Chapter 4 Function Minimization Algorithms (I)

Overview (1/2)

- In this chapter, we will look at two approaches to finding all of the prime implicants of a function and then algorithms for finding minimum sum of products solutions
- The first approach to finding prime implicants is referred to as the Quine-McCluskey method. It starts with minterms and uses, repeatedly, the adjacency property

$$ab+ab'=a$$

- The second approach is *iterated consensus*. It starts with any set of terms that covers the function and uses the consensus operation and the absorption property

$$a+ab=a$$

Overview (2/2)

- Each of these methods has been computerized and is effective for a larger number of variables than the Karnaugh map, although the amount of computing becomes excessive for many practical problems

Quine-McCluskey Method for One Output (1/9)

- The Quine-McCluskey method is used to find all of the prime implicants of a function, which can be used to find the minimum sum of products expression(s) for that function
- The Quine-McCluskey algorithm was developed by W. V. Quine and Edward J. McCluskey in the 1950s
- Consider the function of Example 3.6:

$$f(w,x,y,z)=\Sigma m(0,4,5,7,8,11,12,15)$$

Quine-McCluskey Method for One Output (2/9)

- Our initial list, grouped by the number of 1's, is

		Numeric
A	0 0 0 0	0

B	0 1 0 0	4
C	1 0 0 0	8

D	0 1 0 1	5
E	1 1 0 0	12

F	0 1 1 1	7
G	1 0 1 1	11

H	1 1 1 1	15

where we have labeled the terms for easy reference

Quine-McCluskey Method for One Output (3/9)

- We now apply the adjacency property to each pair of terms. We need only consider terms in consecutive groups and produce a second column of terms with one variable missing:

$A + B = J = 0 - 0\ 0$ (where the dash represents a missing variable)

$A + C = K = -\ 0\ 0\ 0$

$B + D = L = 0\ 1\ 0\ -$

$B + E = M = -\ 1\ 0\ 0$

$C + D =$ none

$C + E = N = 1 - 0\ 0$

$D + F = O = 0\ 1 - 1$

$D + G =$ none

$E + F =$ none

$E + G =$ none

$F + H = P = -\ 1\ 1\ 1$

$G + H = Q = 1 - 1\ 1$

Quine-McCluskey Method for One Output (4/9)

- Some pairs of terms, even in adjacent groups, cannot be combined because they differ in more than one place, such as terms *C* and *D*
- Whenever a term is used to produce another term, it is checked off; it is not a prime implicant

Table 4.1 Quine-McCluskey prime implicant computation.

A 0000√	*J* 0–00√	*R* ––00
-------------	*K* –000√	
B 0100√	-------------	
C 1000√	*L* 010–	
-------------	*M* –100√	
D 0101√	*N* 1–00√	
E 1100√	-------------	
-------------	*O* 01–1	
F 0111√	-------------	
G 1011√	*P* –111	
-------------	*Q* 1–11	
H 1111√		

Quine-McCluskey Method for One Output (5/9)

- We now repeat the process with the second column. Again, we need only consider terms in consecutive sections of that column. Also, we need only consider terms with dashes in the same position, since they are the only ones with the same three variables. Thus, we find

$$J + N = R = - - 0\ 0$$
$$K + M = R \quad \text{(same term)}$$

- There are no adjacencies between the second and the third group or between the third and fourth group. Since there is only one term in the third column, we are done

Quine-McCluskey Method for One Output (6/9)

- This process is like combining adjacent minterms into a larger implicant. We repeat the process with the second column. Again, we need only consider terms in consecutive sections of the that column (number of 1's differing by only one). This process ends till all the feasible implicants are combined
- The prime implicant are

L	0	1	0	–	$w'xy'$
O	0	1	–	1	$w'xz$
P	–	1	1	1	xyz
Q	1	–	1	1	wyz
R	–	–	0	0	$y'z'$

Quine-McCluskey Method for One Output (7/9)

- This process works for larger number of variables, but the number of minterms and other implicants can increase rapidly. This process has been computerized
- If there are don’t cares in the problem, all of them must be included in the first column of the table, since don’t cares are part of prime implicants

Quine-McCluskey Method for One Output (8/9)

- **Example 4.1** Find the prime implicants of the function

$$g(w,x,y,z)=\Sigma m(1,3,4,6,11)+\Sigma d(0,8,10,12,13)$$

- The process proceeds as before

0000√	000–	––00
--------	0–00√	
0001√	–000√	
0100√	--------	
1000√	00–1	
--------	01–0	
0011√	–100√	
0110√	10–0	
1010√	1–00√	
1100√	--------	
--------	–011	
1011√	101–	
1101√	110–	

Quine-McCluskey Method for One Output (9/9)

- **Example 4.1 (Cont'd)** The prime implicants are

$w'x'y'$	$x'yz$
$w'x'z$	$wx'y$
$w'xz'$	wxy'
$wx'z'$	$y'z'$

- Notice wxy' and $wx'z'$ are prime implicants, they consist of all don't cares and would never be used in a minimum solution

Iterated Consensus for One Output (1/12)

- In this section, we will use the iterated consensus algorithm to list all of the prime implicants of a function
- We will first define the relationship *included in*. Product term t_1 is *included in* product term t_2 (written $t_1 \leqq t_2$) if t_2 is 1 whenever t_1 is 1 (and elsewhere, too, if the two terms are not equal)
- It means that t_1 is a subgroup of t_2. If an implicant t_1, is included in another implicant t_2, then t_1 is not a prime implicant since

$$t_1 + t_2 = xt_2 + t_2 = t_2$$

Iterated Consensus for One Output (2/12)

- The iterated consensus algorithm for single functions is as follows:

1. Find a list of product terms (implicants) that cover the function. Make sure that no term is equal to or included in any other term on the list. (The terms on the list could be prime implicants or minterms or any other set of implicants. However, the rest of the algorithm proceeds more quickly if we start with prime implicants.)
2. For each pair of terms, t_i and t_j (including terms added to the list in step 3), compute $t_i ¢ t_j$

Iterated Consensus for One Output (3/12)

- **Iterated consensus algorithm (*Cont'd*)**

3. If the consensus is defined, and the consensus term is not equal to or included in a term already on the list, add it to the list
4. Delete all terms that are included in the new term added to the list
5. The process ends when all possible consensus operations have been performed. The terms remaining on the list are ALL of the prime implicants

Iterated Consensus for One Output (4/12)

- Consider finding the prime implicants of the function

$$f(w,x,y,z)=\Sigma m(0,4,5,7,8,11,12,15)$$

- We choose as a starting point a set of product terms that cover the function; they include some prime implicants and a minterm, as well as other implicants

A	$w'x'y'z'$
B	$w'xy'$
C	$wy'z'$
D	xyz
E	wyz

- We labeled the terms for reference and go in the order

B ¢ A, C ¢ B, C ¢ A, D ¢ C, . . . ,

Iterated Consensus for One Output (5/12)

- Omitting any computation when the term has been removed from the list. When a term is removed, we cross it out. The first consensus, $B ¢ A$, produces $w'y'z'$; A is included in that term and can thus be removed. After the first step, the list becomes

~~A~~	~~$w'x'y'z'$~~
B	$w'xy'$
C	$wy'z'$
D	wyz
E	wyz
F	$w'y'z'$

Iterated Consensus for One Output (6/12)

- The process is repeated and we obtain Table 4.2

Table 4.2 Computing the prime implicants.

~~A~~	~~$w'x'y'z'$~~		
B	$w'xy'$		
~~C~~	~~$wy'z'$~~		
D	xyz		
E	wyz		
~~F~~	~~$w'y'z'$~~	B ¢ $A \geq A$ (remove A)	
~~G~~	~~$xy'z'$~~	C ¢ B	
		D ¢ C	undefined
H	$w'xz$	D ¢ B	
		E ¢ D	undefined
		E ¢ C	undefined
		E ¢ B	undefined
		F ¢ E	undefined
		F ¢ D	undefined
J	$y'z'$	F ¢ $C \geq G, F, C$ (remove G, F, C)	
		H ¢ $E = D$ (do not add)	
		H ¢ D	undefined
		H ¢ B	undefined
		J ¢ $H = B$ (do not add)	
		J ¢ E	undefined
		J ¢ D	undefined
		J ¢ B	undefined

Iterated Consensus for One Output (7/12)

- The terms that remain, *B*, *D*, *E*, *H*, and *J*, that is *w'xy*, *xyz*, *wyz*, *w'xz*, and *y'z'*, are all the prime implicants. The minimum sum of products expression(s) will use some of these, typically not all of them
- The process can be simplified by using a numeric representation of the terms. If a variable is missing from a term, a dash (-) is used in its place so that each term has four entries. The process is shown in Table 4.3

Iterated Consensus for One Output (8/12)

Table 4.3 Numeric computation of prime implicants.

~~*A*~~	~~0~~	~~0~~	~~0~~	~~0~~	
B	0	1	0	–	
~~*C*~~	~~1~~	~~–~~	~~0~~	~~0~~	
D	–	1	1	1	
E	1	–	1	1	
~~*F*~~	~~0~~	~~–~~	~~0~~	~~0~~	*B* ¢ *A* ≥ *A*
~~*G*~~	~~–~~	~~1~~	~~0~~	~~0~~	*C* ¢ *B*
H	0	1	–	1	*D* ¢ *B* (*D* ¢ *C* undefined)
					(*E* ¢ *D*, *E* ¢ *C*, *E* ¢ *B*, *F* ¢ *E*, *F* ¢ *D* undefined)
J	–	–	0	0	*F* ¢ *C* ≥ *G*, *F*, *C*
					(*H* ¢ *E* = *D*; *H* ¢ *D*, *H* ¢ *B* undefined; *J* ¢ *H* = *B*; *J* ¢ *E*, *J* ¢ *D*, *J* ¢ *B* undefined)

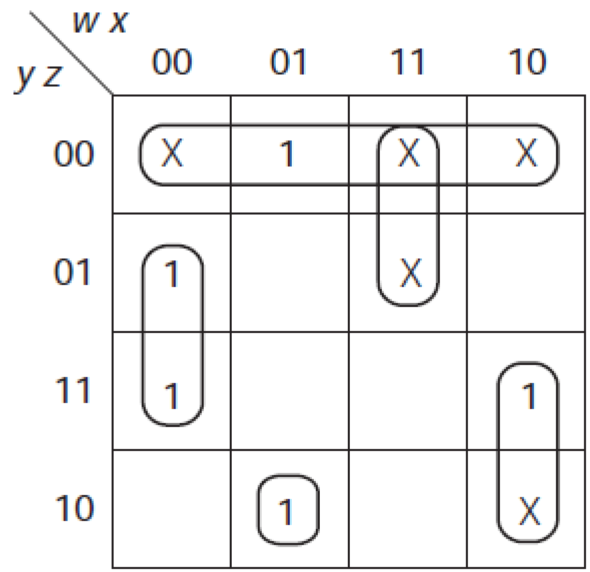
Iterated Consensus for One Output (9/12)

- If there are don't cares in the function, all of them must be included in at least one of the terms to start the process. The resulting list of prime implicants will then include all possible prime implicants (including possibly some that are made up of only don't cares)

Iterated Consensus for One Output (10/12)

- **Example 4.2.** Find all the prime implicants of the function

$$f(w,x,y,z)=\Sigma m(1,3,4,6,11)+\Sigma d(0,8,10,12,13)$$



- Using the map above, we chose the following list of implicants as a starting point:

Iterated Consensus for One Output (11/12)

- **Example 4.2. (Cont'd)**

A	$y'z'$	–	–	0	0
B	$w'x'z$	0	0	–	1
C	$w'xyz'$	0	1	1	0
D	wxy'	1	1	0	–
E	$wx'y$	1	0	1	–

- All of these, except the third, are prime implicants. It does not matter what set of terms we start with (as long as all of the 1's and don't cares are included in at least one term); we will get the same result

Iterated Consensus for One Output (12/12)

- **Example 4.2. (Cont'd)** The process proceeds:

A	–	–	0	0	
B	0	0	–	1	
~~*C*~~	~~0~~	~~1~~	~~1~~	~~0~~	
D	1	1	0	–	
E	1	0	1	–	
F	0	0	0	–	*B* ¢ *A* *C* ¢ *B* undefined
G	0	1	–	0	*C* ¢ *A* ≥ *C* *D* ¢ *B*, *D* ¢ *A*, *E* ¢ *D* undefined
H	–	0	1	1	*E* ¢ *B*
J	1	0	–	0	*E* ¢ *A*

F ¢ *E*, *F* ¢ *D*, *F* ¢ *B*, *F* ¢ *A* undefined, *G* ¢ *F* = 0 – 0 0 ≤ *A*;
G ¢ *E* undefined; *G* ¢ *D* ≤ *A*; *G* ¢ *B*, *G* ¢ *A*, *H* ¢ *G* undefined;
H ¢ *F* = *B*; *H* ¢ *E*, *H* ¢ *D*, *H* ¢ *B*, *H* ¢ *A*, undefined; *J* ¢ *H* = *E*;
J ¢ *G*, *J* ¢ *E*, *J* ¢ *B*, *J* ¢ *A* undefined; *J* ¢ *F* ≤ *A*, *J* ¢ *D* ≤ *A*

- All terms but term *C* are prime implicants

Prime Implicant Tables for One Output (1/4)

- Once we have a complete list of prime implicants, using either Quine-McCluskey or iterated consensus, a table is constructed with one row for each prime implicant and one column for each minterm included in the function (not don't care). An X is entered in the column of a minterm that is covered by prime implicant

Table 4.4 A prime implicant (PI) table.

PI	Numeric	$	Label	0	4	5	7	8	11	12	15
$w'xy'$	0 1 0 –	4	*A*		X	X					
xyz	– 1 1 1	4	*B*				X				X
wyz	1 – 1 1	4	*C*						X		X
$w'xz$	0 1 – 1	4	*D*			X	X				
$y'z'$	– – 0 0	3	*E*	X	X			X		X	

Prime Implicant Tables for One Output (2/4)

- The third column of Table 4.4 is the number of gate inputs when that term is used in a two-level circuit, that is, just one for each literal plus one for the input to the output gate (OR)
- If there is more than one set of rows to cover all the minterms, the total number of gate inputs ($ column) is minimized
- The first step in the process is to find essential prime implicants. The minterms are shaded if covered by the essential prime implicants and an asterisk is placed next to the prime implicant in Table 4.5

Prime Implicant Tables for One Output (3/4)

- Note that all of the minterms covered by the essential prime implicants are checked, not just those columns with shaded X's

Table 4.5 Finding essential prime implicants.

				√	√			√	√	√	√
PI	**Numeric**	**$**	**Label**	**0**	**4**	**5**	**7**	**8**	**11**	**12**	**15**
$w'xy'$	0 1 0 –	4	*A*		X	X					
xyz	– 1 1 1	4	*B*				X				X
$wyz^\star$	1 – 1 1	4	*C*						X		X
$w'xz$	0 1 – 1	4	*D*			X	X				
$y'z'^\star$	– – 0 0	3	*E*	X	X			X		X	

Prime Implicant Tables for One Output (4/4)

- The table is now reduced to that of Table 4.6 by eliminating the essential prime implicant rows and the covered minterms. In this simple example, the solution is

$$C + E + D = wyz + y'z' + w'xz$$

Table 4.6 The reduced table.

$	Label	5	7
4	*A*	X	
4	*B*		X
4	*D*	X	X