Cofactoring Using The Positional Cube Notation

October 2024

1 Introduction

In this document we discuss how to compute the cofactor of a function H with respect to an implicant β. We do so using a graphical approach and the positional cube notation. The discussion uses the function $H : \mathbb{B}^4 \to \mathbb{B}$

$$H = a'd + ac + ab'c' \tag{1}$$

Figure 1 shows the function H and the implicants listed in the definition. We want to compute the cofactor of H with respect to the cube

$$\beta = cd \tag{2}$$

So the goal is to evaluate

$$H_\beta = H_{cd} = ? \tag{3}$$

We first do it graphically, and then do it using the positional cube notation.

2 Graphical Approach

Computing the cofactor of H with respect to β corresponds to projecting the function H onto the sub-space defined by β. This is graphically represented in Figure 2. From this graphical analysis we can observe that $\beta = cd$ is contained in H, i.e.,

$$\beta = cd \subseteq H \tag{4}$$

The reason is that every minterm of $\beta = cd$ is also a minterm of H. As you can see from Figure 2, if an implicant is contained in a function H, then the generalized cofactor is a tautology

$$H_\beta = \top \tag{5}$$

This provides an operational strategy for checking if an implicant is contained in a function:

1. Compute the cofactor H_β
2. Check if the cofactor is a tautology.

The question is: how can we do the same thing in a non graphical way? What kind of data structure can we use and how can we manipulate it?

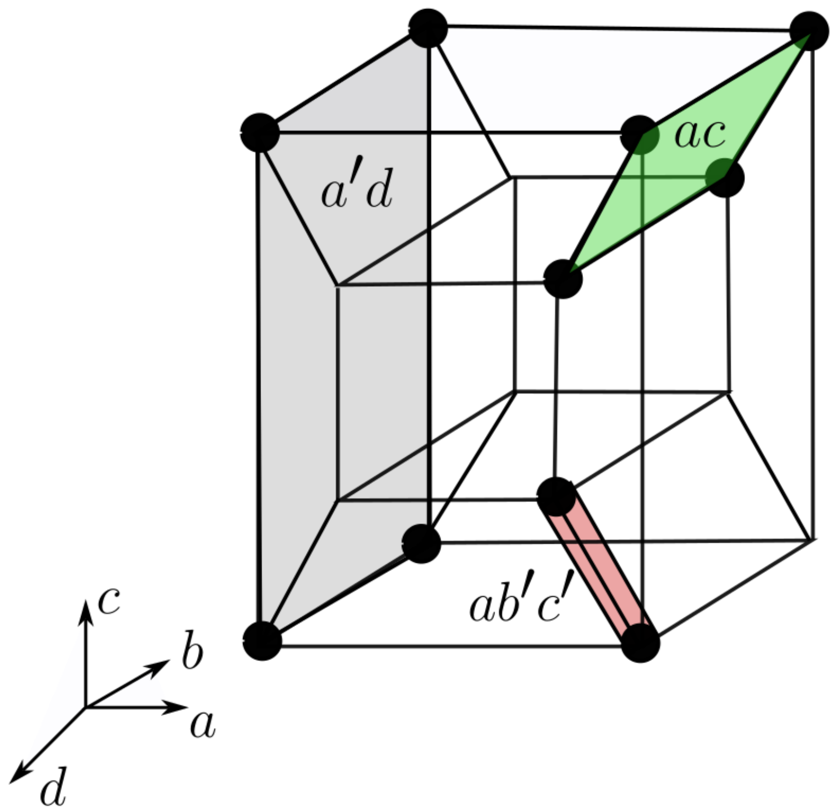

Figure 1: Initial cube and the implicants used for the cover

3 Positional Cube Notation Approach

The positional cube notation is a binary encoding of implicants that allows us to efficiently manipulate Boolean functions through bitwise operations. Remember that our goal is to compute the cofactor of H with respect to the cube $\beta = cd$.

We start by constructing a table in which each row is associated to a cube and each column to a variable. Since the cube $a'd$ intersects the cubes a' and d, we put a 0 and a 1 in the column a and d. Since the cube $a'd$ intersects both b, b', c and c', we put a *don't care* ($*$) in the column of these variables.

	a	b	c	d
$\alpha_1 = a'd$	0	$*$	$*$	1
$\alpha_2 = ac$	1	$*$	1	$*$
$\alpha_3 = ab'c'$	1	0	0	$*$

Table 1: Step 1 of building the encoding

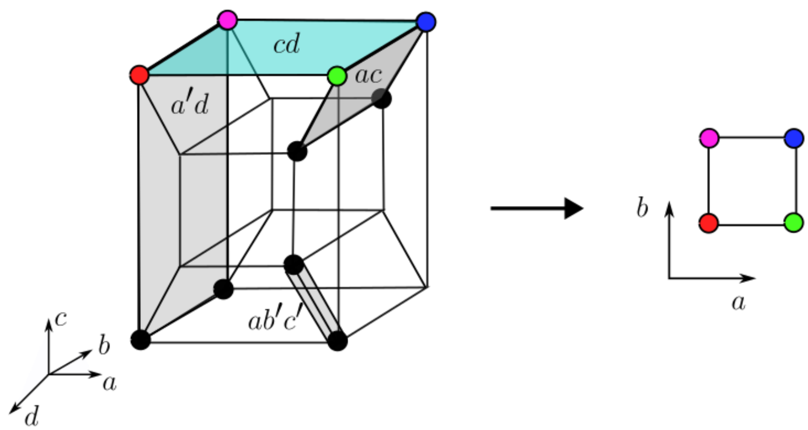

Figure 2: Initial cube and the implicants used for the cover

After this, we introducing the positional cube notation encoding:

$\varnothing$	00
1	01
0	10
$*$	11

Table 2: Encoding of the positional cube notation.

We can now rewrite the table representing the function H as

	a	b	c	d
$\alpha_1 = a'd$	10	11	11	01
$\alpha_2 = ac$	01	11	01	11
$\alpha_3 = ab'c'$	01	10	10	11

Table 3: Step 2: express the function using the positional cube notation.

Similarly, the cube $\beta = cd$ reads

	a	b	c	d
$\beta = cd$	11	11	01	01

Table 4: Step 3: express the cube with respect to which we want to compute the cofactor in the positional cube notation.

Now we need to compute the cofactor H_{cd}. The cofactor is also a Boolean function, so we can represent it with a table just like we did for H. The first thing to notice is that we can create a table whose rows are associated to the same cubes of the function H. Indeed, based on the intuition gained while discussing the graphical approach, the cofactor is the projection of the function onto the subspace in which β is 1. Since it is a projection, it is impossible that new cubes appear, so that we can safely use the same cubes that are present in the cover for H. This implies that the problem of computing the cofactor H_β corresponds to the problem of filling in the entries of Table 8

	a	b	c	d
$a'd_\beta$	$x_{1a}y_{1a}$	$x_{1b}y_{1b}$	$x_{1c}y_{1c}$	$x_{1d}y_{1d}$
ac_β	$x_{2a}y_{2a}$	$x_{2b}y_{2b}$	$x_{2c}y_{2c}$	$x_{2d}y_{2d}$
$ab'c'_\beta$	$x_{3a}y_{3a}$	$x_{3b}y_{3b}$	$x_{3c}y_{2c}$	$x_{3d}y_{3d}$

Table 5: Table of H_β to be defined

The question is: how do we get the terms $x_{ij}y_{ij}$ knowing the positional cube expression of H and β? The answer is divided into two steps:

1. Remove the implicants that have a void intersection with β.

2. Perform the projection of cubes that have a non-void intersection with β.

3.1 Checking for void intersection

Two cubes α and β have void intersection when $\alpha \cap \beta = \varnothing$. When representing the cubes as products of literals, the intersection is in one to one correspondence with the Boolean AND operator. For instance, the cubes $\alpha_3 = ab'c'$ and $\beta = cd$ have void intersection because $\alpha_3 \cdot \beta = ab'(c' \cdot c)d = 0$. Figure 2 graphically shows that these cubes have void intersection. Clearly, you can easily see if two cubes have void intersection just by taking the bitwise AND, as we did for α_3 and β. However, to gain confidence with the positional cube notation, let us see how this bitwise operation translates to operations between covers.

In the positional cube notation we can check for void intersection in the same way, just by performing the bitwise AND between the terms. Let us consider simple literals

1. $x \cdot x' = 0 \Leftrightarrow 01 \cdot 10 = 00$, which is the empty set.

2. $* \cdot x = x \Leftrightarrow 11 \cdot 01 = 01$.

3. $* \cdot x' = x' \Leftrightarrow 11 \cdot 10 = 10$.

If we now consider more complex cubes α and β, built as the product of literals, it should be clear that performing the bitwise AND of each column element in the positional cube notation, it is sufficient that a single column entry is 00 to conclude that the intersection between the cubes is the empty set. Let us do it for the cube $\alpha = ab'c'$:

$$\alpha_3 \cdot \beta = \begin{bmatrix} 01 & 10 & 10 & 11 \end{bmatrix} \cdot \begin{bmatrix} 11 & 11 & 01 & 01 \end{bmatrix} = \begin{bmatrix} 01 & 10 & 00 & 01 \end{bmatrix}$$

This corresponds to do the following

$$\alpha_3 \cdot \beta = ab'c' \cdot cd = (a=1) \cap (b=0) \cap (c=0) \cap (d=*) \cap (a=*) \cap (b=*) \cap (c=1) \cap (d=1) =$$

$$\alpha_3 \cdot \beta = ((a=1) \cap (a=*)) \cap ((b=0) \cap (b=*)) \cap ((c=0) \cap (c=1)) \cap ((d=*) \cap (d=1))$$

$$\alpha_3 \cdot \beta = (a=1) \cap (b=0) \cap \varnothing \cap (d=1) = \varnothing$$

After observing that the cube α_3 has void intersection with β, we can remove the last row from the table of the cofactor H_β In the next section we see how

	a	b	c	d
$a'd_\beta$	$x_{1a}y_{1a}$	$x_{1b}y_{1b}$	$x_{1c}y_{1c}$	$x_{1d}y_{1d}$
ac_β	$x_{2a}y_{2a}$	$x_{2b}y_{2b}$	$x_{2c}y_{2c}$	$x_{2d}y_{2d}$

Table 6: Table of H_β without the void intersections.

to determine the last terms.

3.2 Projection of cubes with non-void intersection

From the previous step we know that $\alpha_1 = a'd$ and $\beta = cd$ have a non-void intersection. How can we compute the cofactor $\alpha_{1,\beta}$? The recipe in the positional cube notation is the following

	a	b	c	d
$a'd_\beta$	$x_{1a}y_{1a} = (10) + (11)'$	$x_{1b}y_{1b} = (11) + (11)'$	$x_{1c}y_{1c} = (11) + (01)'$	$x_{1d}y_{1d} = (01) + (01)'$
ac_β	$x_{2a}y_{2a} = (01) + (11)'$	$x_{2b}y_{2b} = (11) + (11)'$	$x_{2c}y_{2c} = (01) + (01)'$	$x_{2d}y_{2d} = (11) + (01)'$

Table 7: Application of the formula $(H_\beta)_{i,j} = H_{i,j} + (\beta_j)'$.

That gives us

	a	b	c	d
$a'd_\beta$	10	11	11	11
ac_β	01	11	11	11

Table 8: Result using the positional cube notation.

But where does this formula come from? If you just want to pass the exam, you can stop here, apply the formula, and proceed using the information at page 297. If you are curious, you can use the following considerations to better understand the formula $(H_\beta)_{i,j} = H_{i,j} + (\beta_j)'$.

3.3 Understanding the formula

First, let us notice that we can perform the cofactoring one variable at a time.

$$H_{\beta=cd} = (H_c)_d = (H_d)_c$$

Let us consider H_c. The projections onto the space where $c = 1$ are the positive cofactor with respect to c

- $\alpha_{1,c} = a'd_c = a'd$
- $\alpha_{2,c} = ac_c = a$

In the positional cube notation this becomes

- $\alpha_{1,c} = \begin{bmatrix}10 & 11 & 11 & 01\end{bmatrix} ? \begin{bmatrix}11 & 11 & 01 & 11\end{bmatrix} = \begin{bmatrix}10 & 11 & 11 & 01\end{bmatrix}$
- $\alpha_{2,c} = \begin{bmatrix}01 & 11 & 01 & 11\end{bmatrix} ? \begin{bmatrix}11 & 11 & 01 & 11\end{bmatrix} = \begin{bmatrix}01 & 11 & 11 & 11\end{bmatrix}$

Next, we can project onto the d subspace

- $\alpha_{1,cd} = \begin{bmatrix}10 & 11 & 11 & 01\end{bmatrix} ? \begin{bmatrix}11 & 11 & 11 & 01\end{bmatrix} = \begin{bmatrix}10 & 11 & 11 & 11\end{bmatrix}$
- $\alpha_{2,cd} = \begin{bmatrix}01 & 11 & 11 & 11\end{bmatrix} ? \begin{bmatrix}11 & 11 & 11 & 01\end{bmatrix} = \begin{bmatrix}01 & 11 & 11 & 11\end{bmatrix}$

How can we find the operation to be applied to obtain the projection? Let us formulate it as a synthesis problem. In particular, we want to find the operation that we should perform to get the entry $(H_\beta)_{i,j} = H_{i,j} ? \beta_j$. To do so let us fill in a table with all possible cases. We will then consider all the terms one by one to explain how we derived them:

$$\begin{pmatrix} H_{i,j} & \beta_j & (H_{\beta,i,j}) \\ 00 & 00 & 00 \\ 00 & 01 & 00 \\ 00 & 10 & 00 \\ 00 & 11 & 00 \\ 01 & 00 & 00 \\ 01 & 01 & 11 \\ 01 & 10 & 00 \\ 01 & 11 & 01 \\ 10 & 00 & 00 \\ 10 & 01 & 00 \\ 10 & 10 & 11 \\ 10 & 11 & 10 \\ 11 & 00 & 00 \\ 11 & 01 & 11 \\ 11 & 10 & 11 \\ 11 & 11 & 11 \end{pmatrix}$$

The cases with $(H_{\beta,i,j}) = \varnothing$ are the ones that can be identified as void intersections, so we can focus on the remaining ones

$$\begin{pmatrix} H_{i,j} & \beta_j & (H_{\beta,i,j}) \\ 01 & 01 & 11 \\ 01 & 11 & 01 \\ 10 & 10 & 11 \\ 10 & 11 & 10 \\ 11 & 01 & 11 \\ 11 & 10 & 11 \\ 11 & 11 & 11 \end{pmatrix}$$

Let us analyze them one by one:

1. $H_{i,j} = 01$ and $\beta_j = 01$ could be the c column for ac when cofactoring with respect to c. Clearly, the cofactor is a, so c becomes a don't care in the cofactor, and hence $(H_{\beta,i,j}) = 11$.

2. $H_{i,j} = 01$ and $\beta_j = 11$ could be the c column for ac when cofactoring with respect to b. Clearly, the cofactor is ac, so c remains care value and $(H_{\beta,i,j}) = 01$.

3. $H_{i,j} = 11$ and $\beta_j = 01$ could be the b column for ac when cofactoring with respect to c. The cofactor is a, so b remains a *don't care* and $(H_{\beta,i,j}) = 11$.

In this way it is possible to fill in all possible cases.

$$\begin{pmatrix} H_{i,j} & \beta_j & (H_{\beta,i,j}) \\ 01 & 01 & 11 \\ 01 & 11 & 01 \\ 10 & 10 & 11 \\ 10 & 11 & 10 \\ 11 & 01 & 11 \\ 11 & 10 & 11 \\ 11 & 11 & 11 \end{pmatrix}$$

Finally, to get the Boolean expression, we isolate the first and the second bit of each column:

$$\begin{pmatrix} H_{i,j} & \beta_j & (H_{\beta,i,j}) \\ 0|1 & 0|1 & 1|1 \\ 0|1 & 1|1 & 0|1 \\ 1|0 & 1|0 & 1|1 \\ 1|0 & 1|1 & 1|0 \\ 1|1 & 0|1 & 1|1 \\ 1|1 & 1|0 & 1|1 \\ 1|1 & 1|1 & 1|1 \end{pmatrix} \rightarrow \begin{pmatrix} H^1_{i,j} & \beta^1_j & (H^1_{\beta,i,j}) \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \Big| \begin{pmatrix} H^2_{i,j} & \beta^2_j & (H^2_{\beta,i,j}) \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

By isolating the unique terms, it is possible to see that

$$\begin{pmatrix} H^1_{i,j} & \beta^1_j & (H^1_{\beta,i,j}) \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} | \begin{pmatrix} H^2_{i,j} & \beta^2_j & (H^2_{\beta,i,j}) \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Which finally yields $H^k_{\beta,i,j} = H^k_{i,j} + (\beta^k_j)'$