Optimal Decision Tree Size vs. Minimal Disjoint CDNF size for Boolean Funcions of Degree 3

Livia Overand

June 29, 2005

1 Introduction

Bshouty, Bisht, and Khoury show a specific set of assignments that can be isolated by a disjoint CDNF of size 5 but that requires a decision tree of size at least 6. This example proves Bshouty's claim that there is a set of assignments S that has a complete set that isolates it of size s but any complete set that isolates S that corresponds to a decision tree is of size $\Omega(s^{\frac{\log 6}{\log 5}})$. I claim that there does not exist any other sets of assignments on three variables that would improve Bshouty's result. Furthermore, there are exactly 8 sets of assignments on three variables that can be isolated by a minimal disjoint CDNF of size 5 and an optimal decision tree of size 6. All other sets of assignments on three variables can be isolated by a minimal disjoint CDNF and an optimal decision tree of equal sizes. Although Bshouty's claim as stated considers sets of assignments, it is equivalent to consider Boolean functions. That is to say, the number of Boolean functions on three variables is equal to the number of sets of assignments on three variables. My claim can be proved by exhaustively considering all Boolean functions on 3 variables and determining the optimal decision tree size and the minimal disjoint CDNF size of the functions.

2 Definitions

A *Karnaugh map* (K-map) is a graphical representation of a boolean function which is particularly helpful in minimizing the DNF representation of a function. A K-map in three variables is a rectangle divided into eight cells. The cells represent the eight possible minterms in three variables. A 1 is placed in the cell if the value of the function is 1 for the corresponding minterm. Two cells are said to be adjacent if the minterms that they represent differ in exactly one literal.

A *disjoint DNF* is a DNF whose terms are disjoint, that is, the conjunction of every two terms in the DNF is 0.

A *disjoint CDNF* is a pair of disjoint DNF (P,Q) such that the $\bar{P} = Q$. The *size* of a disjoint CDNF (P,Q) is the total number of terms in P and Q. Consequently, a boolean function can be expressed as a disjoint CDNF where a disjoint DNF, P, has the value 1 when the function equals 1, and where a disjoint DNF, Q, has the value 1 when the function equals 0. A *minimal disjoint CDNF* has the fewest number of terms as any other possible disjoint CDNF representation.

A *decision tree* is a rooted binary tree whose internal nodes are labeled with variables $\{x_1, x_2, \ldots, x_n\}$ and whose leaves are labeled with constants 0,1. Each internal node has precisely two outgoing edges, where the right edge represents the assignment 1, and the left edge represents the assignment 0. The *size* of a decision tree is the number of leaves of the tree. An *optimal decision tree* has the fewest number of leaves of any possible representation of a tree.

The representation of a Boolean function as a K-map relates to its representation as a disjoint CDNF in the following way. Each cell or block of cells containing a 1 in a K-map is represented by its corresponding product of literals in exactly one term of a disjoint DNF, P. Each cell or block of cells not containing a 1 in a K-map is represented by its corresponding product of literals in exactly one term of a disjoint DNF, Q.

The representation of a Boolean function as a K-map relates to its representation as a decision tree in the following way. Each cell or block of cells containing a 1 in a K-map corresponds to a path in a decision tree that results in a 1 leaf. Each cell or block of cells not containing a 1 in a K-map corresponds to a path in a decision tree that results in a 0 leaf. More specifically, for Boolean functions of degree 3, each cell in a K-map containing a 1 that cannot be grouped into a block of cells corresponds to a path in a decision tree that results in a 1 leaf at depth at least three. Each block of two cells containing 1's in a K-map corresponds to a path in a decision tree that results in a 1 leaf at depth at least two. Each block of four cells containing 1's in a K-map corresponds to a path in a decision tree that results in a 1 leaf at depth at least one.

The representation of a Boolean function as a disjoint CDNF relates to its representation as a decision tree in the following way. For each term in the disjoint DNF, P, the corresponding path in a decision tree will result in a 1 leaf. For each term in the disjoint DNF, Q, the corresponding path in a decision tree will result in a 0 leaf.

Figure 1 is an example of a K-map. The Boolean function represented in figure 1 corresponds to the DNF expression $x_2x_3 + x_1x_2$, the disjoint DNF expression $x_1x_2 + \bar{x_1}x_2x_3$, and the minimal disjoint CDNF expression $(x_1x_2 + \bar{x_1}x_2x_3,\ \bar{x_2} + \bar{x_1}x_2\bar{x_3})$. The cells representing the minterms $x_1x_2x_3$ and $\bar{x_1}x_2x_3$ are adjacent. Figure 2 shows an optimal decision tree representation of the same Boolean function represented in figure 1. Note that the size of the optimal decision tree and the minimal disjoint CDNF are both 4.

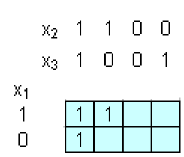


Figure 1: K-map representation of $x_2x_3 + x_1x_2$

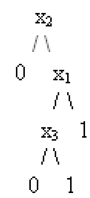


Figure 2: optimal decision tree representation of $x_2x_3 + x_1x_2$

3 Methodology

I determined the minimal disjoint CDNF sizes for all Boolean functions on three variables by using Karnaugh maps (K-maps). All $2^{2^3} = 256$ Boolean functions on three variables can be considered by taking all possible combinations of a 1 appearing in the 0-8 cells of a K-map. It is not necessary, however, to look at 256 different K-maps. First, this number can be cut in half by observing that a Boolean function and its complement have the same CDNF representation and the same decision tree representation except with 0 and 1 leaves exchanged. For example, the function corresponding to figure 1 has minimal disjoint CDNF representation $(x_1x_2 + \bar{x_1}x_2x_3,\ \bar{x_2} + \bar{x_1}x_2\bar{x_3})$ and optimal decision tree representation as depicted in figure 2. The complement of this function has CDNF representation $(\bar{x_2} + \bar{x_1}x_2\bar{x_3},\ x_1x_2 + \bar{x_1}x_2x_3)$ and decision tree representation as depicted in figure 3. Consequently, each K-map considered can account for two Boolean functions.

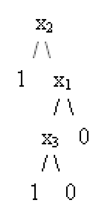


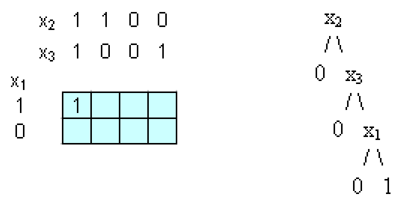
Figure 3: optimal decision tree representation for the complement of $x_2x_3 + x_1x_2$

The second way I was able to reduce the number of K-maps necessary to consider all Boolean functions on three variables was by observing that there are certain characteristics that always result in the same case of CDNF size and decision tree size. The determining characteristics are the number of cells containing a 1, the adjacency or nonadjacency of cells, the number of cells adjacent to a given cell, and the number of literals that differ between cells. Using this classification of K-maps I was able to determine 14 specific cases of CDNF size and decision tree size that accounts for all 256 Boolean functions on three variables.

4 Cases

4.1 Case 1

There are 8 Boolean functions on three variables represented by a K-map with only one cell that contains a 1. Likewise, there are 8 Boolean functions on three variables represented by a K-map with only one cell that does not contain a 1. These 16 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 4. Figure 4 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.



$$(x_1x_2x_3,\ \bar{x_2} + x_2\bar{x_3} + \bar{x_1}x_2x_3)$$

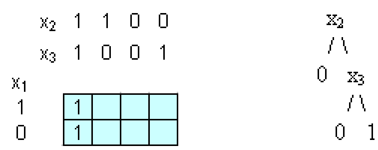
Figure 4: Case 1 K-map, decision tree, and disjoint CDNF

4.2 Case 2

Case 2 Boolean functions are represented by a K-map with two adjacent cells that contain a 1. There are 12 of these functions on three variables. In order to verify this, note that there are 3 ways to choose which variable differs between the two cells, $\frac{2^1}{2}$ ways to assign a value to this variable, and 2^2 ways to assign values to the remaining to variables. Likewise, there are 12 Boolean functions on three variables represented by a K-map with two adjacent cells that do not contain a 1. These 24 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 3. Figure 5 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

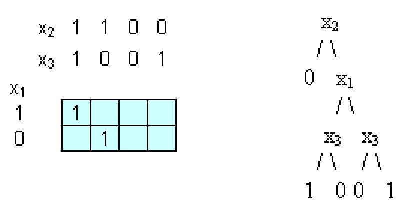
4.3 Case 3

Case 3 Boolean functions are represented by a K-map with two nonadjacent cells, whose corresponding minterms differ in only 2 literals, that contain a 1. There are 12 of these functions on three variables. In order to verify this, note that there are 3 ways to choose which variable agrees between the two cells, 2 ways to assign a value to this variable, and $\frac{2^2}{2}$ ways to assign values to the other variables. Likewise, there are 12 Boolean functions on three variables represented by a K-map with two nonadjacent cells, whose corresponding minterms differ in only 2 literals, that do not contain a 1. These 24 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 5. Figure 6 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.



$$(x_2 x_3,\ \bar{x_2} + x_2 \bar{x_3})$$

Figure 5: Case 2 K-map, decision tree, and disjoint CDNF

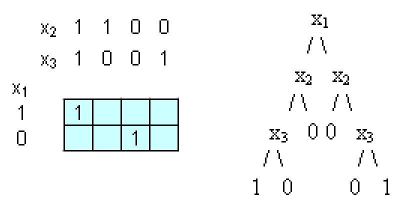


$$(x_1 x_2 x_3 + \bar{x_1} x_2 \bar{x_3},\ \bar{x_2} + x_1 x_2 \bar{x_3} + \bar{x_1} x_2 x_3)$$

Figure 6: Case 3 K-map, decision tree, and disjoint CDNF

4.4 Case 4

Case 4 Boolean functions are represented by a K-map with two nonadjacent cells, whose corresponding minterms differ in all 3 literals, that contain a 1. There are 4 of these functions on three variables. In order to verify this, note that there are $\frac{2^3}{2}$ ways to assign values to three variables in two cells which are complements of each other. Likewise, there are 4 Boolean functions on three variables represented by a K-map with two nonadjacent cells, whose corresponding minterms differ in all 3 literals, that do not contain a 1. These 8 functions correspond to a minimal disjoint CDNF of size 5 and to an optimal decision tree of size 6. These are the only 8 Boolean functions on 3 variables whose minimal CDNF size differs from its optimal tree size. Figure 7 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function. Note that the assignments corresponding to the terms $x_1\bar{x_3}$ and $\bar{x_2}x_3$ are computed by the decision tree over three variables.



$$(x_1x_2x_3 + \bar{x_1}\bar{x_2}\bar{x_3},\ \bar{x_1}x_2 + x_1\bar{x_3} + \bar{x_2}x_3)$$

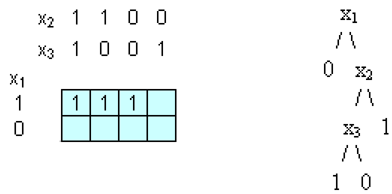
Figure 7: Case 4 K-map, decision tree, and disjoint CDNF

4.5 Case 5

Case 5 Boolean functions are represented by a K-map with three adjacent cells that contain a 1. There are 24 of these functions on three variables. In order to verify this, note that there are 3 ways to choose which variable is the same in all three cells, 2 ways to assign a value to this variable, and $\binom{2^2}{3}$ ways to assign values to the other variables. Likewise, there are 24 Boolean functions on three variables represented by a K-map with three adjacent cells that do not contain a 1. These 48 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 4. Figure 8 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

4.6 Case 6

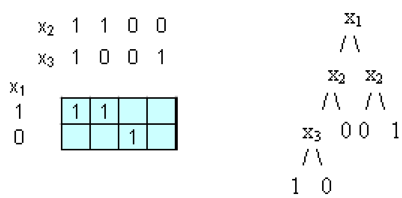
Case 6 Boolean functions are represented by a K-map with two adjacent cells and one nonadjacent cell that contain a 1. There are 24 of these functions on three variables. In order to verify this, note that there are 3 ways to choose which variable differs between the two adjacent cells, $\frac{2^1}{2}$ ways to assign a value to this variable, and 2^2 ways to assign values to the remaining variables in the adjacent cells. Once the two adjacent cells are fixed, two of the variables in the nonadjacent cell are determined because they must be the complement



$$(x_1x_2 + x_1\bar{x_2}\bar{x_3},\ \bar{x_1} + x_1\bar{x_2}x_3)$$

Figure 8: Case 5 K-map, decision tree, and disjoint CDNF

of the two variables that agree in the adjacent cells. There are 2 ways to assign a value to the third variable in the nonadjacent cell. Likewise, there are 24 Boolean functions on three variables represented by a K-map with two adjacent cells and one nonadjacent cell that do not contain a 1. These 48 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 5. Figure 9 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

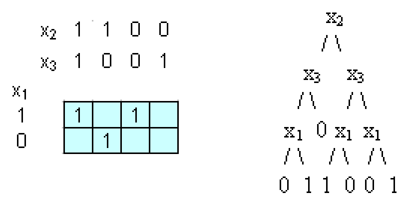


$$(x_1x_2 + \bar{x_1}\bar{x_2}\bar{x_3},\ \bar{x_1}x_2 + x_1\bar{x_2} + \bar{x_1}\bar{x_2}x_3)$$

Figure 9: Case 6 K-map, decision tree, and disjoint CDNF

4.7 Case 7

Case 7 Boolean fuctions are represented by a K-map with three nonadjacent cells that contain a 1. There are 8 of these functions on three variables. In order to verify this, note that there are 8 ways to choose the first cell, and then $\binom{3}{2}$ ways to choose the remaining two cells. Take the product of the number of ways to choose these cells and then divide by 3 in order to count each combination only once. Likewise, there are 8 functions on three variables represented by a K-map with three nonadjacent cells that do not contain a 1. These 16 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 7. Figure 10 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

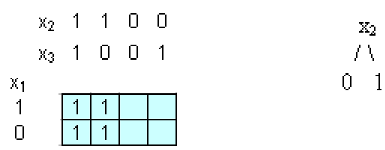


$$(x_1x_2x_3 + \bar{x_1}x_2\bar{x_3} + x_1\bar{x_2}\bar{x_3},\ \bar{x_1}x_2x_3 + x_1x_2\bar{x_3} + \bar{x_1}\bar{x_2}\bar{x_3} + \bar{x_2}x_3)$$

Figure 10: Case 7 K-map, decision tree, and disjoint CDNF

4.8 Case 8

Case 8 Boolean functions are represented by a K-map with four adjacent cells, whose corresponding minterms all share a common literal, that contain a 1. There are 6 of these functions on three variables. In order to verify this, note that four adjacent cells can be represented as a single variable. There are 3 ways to choose this variable and 2 ways to assign its value. These 6 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 2. Figure 11 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

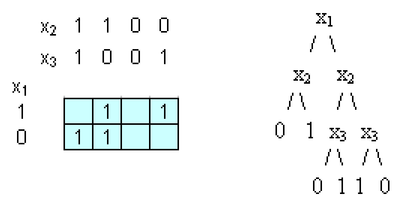


$$(x_2,\ \bar{x_2})$$

Figure 11: Case 8 K-map, decision tree, and disjoint CDNF

4.9 Case 9

Case 9 Boolean functions are represented by a K-map with three adjacent cells, whose corresponding minterms all share a common literal, and one nonadjacent cell that contain a 1. There are 24 of these functions on three variables. In order to verify this, note that for three adjacent cells to occur there must be one cell that is adjacent to two other cells. The other two cells are not adjacent to each other, but are connected by the first cell. To have a fourth cell that is not adjacent to the first three, the fourth cell must differ in at least two variables from any other cell containing a one. The only way for this to occur is if the fourth cell is the complement of the cell that is adjacent to the two others. Therefore, the number of functions for this case is equal to the number of functions that are represented by a k-map with three adjacent cells. These 24 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 6. Figure 12 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.



$$(\bar{x_1}x_2 + x_1x_2\bar{x_3} + x_1\bar{x_2}x_3,\ x_1x_2x_3 + \bar{x_1}\bar{x_2} + x_1\bar{x_2}\bar{x_3})$$

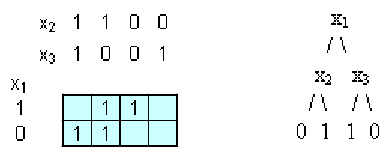
Figure 12: Case 9 K-map, decision tree, and disjoint CDNF

4.10 Case 10

Case 10 Boolean functions are represented by a K-map with four cells that contain a 1, where two of the cells are adjacent to two cells containing a 1, and the other two cells are adjacent to one cell containing a 1. There are 24 of these functions on three variables. In order to verify this, consider the two cells that are adjacent to each other and each adjacent to another cell. There are 3 ways to choose the variable that they do not share in common, $\frac{2^1}{2}$ ways to assign a value to this variable, and 2^2 ways to assign vaules to the other two variables in each cell. Next, condsider one of these cells. For this cell to be adjacent to a second cell it must differ in a different variable than from the first adjacent cell. There are 2 ways to choose this variable. Finally, there is only one remaining possibility for the fourth cell. These 24 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 4. Figure 13 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

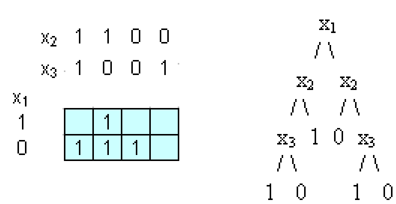
4.11 Case 11

Case 11 Boolean functions are represented by a K-map with four cells that contain a 1, where three of the cells are adjacent to one cell containing a 1, and the other cell is adjacent to three other cells containing a 1. There are 8 of these functions on three variables. In order to verify this, note that there are 8 ways to choose the cell that is adjacent to three others. Since any cell can be adjacent to at most three other cells, once the first cell is chosen there is only one way to place the other three. These 8 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 6. Figure 14 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.



$$(\bar{x_1}x_2 + x_1\bar{x_3},\ \ x_1x_3 + \bar{x_1}\bar{x_2})$$

Figure 13: Case 10 K-map, decision tree, and disjoint CDNF

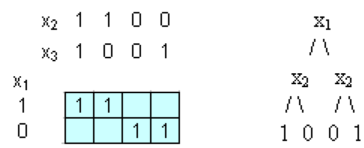


$$(\bar{x_1}x_2 + x_1x_2\bar{x_3} + \bar{x_1}\bar{x_2}\bar{x_3},\ \ x_1x_2x_3 + x_1\bar{x_2} + \bar{x_1}\bar{x_2}x_3)$$

Figure 14: Case 11 K-map, decision tree, and disjoint CDNF

4.12 Case 12

Case 12 Boolean functions are represented by a K-map with four cells consisting of two disjoint pairs of adjacent cells that contain a 1. There are 6 of these functions on three variables. In order to verify this, note that the number of functions for this case is equal to the number of functions that are represented by a k-map with two adjacent cells divided by two, since the pairs of adjacent cells are complements of each other. These 6 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 4. Figure 15 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.

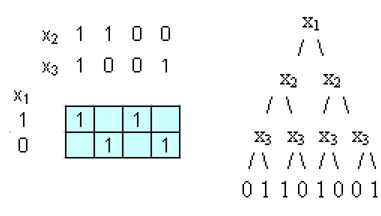


$$(x_1x_2 + \bar{x_1}\bar{x_2},\ \bar{x_1}x_2 + x_1\bar{x_2})$$

Figure 15: Case 12 K-map, decision tree, and disjoint CDNF

4.13 Case 13

Case 13 Boolean functions are represented by a K-map with four nonadjacent cells that contain a 1. There are 2 of these functions on three variables. In order to verify this, note that given a fixed cell there is only one way to then choose three other cells that each agree with the first cell in only one variable. So there are $\frac{2^3}{4}$ ways to choose these cells. These 2 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 8. Figure 16 shows the K-map, minimal disjoint CDNF, and optimal decision tree for one such function.



$$(x_1x_2x_3 + \bar{x_1}x_2\bar{x_3} + x_1\bar{x_2}\bar{x_3} + \bar{x_1}\bar{x_2}x_3,\ \bar{x_1}x_2x_3 + x_1x_2\bar{x_3} + \bar{x_1}\bar{x_2}\bar{x_3} + x_1\bar{x_2}x_3)$$

Figure 16: Case 13 K-map, decision tree, and disjoint CDNF

4.14 Case 14

There is 1 Boolean function on three variables represented by a K-map with all 8 cells containing a 1. Likewise, there is 1 Boolean function on three variables represented by a K-map with all 8 cells not containing a 1. These 2 functions correspond to a minimal disjoint CDNF and to an optimal decision tree of size 1. The disjoint CDNF for the always true function is $(1, \emptyset)$. The decision tree for the always true function is a single node labled 1. Figure 17 shows the K-map for the always true function.

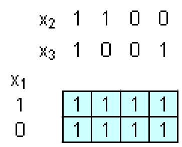


Figure 17: Case 14 K-map

5 conclusion

Bshouty shows that there is a polynomial gap between the minimal size of of a complete set of terms that isolates S and the minimal size of a complete set of terms that corresponds to a decision tree that isolates S. The 14 cases I have previously outlined account for all 256 sets of assignments on three variables and thus proves my claim that if any set of assignments on three variables can be isolated by a minimal disjoint CDNF of size s, then the size of its corresponding optimal decision tree is at most $s^{\frac{\log 6}{\log 5}}$. Using K-maps as a means to compare the minimal disjoint CDNF representation and the optimal decision tree representation of Boolean functons on three variables has made it easy to see why this polynomial gap in sizes occurs.

The K-map representation of Case 4 Boolean functions is characterized by two nonadjacent cells containing a 1, whose corresponding minterms differ in all three literals. Consequently the corresponding optimal decision tree must have two seperate paths each resulting in a single 1 leaf at depth three. The remaining four edges must lead to 0 leaves. This yields an optimal decision tree of size 6. In relation to the minimal disjoint CDNF representation, each nonadjacent cell containing a 1 corresponds to a term of three literals in the disjoint DNF, P. The empty cells, however, can be covered by three blocks of two cells. Each block corresponds to a term of two literals in the disjoint DNF, Q. This yields a minimal disjoint CDNF of size 5.

I hypothesize that the results are quite the same for Booelan functions of degree greater than three. That is to say, I believe that on all Boolean functions of degree n there is a gap between the size of a minimal disjoint CDNF and the size of an optimal decision tree whenever the corresponding K-map has at least two nonadjacent cells that differ in at least three literals that contain a one.