Technische Informatik: Abgabe 5

Michael Mardaus　　Andrey Tyukin

28. November 2013

Exercise 5.1 (Quine McCluskey and K-maps)

a) We apply the Quine McCluskey algorithm to obtain optimal disjunctive representation of f.

#(0)	decimal	$x_1x_2x_3x_4x_5$	Reduction-Links						
1	15	01111							
	30	11110	$\star^{1}$	$\star^{2}$					
	29	11101			$\star^{3}$	$\star^{4}$	$\star^{5}$		
2	28	11100	$\star^{1}$		$\star^{3}$			$\star^{6}$	
	25	11001			$\star^{8}$	$\star^{4}$			$\star^{7}$
	22	10110		$\star^{2}$					
	21	10101	$\star^{9}$				$\star^{5}$		
3	24	11000		$\star^{10}$				$\star^{6}$	$\star^{7}$
	17	10001	$\star^{9}$		$\star^{8}$				
	9	01001			$\star^{11}$				
4	8	01000	$\star^{12}$	$\star^{10}$	$\star^{11}$				
	2	00010		$\star^{13}$					
5	0	00000	$\star^{12}$	$\star^{13}$					

$\Downarrow$　　$\Downarrow$

#(0)	decimal	$x_1x_2x_3x_4x_5$	Reduction-Links			
1	15	01111				
	28,30	111*0				
	22,30	1*110				
	28,29	1110*	$\star^{1}$			
	25,29	11*01		$\star^{2}$	$\star^{3}$	
	21,29	1*101				$\star^{4}$
2	24,28	11*00		$\star^{2}$		
	24,25	1100*	$\star^{1}$			
	17,25	1*001				$\star^{4}$
	17,21	10*01			$\star^{3}$	
3	8,24	*1000				
	8,9	0100*				
4	0,8	0*000				
	0,2	000*0				

$\Downarrow$　　$\Downarrow$

#(0)	decimal	$x_1x_2x_3x_4x_5$
1	15	01111
	28,30	111*0
	22,30	1*110
	24,25,28,29	11*0*
	17,21,25,29	1**01
3	8,24	*1000
	8,9	0100*
4	0,2	000*0

$\Downarrow$ $\Downarrow$

Primimplicant	0	2	8	9	15	17	21	22	24	25	28	29	30	
01111					1									
111*0											1		1	reduced r 4
1*110								1					1	
11*0*									1	1	1	1		
1**01						1	1			1		1		
*1000			1						1					reduced r 4
0100*			1	1										
000*0	1	1												
reduced b/c			c9				c17			c17		c17	c22	

...here lives the big table for row and column cheating...

$\Longrightarrow f = \neg x_1x_2\neg x_3\neg x_4 + \neg x_1\neg x_2\neg x_3\neg x_5 + \neg x_1x_2x_3x_4x_5 + x_1x_2\neg x_4 + x_1\neg x_4x_5 + x_1x_3x_4\neg x_5$

b) Use K-Maps to find a cost-minimal DNF.

$x_1 = 0$				
	x_4x_5			
x_2x_3	00	01	11	10
00	1			1
01				
11			1	
10	1	1		

$x_1 = 1$				
	x_4x_5			
x_2x_3	00	01	11	10
00		1		
01		1		1
11	1	1		1
10	1	1		

which yields: $f = \neg x_1x_2\neg x_3\neg x_4 + \neg x_1\neg x_2\neg x_3\neg x_5 + \neg x_1x_2x_3x_4x_5 + x_1x_2\neg x_4 + x_1\neg x_4x_5 + x_1x_3x_4\neg x_5$
The same result as in part a).

Exercise 5.2 (Row and Column-Rules are not a function)

Primimplic.	$C_1 = 0$	$C_2 = 1$	$C_3 = 2$
$R_1 = 01$		1	
$R_2 = 0*$	1	1	
$R_3 = *0$	1		1

We deleted Column C_1 with the Column-Rule, because $C_3 \leq C_1$ and overlaps it. Now we can either use $R_3 + R_2 = \neg x_2 + \neg x_1$ or $R_3 + R_1 = \neg x_2 + \neg x_1 x_2$ as our minimal solution.

Exercise 5.3 (Greedy Algorithm)

We consider boolean matrices with at least one 1 in each column and at most two 1's in each row. The greedy algorithm repeatedly traverses all rows and in each step chooses the row that covers the most columns, until all colums are covered. We assume top-down traversal order, furthermore we assume that if two rows cover the same number of columns, that the one with smaller index is chosen.

a) Claim: The greedy algorithm generally does not find the optimal solution.

Proof: Consider the following example:

1	1	0	0
1	0	1	0
0	1	0	1

In the first iteration each row covers two columns, therefore the algorithm would just pick the first row. In subsequent steps, it also has to take second and third row into the cover, because both are necessary (there is only one 1 in third and fourth column). However, the optimal solution does not require all rows: the second and third row alone would be sufficient.

b) Let n be the number of rows and m be the number of columns of the matrix (the restrictions on the numbers of 1's in rows and column shall still hold).

Claim: There exists a cover with $m - k \leq m$ rows iff there exists $K \subset \{1, \ldots, n\}$ of cardinality k such that rows $\{z_j\}_{j \in K}$ cover $2k$ columns.

Proof: First, let there be such a subset K. By assumption, $\{z_j\}_{j \in K}$ already cover $2k$ columns, and $m - 2k$ columns remain uncovered. Because there is at least one 1 in each of the remaining $m - 2k$ columns, we need at most $l \leq m - 2k$ more rows to cover these columns. Together, we need $k + l \leq k + (m - 2k) = m - k$ rows to cover all columns.

For the opposite direction, let $C \subset \{1, \ldots, n\}$ with $|C| = m - k$ be a subset of row-indices which cover all m columns. Let $\hat{K} \subset C$ be a maximal set with the property that $\{z_j\}_{j \in \hat{K}}$ cover $2|\hat{K}|$ rows. Because of the maximality and the restriction that there are at most two 1's per row, adding further rows with indices from $C - \hat{K}$ to $\hat{K}$ increases the number of covered columns at most by 1 per added row, that is, we can cover at most $2|\hat{K}| + (|C| - |\hat{K}|) = |\hat{K}| + |C| = |\hat{K}| + m - k$ rows. If $|\hat{K}| < k$ would hold, then we could cover at most $m - 1$ columns, which contradicts the assumption that rows with indices from C cover *all* columns. Therefore $|\hat{K}| \geq k$ must hold. In particular, there exists a subset $K \subset \hat{K}$ of size exactly k such that $\{z_j\}_{j \in K}$ cover exactly $2k$ columns.