Logic Synthesis & Verification, Fall 2024

National Taiwan University

Reference Solution for Problem Set 4

1 Weak Division

First note that given $F, G, H, R = F \setminus GH$ must be uniquely determined. Thus, it suffices to show that H is unique. We show that the set of all valid H is exactly the powerset of some set H by showing that

1. if H_1 is valid and $H \subseteq H1$, then H is also valid, and
2. if H_1 and H_2 are valid, then $H = H1 \cup H2$ is also valid.

The condition for disjoint support holds trivially, thus it suffices to check whether $GH \subseteq F$. Next observe that for some H with $G \perp H$, we have $GH = \{gh | g \in G, h \in H\}$. For statement 1, since $GH \subseteq GH_1 \subseteq F$, the statement holds. For statement 2, since $GH = G(H1 \cup H2) = GH1 \cup GH2 \subseteq F$, the statement holds as well.

Finally, since $R = F \setminus GH$, the H that minimizes R must be the unique largest H (the universe), which completes the proof.

2 Kernelling and Factoring

(a) The kernelling tree of `KERNEL(0,F)` are shown in fig. 1.

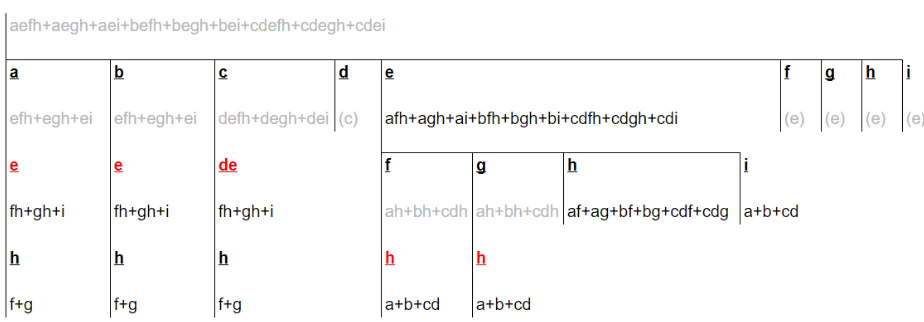

Fig. 1: The kerneling tree that shows the process of `KERNEL(0,F)`. The underlined cubes are used to divide the Boolean functions. The quotients in gray text are not kernels as they are not cube free. The cube in red text are the common literals removed by `MAKE_CUBE_FREE()`.

The kernels and their corresponding co-kernels are shown in fig. 2. Note that the kernels are not necessarilly in factored form.

Co-kernel	Kernel
aeh, beh, cdeh	f+g
efh, egh, ei	a+b+cd
ae, be, cde	h(f+g)+i
eh	(a+b+cd)(f+g)
e	(a+b+cd)(h(f+g)+i)

Fig. 2: Kernels and co-kernels.

(b) As shown in fig. 3, we first generate 2-cube divisors by

$$D = \{d | d = \texttt{make\_cube\_free}(c_i + c_j)\}$$

where c_i and c_j are any pair of cubes.

	aegh	aei	aefh	befh	begh	bei	cdefh	cdegh	cdei
aegh		gh+i	f+g	ag+bf	a+b	agh+bi	ag+cdf	a+cd	agh+cdi
aei			i+fh	ai+bfh	ai+bgh	a+b	ai+cdfh	ai+cdgh	a+cd
aefh				a+b	af+bg	afh+bi	a+cd	af+cdg	afh+cdi
befh					f+g	fh+i	b+cd	bf+cdg	bfh+cdi
begh						gh+i	bg+cdf	b+cd	bgh+cdi
bei							bi+cdfh	bi+cdgh	b+cd
cdefh								f+g	fh+i
cdegh									gh+i
cdei									

Fig. 3: 2-cube divisors. Gray ones are duplicated divisors.

Then, we compute the complement of these 2-cube divisors and keep those with exactly 2 cubes. The results are shown in fig. 4.
2-cube divisors are

$ae, af, ag, ah, ai, be, bf, bg, bh, bi, cd, ce, cf, cg, ch, ci, de, df, dg, dh, di, ef, eg, eh, ei, fh, gh$

The complmenet of 2-cube divosrs are

$$\begin{aligned}
&a' + e', a' + f', a' + g', a' + h', a' + i',\\
&b' + e', b' + f', b' + g', b' + h', b' + i',\\
&c' + d', c' + e', c' + f', c' + g', c' + h', c' + i',\\
&d' + e', d' + f', d' + g', d' + h', d' + i',\\
&e' + f', e' + g', e' + h', e' + i',\\
&f' + h', g' + h'
\end{aligned}$$

Only $f + g$ is a kernel.

gh+i	g'i'+h'i'
f+g	-
ag+bf	-
a+b	-
agh+bi	-
ag+cdf	-
a+cd	a'c'+a'd'
agh+cdi	-
i+fh	f'i'+h'i'
ai+bfh	-
ai+bgh	-
ai+cdfh	-
ai+cdgh	-
af+bg	-
afh+bi	-
af+cdg	-
afh+cdi	-
b+cd	b'c+b'd'
bf+cdg	-
bfh+cdi	-
bg+cdf	-
bgh+cdi	-
bi+cdfh	-
bi+cdgh	-

Fig. 4: 2-cube divisors and their (2-cube) complements.

(c)
```
GFACTOR(F,DIVISOR,weak)
    D=a+b+cd
    Q=ehf+ehg+ei
    |Q|!=1
    Q=hf+hg+i
    D=ae+be+cde
    R=0
    D is not cube free
    C=e
    return LF(F, e, a+b+cd, weak)=e(h(f+g)+i)(a+b+cd)
        L=e
        Q=afh+agh+ai+bfh+bgh+bi+cdfh+cdgh+cdi
        R=0
        C=0
        Q=afh+agh+ai+bfh+bgh+bi+cdfh+cdgh+cdi
        Q=GFACTOR(Q,DIVISOR,weak)=(h(f+g)+i)(a+b+cd)
            D=a+b+cd
            Q=hf+hg+i
            Q=Q
            D=a+b+cd
            R=0
            D is cube free
            Q=GFACTOR(hf+hg+i,DIVISOR,weak)=h(f+g)+i
                D=f+g
```

```
                    Q=h
                    |Q|=1
                    return LF(hf+hg+i,h,DIVISOR,weak)=h(f+g)+i
                        L=h
                        Q=f+g
                        R=i
                        C=0
                        Q=Q
                        Q=GFACTOR(f+g,DIVISOR,weak)=f+g
                            D=0
                            return f+g
                        R=GFACTOR(i,DIVISOR,weak)=i
                            D=0
                            return i
                        return h(f+g)+i
                D=GFACTOR(a+b+cd,DIVISOR,weak)=a+b+cd
                    D=0
                    return a+b+cd
                R=GFACTOR(0,DIVISOR,weak)=0
                    D=0
                    return 0
                return (h(f+g)+i)(a+b+cd)
            R=GFACTOR(0,DIVISOR,weak)=0
                D=0
                return 0
            return e(h(f+g)+i)(a+b+cd)
```

(d) GFACTOR(F,DIVISOR,weak)

```
    D=a+cd
    D!=0
    Q=efh+egh+ei
    |Q|!=1
    Q=fh+gh+i
    D=ae+be+cde
    R=0
    D is not cube free
    ... (same as in (c))
```

3 Extraction and Rectangle Covering

The level-0 kernels and corresponding co-kernels of F and G are listed below.

The co-kernel cube matrix is shown below.

We pick columns e, f, g and all corresponding rows, and name the new node $h = (e + f + g)$. After updating, we have $F = abh + adh + ch$ and $G = abc + abh + dh + cd$. $V_1 = 39 - 13 - 3 = 23$.

The updated co-kernel cube matrix is shown below.

	Kernel	Co-Kernels
F	$e+f+g$	ab, ad, c
	$b+d$	ae, af, ag
G	$c+e+f+g$	ab, d
	$ab+d$	c, e, f, g

		ab	b	c	d	e	f	g
F	ab					abe	abf	abg
F	ad					ade	adf	adg
F	c					ce	cf	cg
F	ae		abe		ade			
F	af		abf		adf			
F	ag		abg		adg			
G	ab			abc		abe	abf	abg
G	d			cd		de	df	dg
G	c	abc			cd			
G	e	abe			de			
G	f	abf			df			
G	g	abg			dg			

		ab	b	c	d	h
F	ah		abh		adh	
G	ab			abc		abh
G	d			cd		ch
G	c	abc			cd	
G	h	abh			dh	

Since combining the two is no longer beneficial, we choose $i = b + d$ and $j = c + h$, and get $F = ahi + ch$ and $G = abj + dj$. Since $V_2 = 6 - 3 - 2 = 1 > 0$ and $V_3 = 10 - 5 - 2 = 3 > 0$, both extractions are beneficial.

Since no more improvement can be done, we conclude that $F = ahi + ch$, $G = abj + dj$, $h = e + f + g$, $i = b + d$, and $j = c + h$.

4 Pre-Image Computation

The Boolean relation of output and input variables is

$$R(\mathbf{x}, \mathbf{y}) = (y_1 \equiv ab \oplus bc \oplus ac)(y_2 \equiv bc + bd + cd)(y_3 \equiv \neg a + \neg d)$$

The characteristic function of the subset of output vector is

$$A(\mathbf{y}) = (\neg y_1 y_2 + \neg y_3)$$

The pre-image of $A(\mathbf{y})$ is

$$\begin{aligned}
&\exists\, \mathbf{y}.A(\mathbf{y})R(\mathbf{x}, \mathbf{y}) \\
&= \exists\, y_1 y_2 y_3.(\neg y_1 y_2 + \neg y_3)(y_1 \equiv ab \oplus bc \oplus ac)(y_2 \equiv bc + bd + cd)(y_3 \equiv \neg a + \neg d) \\
&= \exists\, y_1 y_2.(y_1 \equiv ab \oplus bc \oplus ac)(y_2 \equiv bc + bd + cd)(ad + (\neg y_1 y_2)(\neg a + \neg d) \\
&= \exists\, y_1 y_2.(y_1 \equiv ab \oplus bc \oplus ac)(y_2 \equiv bc + bd + cd)(ad + \neg a \neg y_1 y_2 + \neg d \neg y_1 y_2) \\
&= \exists\, y_1.(y_1 \equiv ab \oplus bc \oplus ac)((\neg(bc + bd + cd)ad) + (bc + bd + cd)(ad + \neg a \neg y_1 + \neg d \neg y_1)) \\
&= \exists\, y_1.(y_1 \equiv ab \oplus bc \oplus ac)((\neg b + \neg c)(\neg b + \neg d)(\neg c + \neg d)ad + (ad + \neg y_1(\neg a + \neg d))(bc + bd + cd)) \\
&= \exists\, y_1.(y_1 \equiv ab \oplus bc \oplus ac)(a\neg b \neg cd + ad(b + c) + \neg y_1(\neg a + \neg d)(bc + bd + cd)) \\
&= \exists\, y_1.(y_1 \equiv ab \oplus bc \oplus ac)(ad(\neg b \neg c + b + c) + \neg y_1(\neg a(bc + bd + cd) + bc\neg d)) \\
&= \neg(ab \oplus bc \oplus ac)(ad(\neg b \neg c + b + c)a + (\neg a(bc + bd + ed) + bc\neg d)) + (ab \oplus bc \oplus ac)ad(\neg b \neg c + b + c) \\
&= ad(\neg b \neg c + b + c) + \neg(ab \oplus bc \oplus ac)(\neg a(bc + bd + cd) + bc\neg d) \\
&= ad(\neg b \neg c + b + c) + \neg(ab \oplus bc \oplus ac)\neg a(bc + bd + cd) + \neg(ab \oplus bc \oplus ac)bc\neg d \\
&= a\neg b \neg cd + abd + acd + \neg(bc)\neg a(bc + bd + cd) \\
&= a\neg b \neg cd + abd + acd + \neg a \neg bcd + \neg ab \neg cd
\end{aligned}$$

The elements in the pre-image are

$$\{a\neg b \neg cd, abcd, ab\neg cd, a\neg bcd, \neg a \neg bcd, \neg ab\neg cd\}$$

5 Functional Dependency

We first calculate the image of the on-set of f_1 on $\{y_2, y_3\}$:

$$
\begin{aligned}
h_{\text{on}}(y_2, y_3) &= \exists\, a, b, c, d, e.(a \oplus d \oplus e)(y_2 \equiv a \oplus b \oplus c)(y_3 \equiv b \oplus c \oplus d \oplus e) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)((a \oplus d)(y_3 \oplus b \oplus c \oplus d)' \vee (a \oplus d)'(y_3 \oplus b \oplus c \oplus d)) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)((a \oplus d) \oplus (y_3 \oplus b \oplus c \oplus d)) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)(a \oplus y_3 \oplus b \oplus c) \\
&= \exists\, a, b, c.(y_2 \equiv a \oplus b \oplus c)(a \oplus y_3 \oplus b \oplus c) \\
&= \exists\, a, b, c.(y_2 \equiv a \oplus b \oplus c)(y_3 \equiv a \oplus b \oplus c)' \\
&= y_2 \oplus y_3
\end{aligned}
$$

Similary, we calculate the image of the off-set of f_1 on $\{y_2, y_3\}$:

$$
\begin{aligned}
h_{\text{off}}(y_2, y_3) &= \exists\, a, b, c, d, e.(a \oplus d \oplus e)'(y_2 \equiv a \oplus b \oplus c)(y_3 \equiv b \oplus c \oplus d \oplus e) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)((a \oplus d)'(y_3 \oplus b \oplus c \oplus d)' \vee (a \oplus d)(y_3 \oplus b \oplus c \oplus d)) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)((a \oplus d) \equiv (y_3 \oplus b \oplus c \oplus d)) \\
&= \exists\, a, b, c, d.(y_2 \equiv a \oplus b \oplus c)(a \equiv y_3 \oplus b \oplus c) \\
&= \exists\, a, b, c.(y_2 \equiv a \oplus b \oplus c)(a \equiv y_3 \oplus b \oplus c) \\
&= \exists\, a, b, c.(y_2 \equiv a \oplus b \oplus c)(y_3 \equiv a \oplus b \oplus c) \\
&= y_2 \equiv y_3
\end{aligned}
$$

We can see that

$$h_{\text{on}}(y_2, y_3) \wedge h_{\text{off}}(y_2, y_3) = (y_2 \oplus y_3) \wedge (y_2 \equiv y_3) = \perp$$

Therefore, f_1 can be re-expressed with $h(y_2, y_3) = y_2 \oplus y_3$.

6 SDC and ODC

(a) (5%) The SDC of the entire network is given by

$$
\begin{aligned}
\sum_{i=1}^{6} y_i \oplus f_i = (y_1 \oplus (x_1 \vee \neg x_2)) \vee (y_2 \oplus (\neg x_2 x_3)) \vee (y_3 \oplus (\neg x_3 \wedge x_4)) \\
\vee (y_4 \oplus ((\neg y_1 \neg y_2 y_3) \vee (\neg y_1 y_2 \neg y_3) \vee (y_1 \neg y_2 \neg y_3) \\
\vee (y_1 y_2 y_3))) \vee (z_1 \oplus (y_1 \vee y_4)) \vee (z_2 \oplus (y_3 \wedge y_4))\,.
\end{aligned}
$$

(b) (5%) Recall that $\text{SDC}_4 = (y_1 \oplus (x_1 \vee \neg x_2)) \vee (y_2 \oplus (\neg x_2 x_3)) \vee (y_3 \oplus (\neg x_3 x_4))$. To make it depend only on $\{y_1, y_2, y_3\}$, we have to universally quantified $\{x_1, x_2, x_3, x_4\}$, so that the local SDC is a underestimate of the global SDC. The resulting formula is

$$\forall\, x_1, x_2, x_3, x_4.(y_1 \oplus (x_1 \vee \neg x_2)) \vee (y_2 \oplus (\neg x_2 x_3)) \vee (y_3 \oplus (\neg x_3 x_4))$$

(c) (5%) We first rewrite $z_1 = (x_1 \vee \neg x_2) \vee y_4$ and $z_2 = (\neg x_3 x_4) \wedge y_4$. Thus, we have

$$\begin{aligned}\mathrm{ODC}_{41} &= \overline{\frac{\partial z_1}{\partial y_4}} \\ &= \overline{(x_1 \vee \neg x_2) \oplus 1} \\ &= x_1 \vee \neg x_2\,,\end{aligned}$$

and

$$\begin{aligned}\mathrm{ODC}_{42} &= \overline{\frac{\partial z_2}{\partial y_4}} \\ &= \overline{0 \oplus (\neg x_3 \wedge x_4)} \\ &= x_3 \vee \neg x_4\,.\end{aligned}$$

It follows that $\mathrm{ODC}_4 = \mathrm{ODC}_{41} \wedge \mathrm{ODC}_{42} = (x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_4)$.

7 Dont' Cares in Local Variables

(a) (5%) We first compute

$$\begin{aligned}\mathrm{DC}_4 &= (\mathrm{ODC}_{41} \vee \mathrm{XDC}_1) \wedge (\mathrm{ODC}_{42} \vee \mathrm{XDC}_2) \\ &= (x_1 \vee \neg x_2 \vee \neg x_1 \neg x_2 \neg x_3 x_4) \wedge (x_3 \vee \neg x_4 \vee x_1 x_2 \neg x_3 x_4) \\ &= (x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_4 \vee x_1 x_2) \\ &= x_1 x_3 \vee x_1 \neg x_4 \vee x_1 x_2 \vee \neg x_2 x_3 \vee \neg x_2 \neg x_4 \\ &= x_1 x_2 \vee \neg x_2 x_3 \vee \neg x_2 \neg x_4\,.\end{aligned}$$

Thus,

$$\begin{aligned}D_4 &= \neg(\exists\, x_1, x_2, x_3, x_4.(y_1 \equiv (x_1 \vee \neg x_2)) \wedge (y_2 \equiv (\neg x_2 x_3)) \wedge (y_3 \equiv (\neg x_3 x_4)) \wedge \neg \mathrm{DC}_4) \\ &= y_2 \vee y_1 \neg y_3\,.\end{aligned}$$

(b) (5%) Recall that $f_4 = (\neg y_1 \neg y_2 y_3) \vee (\neg y_1 y_2 \neg y_3) \vee (y_1 \neg y_2 \neg y_3) \vee (y_1 y_2 y_3)$. Thus, we have to minimize the incompletely specified function given by the following K-map.

y_1/y_2y_3	00	01	11	10
0	0	1	x	x
1	x	0	x	x

It's easy to see that the best implementation is $f_4' = \neg y_1 y_3$.