VE527: Assignment #6

Name: Chang Meng
Student ID: 118370910019

December 7, 2018

1 (6%) Prime Cover

Given the following SOP Boolean expressions:

$$wxy + yz + w\bar{y}\bar{z} + \bar{w}x\bar{y}z$$

Is it a prime cover, i.e., are all the cubes in that expression prime implicants? Why or why not?

Answer: No, because $\bar{w}x\bar{y}z$ can be covered by a "larger" implicant $\bar{w}xz$.

2 (10%) Expand Operation in Espresso

Suppose that we are optimizing an SOP Boolean expression $F(a, b, c, d)$ using the reduce-expand-irredundant loop. Assume that we have just done a REDUCE step and we have an intermediate, non-prime 4-cube cover of F as

$$F = \bar{a}b\bar{c}\bar{d} + b\bar{c}d + cd + bc\bar{d}$$

We want to perform an EXPAND operation on the cube $\bar{a}b\bar{c}\bar{d}$. As we learned in lecture, ESPRESSO does this by building a cover of the OFF-set for this current cover, then building a blocking matrix for the cube we seek to expand, and then computing a cover of this matrix, which tells us how to grow this cube.

Apply the recipe and show the result of EXPAND. For simplicity, assume that the OFF-set of F is given to you, which is the following 3-cube cover

$$\bar{F} = \bar{b}\bar{d} + \bar{b}\bar{c} + a\bar{c}\bar{d}$$

(Hint: you can draw Karnaugh map to verify if your answer is correct.)

Answer:
The blocking matrix is:

cube / var	$\bar{b}\bar{d}$	$\bar{b}\bar{c}$	$a\bar{c}\bar{d}$
$\bar{a}$	0	0	1
b	1	1	0
$\bar{c}$	0	0	0
$\bar{d}$	0	0	0

We can only cover the matrix with $\bar{a}$ and b, the corresponding legal cube expansion is $\bar{a}b$. Thre result after EXPAND this cube $\bar{a}b\bar{c}\bar{d}$ is

$$F = \bar{a}b + b\bar{c}d + cd + bc\bar{d}$$

3 (10%) Algebraic Division

Condider the following two functions of variables p, q, r, s, t, u, v, w, x, y, z:

$$F = pt + xz + qt + pyz + rst + qyz + puvw + px + quvw + rsuvw$$

$$D = p + q + rs$$

Use the algebric division algorithm from class to compute $Q = F/D$ and the remainder R.
(Hint: you want to build the table as in the lecture slides: one row for each cube in F; one column for each cube in D; do the cube-wise walk though D and build up the partial quotient solution for Q one column of this table at a time. When you are done, you can obtain the remainder R from the computed quotient.)

Answer:

F cube	D cube p	D cube q	D cube rs
pt	t	–	–
xz	–	–	–
qt	–	t	–
pyz	yz	–	–
rst	–	–	t
qyz	–	yz	–
$puvw$	uvw	–	–
px	x	–	–
$quvw$	–	uvw	–
$rsuvw$	–	–	uvw

$$Q = (t + yz + uvw + x) \cap (t + yz + uvw) \cap (t + uvw) = t + uvw$$
$$R = F - QD = F - (pt + puvw + qt + quvw + rst + rsuvw) = xz + pyz + qyz + px$$

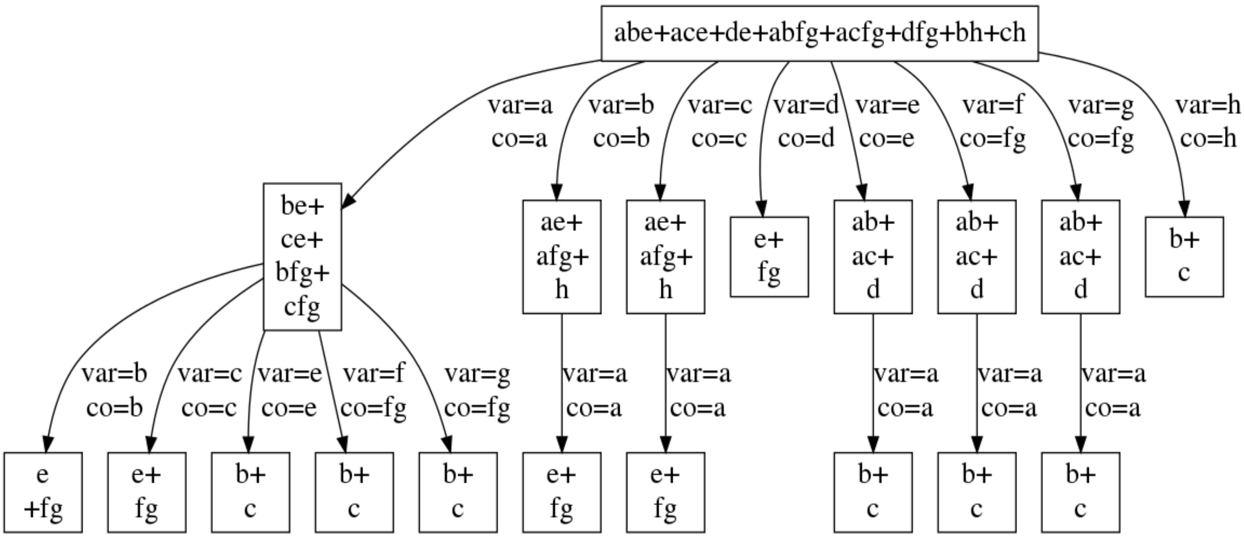
4 (20%) Kerneling

Here is a function represented in algebraic form:

$$F = abe + ace + de + abfg + acfg + dfg + bh + ch$$

Assume that the variable order is a, b, c, d, e, f, g, h. Use the recursive kerneling algorithm discussed in the lecture, and run the algorithm by han =d to extract all the kernels and their associated co-kernels from F. Show all the co-kernel — kernel pairs. Also, show the level number of each kernel (i.e., some kernel is a level-0 kernel, some level is a level-1 kernel, etc.)

Answer:



All the kernel — co-kernel pairs are shown in the table:

kernel	co-kernel	level of kernel
$abe + ace + de + abfg + acfg + dfg + bh + ch$	1	2
$be + ce + bfg + cfg$	a	1
$ae + afg + h$	b or c	1
$ab + ac + d$	e or fg	1
$b + c$	h or ae or afg	0
$e + fg$	d or ab or ac	0

5 (18%) Single Cube Divisor Extraction

Consider the following Boolean logic network with variables a, b, c, d, e, f:

$$R = abf + abcd + abce + cdf$$

$$S = acd + cdef + abce$$

$$T = cdef + af$$

Build the cube-literal matrix associated with this set of functions. List all the single cube divisors that can be extracted based on a non-trival prime rectangle that covers the column c. (Here, a prime rectangle is non-trival if it covers at least two rows and at least two column.) Which of them are the best in terms of the number of literals saved? You should apply the simple formula talked in lecture to compute the number of literals saved.

Answer:
The cube-literal matrix is shown in the table:

	a	b	c	d	e	f
abf	1	1	-	-	-	1
$abcd$	1	1	1	1	-	-
$abce$	1	1	1	-	1	-
cdf	-	-	1	1	-	1
acd	1	-	1	1	-	-
$cdef$	-	-	1	1	1	1
af	1	-	-	-	-	1

All the single cube divisors are listed below:

divisors	C	$\sum Weight(r)$	# saved literals
ac	2	$1+2+1=4$	$(2-1)\times 4-2=2$
bc	2	$1+2=3$	$(2-1)\times 3-2=1$
cd	2	$1+1+1+2=5$	$(2-1)\times 5-2=3$
ce	2	$2+2=4$	$(2-1)\times 4-2=2$
cf	2	$1+2=3$	$(2-1)\times 3-2=1$
abc	3	$1+2=3$	$(3-1)\times 3-3=3$
acd	3	$1+1=2$	$(3-1)\times 2-3=1$
cdf	3	$1+2=3$	$(3-1)\times 3-3=3$

6 (18%) Multiple Cube Divisor Extraction

Suppose that we have the following two Boolean functions, defined over 9 variables p, q, r, s, t, u, v, w, x:

$$G = qt + rst + pqr + quvw + rsuvw$$

$$H = pu + qtx + qu + rsu$$

Apply the method we talked in the lecture to build a co-kernel-cube matrix. List all multiple cube divisors that can be extracted based on a non-trivial prime rectangle. (Here, a prime rectangle is non-trivial if it covers at least two rows and at least two columns.) For each multiple cube divisor you have extracted, draw the the Boolean network. What is the number of literals saved with each extracted multiple cube divisor? You should apply the simple formula talked in lecture to compute the number of literals saved.
To assist you in the construction, here are all the kernels and co-kernels for the two functions:
Function $G = qt + rst + pqr + quvw + rsuvw$

Kernel	Co-kernel
$t+pr+uvw$	q
$t+uvw$	rs
$st+pq+suvw$	r
$q+rs$	t
$q+rs$	uvw

Function $H = pu + qtx + qu + rsu$

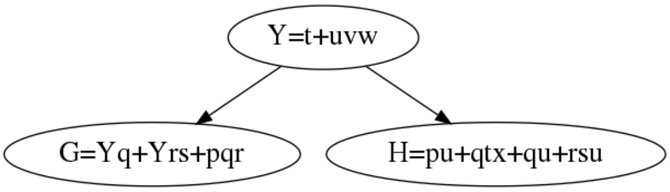
Kernel	Co-kernel
$u+tx$	q
$p+q+rs$	u

Answer:
The co-kernel-cube matrix is:

	t	p	q	u	pr	st	pq	rs	tx	uvw	$suvw$
$G\ q$	1	-	-	-	1	-	-	-	-	1	-
$G\ rs$	1	-	-	-	-	-	-	-	-	1	-
$G\ r$	-	-	-	-	-	1	1	-	-	-	1
$G\ t$	-	-	1	-	-	-	-	1	-	-	-
$G\ uvw$	-	-	1	-	-	-	-	1	-	-	-
$H\ q$	-	-	-	1	-	-	-	-	1	-	-
$H\ u$	-	1	1	-	-	-	-	1	-	-	-

All multiple cube divisors are $t+uvw$ and $q+rs$.
For $t+uvw$:

$$G = (t+uvw)(q+rs) + pqr$$
$$H = pu + qtx + qu + rsu$$



Column weight,

$$Weight(t) = 1$$
$$Weight(uvw) = 3$$

Row weight,

$$Weight(Gq) = 1+1 = 2$$
$$Weight(Grs) = 1+2 = 3$$

$\sum Value(r,c)$,

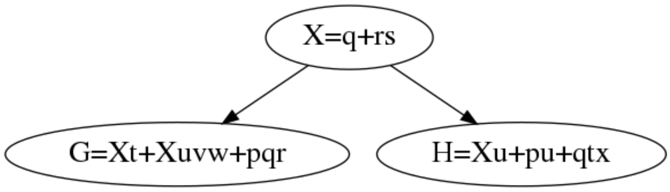
$$2+4+3+5 = 14$$

literals saved:

$$14 - (3+2) - (1+3) = 5$$

For $q+rs$:

$$G = (q+rs)(t+uvw) + pqr$$
$$H = (q+rs)(u) + pu + qtx$$



Column weight,

$$Weight(q) = 1$$
$$Weight(rs) = 2$$

Row weight,

$$Weight(Gt) = 1 + 1 = 2$$
$$Weight(Guvw) = 1 + 3 = 4$$
$$Weight(Hu) = 1 + 1 = 2$$

$\sum Value(r, c)$,

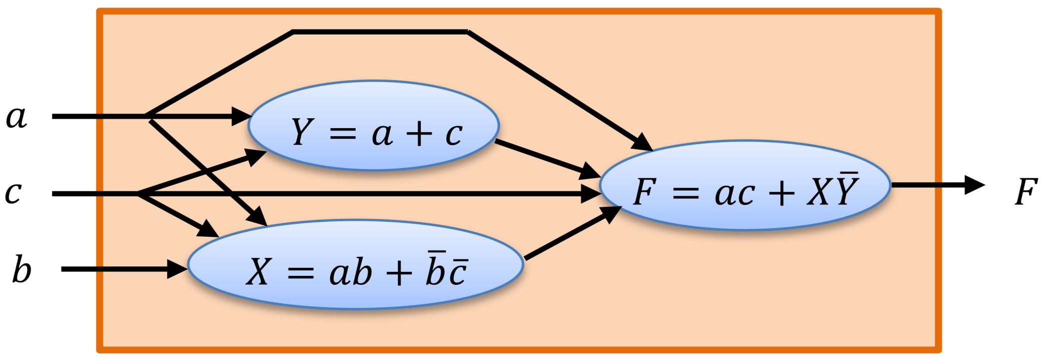
$$2 + 3 + 4 + 5 + 2 + 3 = 19$$

literals saved:

$$19 - (1 + 2) - (2 + 4 + 2) = 8$$

7 (10%) Controllability Don't Cares (CDCs) in Multi-Level Logic

Consider the following small Boolean logic network:



Use the methods from the lecture to obtain CDC_F for node F.
Answer:
The inputs of F are X, Y, a and c. Their SDCs are:

$$SDC_X = X \oplus (ab + \bar{b}\bar{c})$$

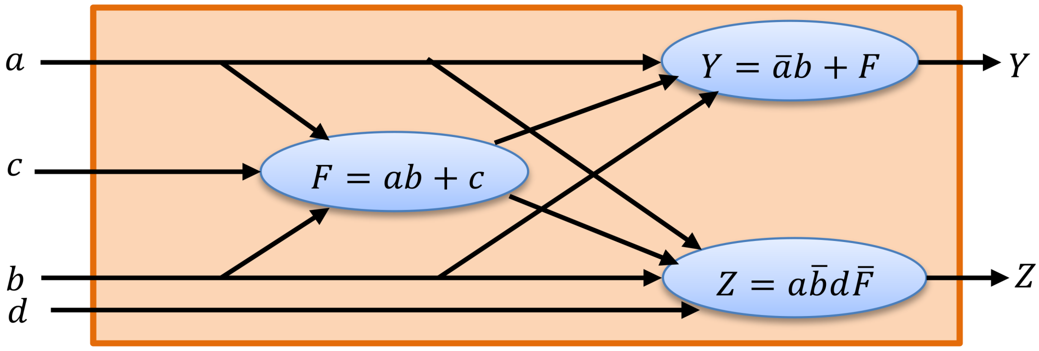
$$SDC_Y = Y \oplus (a + c)$$
$$SDC_a = 0$$
$$SDC_c = 0$$

b is not used in F.

$$
\begin{aligned}
&CDC_F = \\
&(\forall b)[SDC_X + SDC_Y] = \\
&(\forall b)[X \oplus (ab + \bar{b}\bar{c}) + Y \oplus (a + c)] = \\
&[X \oplus (ab + \bar{b}\bar{c}) + Y \oplus (a + c)]_{b=1} \cdot [X \oplus (ab + \bar{b}\bar{c}) + Y \oplus (a + c)]_{b=0} = \\
&[(X \oplus a) + (Y \oplus (a + c))] \cdot [(X \oplus \bar{c}) + (Y \oplus (a + c))] = \\
&(Y \oplus (a + c)) + (X \oplus a)(X \oplus \bar{c}) = \\
&Y\bar{a}\bar{c} + \bar{Y}a + \bar{Y}c + X\bar{a}c + \bar{X}a\bar{c}
\end{aligned}
$$

8 (8%) Observability Don't Cares (ODCs) in Multi-Level Logic

Condider the following small Boolean logic network:



Use the methods from the lecture to obtain ODC_F for node F.
Answer:
The outputs of F are Y and Z. The negation of Boolean difference are:

$$\overline{\partial Y/\partial F} = (\bar{a}b)\overline{\oplus}1 = \bar{a}b$$

$$\overline{\partial Z/\partial F} = (a\bar{b}d)\overline{\oplus}0 = \bar{a} + b + \bar{d}$$

$$\overline{\partial Y/\partial F} \cdot \overline{\partial Z/\partial F} = \bar{a}b + \bar{a}b\bar{d}$$

d is not used in F.

$$
\begin{aligned}
&ODC_F = \\
&(\forall d)[\overline{\partial Y/\partial F} \cdot \overline{\partial Z/\partial F}] = \\
&(\forall d)[\bar{a}b + \bar{a}b\bar{d}] = \\
&[\bar{a}b] \cdot [\bar{a}b] = \\
&\bar{a}b
\end{aligned}
$$