 XXX1 XX
	XXXXXXXX1 XX1 XX
	XXXXXXXXX1 1 XXX
	XXXXXXXXX1 X1 XX
	XXXXXXXXX1 XX1 X
	XXXXXXXXXX1 1 XX
	XXXXXXXXXX1 X1 X
	XXXXXXXXXX1 XX1
	XXXXXXXXXXX1 1 X
	XXXXXXXXXXX1 X1
	XXXXXXXXXXXX1 1
	b) The OFF set of h.

tation of h defines a coloring of the map which has as many colors as the representation has implicants. Since each variable represents a region of the map, and taking the value 1 in the representation means that the region is colored, it is necessary that all the minterms with exactly one variable equal to 1 be mapped to 1.

Thus for a map with n regions, the coloring function h maps the set $I = \{\delta_{i1}\delta_{i2} \cdots \delta_{in} | 1 \le i \le \text{n}; \delta_{ij} = 0, i \neq j; \delta_{ii} = 1\}$ to 1. Again if there were no constraints, the minimum representation would map all minterms to 1 which corresponds to coloring all vertices with the same color. However, there are constraints which require that certain regions not be colored with the same color. In terms of the function h, this means that minterms that contain certain patterns of 1's must be mapped to 0. For example if regions 5 and 9 are contiguous, the bit-wise OR of $\delta_{51}\delta_{52} \cdots \delta_{5n}$ with $\delta_{91}\delta_{92} \cdots \delta_{9n}$ must not be contained in the representation or, alternatively, must not be mapped to 1 by h. Figure 1 shows a 14-region map that can't be colored with three colors. Table 4 defines the function h by specifying I, the set of minterms that is mapped to 1, and H, the set of terms that cannot be mapped to 1 and must therefore be mapped to 0. The remaining space $(\overline{h \vee H})$ is the DON'T CARE space for this minimization problem. The important property of the coloring function is expressed in the following proposition.

Proposition 4 Every region corresponding to a variable that has the value 1 or DON'T CARE in an implicant of the coloring function may be colored with the same color.

Proof Let regions i and j, $i < j$, be adjacent and let $S = S_1S_2 \cdots S_r$ be an implicant of the coloring function with S_i and S_j elements of $\{1, X\}$. If $S_i, S_j \in \{1, X\}$, then $S_kS_2 \cdots S_{i-1}1\ S_{j+1} \cdots S_{j-1}1\ S_{j+1} \cdots S_r$ is mapped to 1. However since i and j are adjacent, the implicant with $S_i = S_j = 1$ and $S_k = X$, $k \neq i, j$, is mapped to 0. Thus, the implicant S cannot be mapped to 1, which proves the proposition.

It should be observed that for a solution that consists of prime implicants, some variables may be allowed the value 1 in more than one implicant. Because a region can receive only one color, this requirement implies that there is a choice in selecting the color of that region given that all other regions are colored. Each region that can be assigned the value 1 in more than one implicant can be assigned independently of the other regions as long as each region is colored. The following proposition describes the conditions under which the minimum representation of h gives all possible grouping and hence all possible colorings of a map.

Proposition 5 The minimum representation of the coloring function h provides all possible colorings of the corresponding map if and only if the representation consists of k essential prime implicants.

Proof From Proposition 4 it follows that coloring a region of a map is equivalent to assigning the value 1 to the corresponding variable in one implicant of the representation of h. If the minimum representation of h provides all colorings, then it must contain all prime implicants of h. If in addition, this representation is minimal, then all prime implicants are required. Thus, the solution must consist of k essential prime implicants. Conversely, assume that a coloring of the map corresponds to k implicants that are not contained in the set of k essential prime implicants in the minimum representation of h. Each of the k implicants can be expanded to a prime implicant. This set of k prime implicants must differ from the original set because they contain the k implicants which are not contained in the original set. Thus there exist two distinct solutions consisting of k prime implicants, which contradicts the hypothesis that the original set is essential.

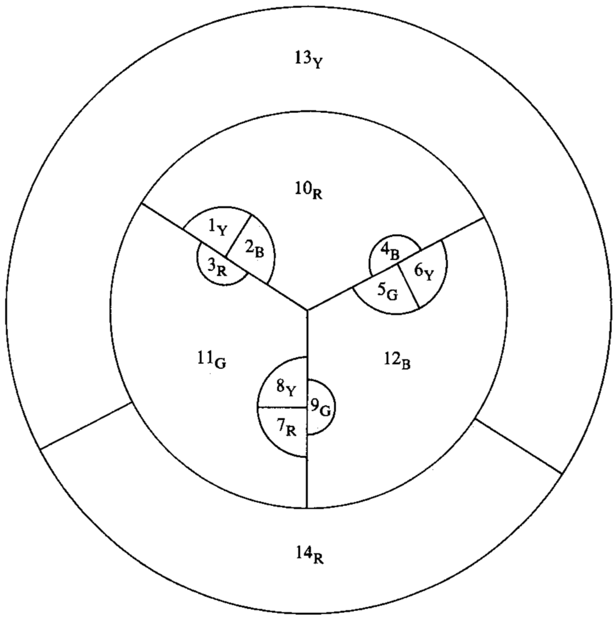
The nature of the prime implicants of the coloring function g are further described in the following proposition.

Proposition 6 If $P = p_1p_2 \cdots p_r$ is a prime implicant of the coloring function g, then $p_i \in \{0, X\}$, $1 \leq i \leq r$, where X means DON'T CARE.

Proof Assume $p_1p_2 \cdots p_r$ is a prime implicant of g and $p_i = 1$. Since the set of implicants that must be mapped to 0 do not intersect with $p_1p_2 \cdots p_{i-1}\ 1\ p_{i+1} \cdots p_r$, they will not intersect with $p_1p_2 \cdots p_{i-1}\ 0\ p_{i+1} \cdots p_r$, i.e., if a set of regions can be colored with the same color, then a subset of the region can be colored with the same color. Thus the prime implicant can be expanded to $p_1p_2 \cdots p_{i-1}Xp_{i+1} \cdots p_r$, a contradiction which proves the proposition.

Figure 2 shows a prime implicant solution for the function h defined in Table 4 and a resultant coloring for the map in Fig. 1.

The above procedure can be used to generate a limitless number of test problems with a known minimum solution size of four cubes. These problems have a very large DON'T CARE space, any desired number of input variables, and are easily generated. If a toroid is available, six-color problems can be generated with the same ease. The method presented is applicable to a class of problems that is larger than just the coloring problems. The approach is applicable to any problem in which it is desired to group objects in sets such that no defined pairwise incompatibility is violated. In those cases, it would be necessary to obtain a minimal set of covering groups by an alternate method if the minimality of the minimization algorithm were being tested.



0 0 X 0 0 0 X 0 0 X 0 0 0 X	Red:	3, 7, 10, 14
0 X 0 X 0 0 0 0 0 0 0 X 0 0	Blue:	2, 4, 12
0 0 0 0 X 0 0 0 X 0 X 0 0 0	Green:	5, 9, 11
X 0 0 0 0 X 0 X 0 0 0 0 X 0	Yellow:	1, 6, 8, 13

Figure 2 A minimum representation of h and a corresponding coloring of the map in Figure 1.

Conclusions

In this paper we have presented two methods of generating Boolean functions having a large number of variables. In both cases, the number of implicants in the minimum representation is known or is easily determined. The examples generated should be useful as test examples for heuristic minimization procedures since they can be constructed with a large number of variables and a relatively small minimum representation.

The minimization of the Boolean functions defined for the covering and incompatibility tables is a method of obtaining a minimum cover for any problem with these constraints. For reasonable size problems, this approach should be quite effective.

Although these examples were constructed as test examples for heuristic Boolean minimization algorithms, they provide information about some useful properties of minimization algorithms. First, it is often convenient to specify a function by specifying two of the three sets $f^{-1}(1), f^{-1}(0)$, and $f^{-1}(\phi)$, i.e., the ON set, the OFF set, and the DON'T CARE set. Thus, the program should have the ability to accept any two of these sets and still produce a minimum representation. In addition it should be observed that proper handling of the DON'T CARE space is necessary since neither of the above problems can be minimized properly if the DON'T CARE space is forced to be either ON or OFF. Finally, the three sets are often expressed conveniently as implicants rather than as minterms. Thus, the program should be capable of accepting this form of input, which should be faster, require less storage space, and reduce the probability of errors.

The authors have used the described method fruitfully in evaluating various heuristic Boolean minimization programs.

References

1. R. M. Bowman and E. S. McVey, "A Method for the Fast Approximate Solution of Large Prime Implicant Charts," *IEEE Trans. Comput.* **C-19,** 169 (1970).
2. C. C. Carroll, "Fast Algorithm for Boolean Function Minimization," AD680305, Project Themis, Auburn University, Auburn, Alabama, 1968 (for Army Missile Command, Huntsville, Alabama).
3. E. Morreale, "Recursive Operators for Prime Implicant and Irredundant Normal Form Determination," *IEEE Trans. Comput.* **C-19,** 504 (1970).
4. D. L. Starner, R. O. Leighon and K. H. Hill, "A Fast Minimization Algorithm for 16 Variable Boolean Functions," submitted to *IEEE Trans. Comput.*
5. E. J. McCluskey, Jr., "Minimization of Boolean Functions," *Bell Syst. Tech. J.* **35,** 1417 (1956).
6. O. Ore, *The Four-Color Problem,* Academic Press, New York, 1967.
7. T. L. Saaty, "Thirteen Colorful Variations on Guthrie's Four-Color Conjecture," *American Mathematical Monthly* **79,** 2 (1972).

Received November 10, 1973

D. L. Ostapko is located at the IBM System Products Division Laboratory in Poughkeepsie, New York 12602; S. J. Hong, a member of the staff of the IBM System Products Division Laboratory in Poughkeepsie, is on temporary assignment at the University of Illinois, Urbana, Illinois 61801.