On Constructing Karnaugh Maps

Don Thompson
Natural Science Division
Pepperdine University
Malibu, CA 90265

Abstract

Karnaugh [5] created a technique for finding the minimal sum of products form of a boolean function, a technique which is commonly included in the now popular Discrete Mathematics textbooks. Karnaugh maps are introduced in these texts as a topological bridge between boolean algebra and combinational networks, affording a technique for reducing the number of logic gates required to represent a boolean function. This paper will discuss a new, simple method for constructing Karnaugh maps recursively .

Introduction

With the advent of the Discrete Mathematics Course in the first two years curriculum, more dialogue is needed on how to teach the material. The purpose of this work is to clarify how Karnaugh maps of arbitrary size are constructed; a point which is usually handled cursorily in the Discrete Mathematics texts but which is somewhat surprising and provides an illuminating example of recursion.

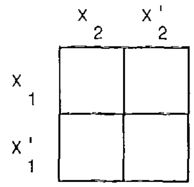
I. Definitions, Notation, Credits

This section reviews the well known pertinent definitions. Those familiar with Karnaugh maps are encouraged to take note of the figures and then skip to part II.

An n-variate boolean function $f(x_1,x_2,x_3,...,x_n)$ mapping $\{1,0\}^n$ to $\{1,0\}$ (where 1 = true and 0 = false), is said to be in sum of products form if f is of the form: $f = \Sigma_j \partial_j$, where each ∂_j is a product of complemented or uncomplemented members of $\{x_1,x_2,...,x_n\}$. In our notation, products are denoted by concatenation and stand for logical "and"; sums are denoted by the "+" symbol and stand for logical "or"; complements are denoted by the " ' " sumbol and stand for logical "not". A function is in canonical sum of products form if each ∂_j contains all n variables (each one complemented or uncomplemented). The function is in minimal sum of products form if it is in sum of products form and has fewer terms and fewer overall factors among the terms than any other sum of products form. It is the last form which is desired; it allows the number of logic gates in the corresponding combinational network to be minimized (at least with respect to sum of products form).

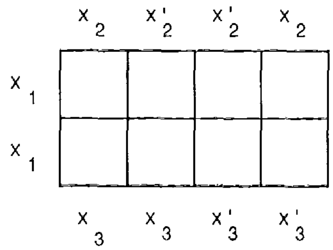
If $A = \{a_1, a_2, ..., a_n\}$ is a set of boolean expressions we say that the boolean expression b *covers* A precisely when b + a = b iff a A. Note that if n > 1, covering terms contain fewer variables than the terms they cover. For example, ab and ab' are *covered* by a; abc, abc',a'bc, and a'bc' are *covered* by b. A *prime implicant* is a minimal covering term, i.e. one which covers a maximal set of terms.

The point is that boolean functions can be reduced from their canonical sum of products form to minimal sum of products form if prime implicants can be extracted. Karnaugh maps provide a method for accomplishing this reduction. A Karnaugh map of order n is an array containing all 2^n possible canonical terms for an n-variate boolean function. (It amounts to a geometric layout of an n-variable truth table.) A more complete, constructive definition will be given below. The Karnaugh maps of order 2 and 3 are depicted in Figure 1. For the map of order 3, the indicated edges are to be viewed as joined together; as if the map forms a cylinder with these edges joined as a vertical seam.



Karnaugh Map of Order 2

Figure 1



Karnaugh Map of Order 3

Notice that in Figure 1, neighboring squares differ in exactly one coordinate, a requirement stemming from the fact that prime implicants can then be formed as contiguous blocks of cells containing 2^k members. Given a boolean function one constructs its Karnaugh Map as follows: First, bring the function to canonical sum of products form, then for each canonical term present in the function, place a 1 in the corresponding cell of the Karnaugh array.

The object of Karnaugh maps is to 'cover' all of the terms of the boolean function via prime implicants of maximal size. For example: consider the function $f(x_1,x_2,x_3) = x_1x_2x_3 + x_1'x_2x_3 + x_1'x_2'x_3 + x_1'x_2'x_3' + x_1x_2'x_3' + x_1x_2x_3'$ in conjunction with Figure 2. The prime implicants are x_2x_3, $x_1'x_3$, $x_1'x_2'$, $x_2'x_3'$, x_1x_2, and x_1x_3'.

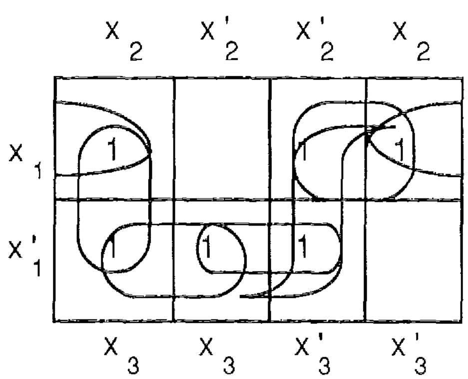


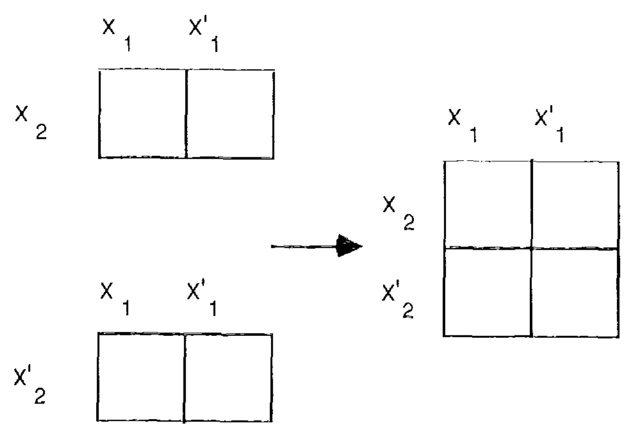
Figure 2

Clearly, an optimum covering is with x_2x_3, $x_1x'_3$, $x'_1x'_2$, hence a minimal sum of products form for f is $x_2x_3 + x_1x_3' + x_1'x_2'$. There exist several discussions of this process in the popular Discrete Mathematics texts, e.g. [1], [2], [3], [4], [6], and [8]. Another popular reduction method known as the Quine-McCluskey algorithm (see [7], [9]) is also usually presented in Discrete Mathematics texts, being the easiest to implement on a computer, owing to its non-geometric approach.

II. How to Build a Karnaugh Map of Order n

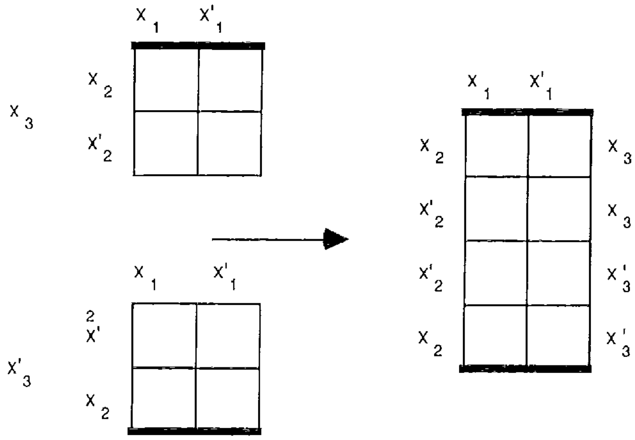
Our intent is to focus on the geometry and construction of Karnaugh Maps rather than on their algorithmic efficiency or the process of identifying prime implicants for a given boolean function.

The process of constructing Karnaugh Maps of arbitrary order is often left as a mystery in the standard Discrete Mathematics texts, whereas Karnaugh [5] hinted at but did not completely describe the process. The process is illustrated via Figures 3, 4, 5, and 6.



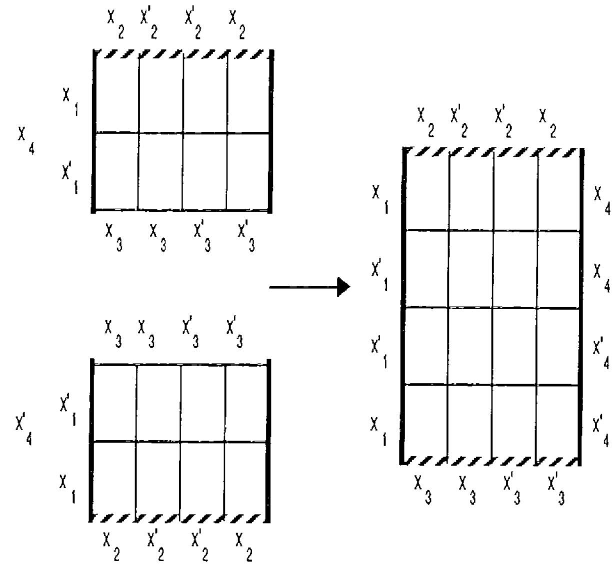
Two maps of order 1 are joined forming a map of order 2.

Figure 3



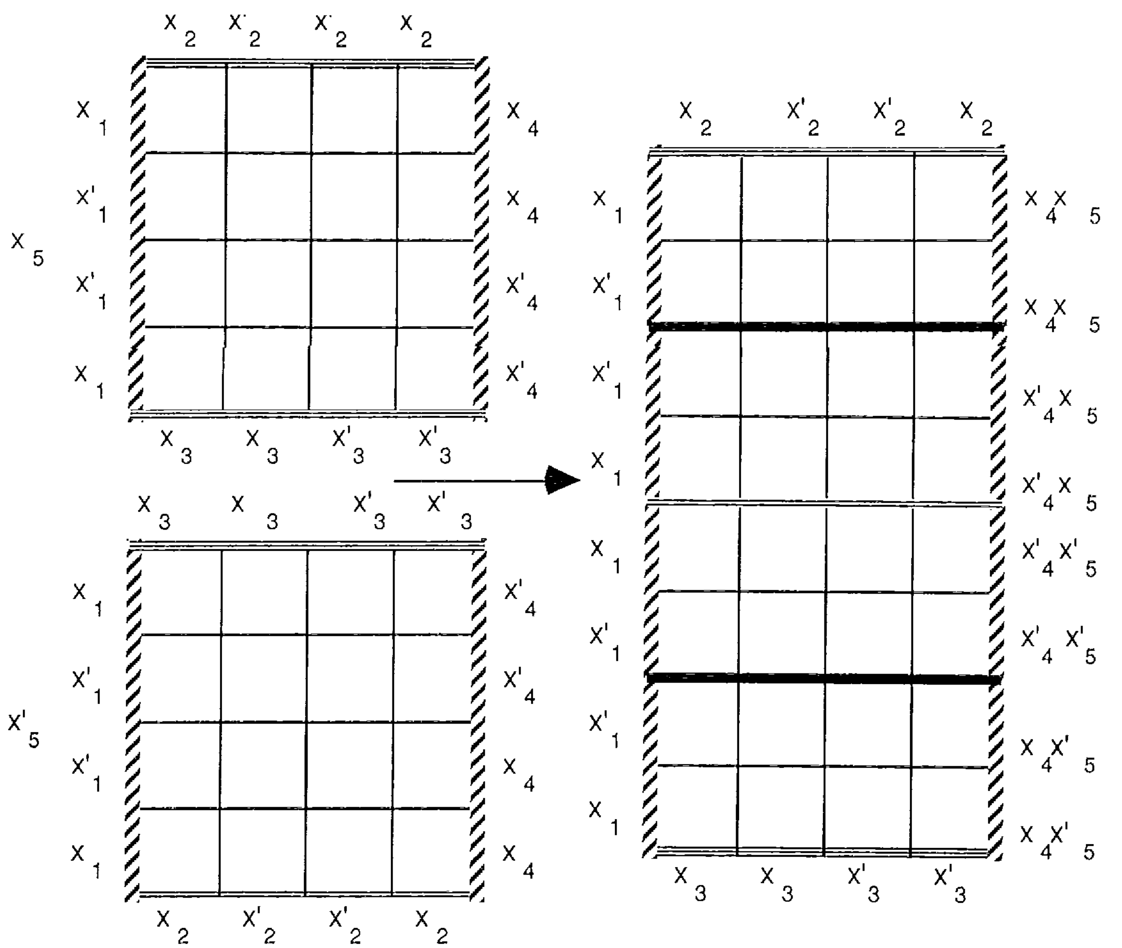
Two maps of order 2 are joined forming a map of order 3.

Figure 4



Two maps of order 3 are joined forming a map of order 4.

Figure 5

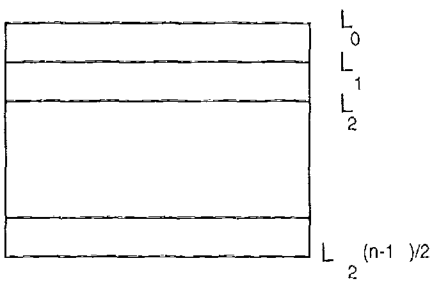


Two maps of order 4 are joined forming a map of order 5

Figure 6

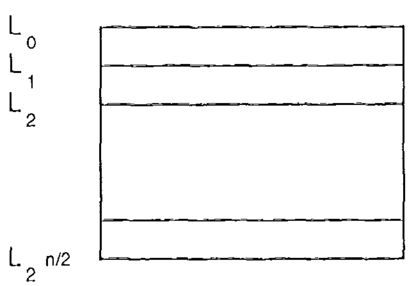
In each case, the edges with like shading are considered joined together. Notice that the construction of the map of order 5 marks a turning point. The requirement that each cell be a neighbor to all cells that differ in exactly one coordinate is now a problem, for each cell needs 5 neighbors but has only 4 sides. In this regard let us introduce a coordinatization scheme.

Let M(n) be a Karnaugh map of order n. M(n) can be viewed as a grid which is $2^{n/2}$ by $2^{n/2}$ if n is even, $2^{(n-1)/2}$ by $2^{(n+1)/2}$ if n is odd. For odd n, label the lines of the grid running lengthwise by L_0, L_1, L_2,...$L_{2^{(n-1)/2}}$. For even n, label the lines of the grid in one of the directions (either one is suitable) by L_0,L_1, ...,$L_{2^{(n/2)}}$, see Figure 7.



n odd

Figure 7



n even

We are now prepared to describe the method for recursively constructing Karnaugh Maps. To construct M(n+1) (n $\geq$1) take two copies of M(n); label the first copy as x_{n+1}, the second copy as x'_{n+1} with each map coordinatized as described above and lines labelled with or without a prime according to whether one is referring to copy 1 or 2, respectively. Glue lines together as follows: Let m = $2^{n/2}$, if n is even; $2^{(n-1)/2}$, if n is odd. Glue L_0 to L_m, L_0' to L_m', L_j to L'_{m-j}; $0 \leq j \leq$ m, j even. See Figure 8.

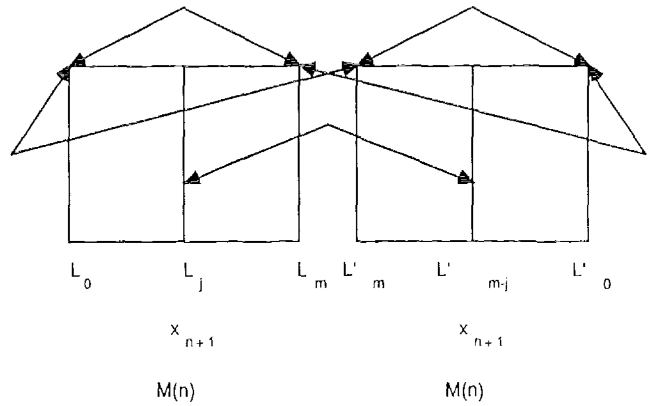


Figure 8

Acknowledgement:

I would like to thank Patricia Baron, one of my students, who assisted me in this project.

References

1. Dierker, Paul and Voxman, William
Discrete Mathematics, Harcourt Brace Jovanovich, 1986.

2. Gersting, Judith
Mathematical Structures, for Computer Science, W.H. Freeman and Company, 1982.

3. Grimaldi, Ralph
Discrete and Combinatorial Mathematics, Addison-Wesley, 1985.

4. Johnsonbaugh, R.
Discrete Mathematics, MacMillan, 1984.

5. Karnaugh, M.
"The Map Method for Synthesis of Combinational Logic Circuits" Transactions of the AIEE, Part I, Vol. 72, No. 9, pp. 593-599, 1953.

6. Korfhage, Robert
Discrete Computational Structures, Academic Press, 1984.

7. McCluskey, E.J.
"Minimization of Boolean functions" Bell System Technical Journal, Vol. 35, nol. 6, pp. 1417-1444, November 1956.

8. Norris, Fletcher
Discrete Structures: An Introduction to Mathematics for Computer Scientists, Prentice-Hall, 1985.

9. Quine, W.V.
"A Way to Simplify Truth Functions"
American Mathematical Monthly, vol. 62. no. 9, pp. 627-631, November 1955.