Vectorialization of the Code

Find an updated version in the following GitHub page: https://github.com/Pierre-Botteron/Algebra-of-Boxes-code [BC23].

Contents

I. $\mathcal{NS}$ seen as $\subseteq \mathcal{M}_{4\times 4}([0,1])$

We use the following correspondence (also used in [FWW09]):

$$\begin{array}{ccc} \mathtt{P} \in \mathcal{F}(\{0,1\}^4,[0,1]) & \longleftrightarrow & M_{\mathtt{P}} = \begin{bmatrix} \mathtt{P}(0,0\,|\,0,0) & \mathtt{P}(0,1\,|\,0,0) & \mathtt{P}(1,0\,|\,0,0) & \mathtt{P}(1,1\,|\,0,0) \\ \mathtt{P}(0,0\,|\,0,1) & \mathtt{P}(0,1\,|\,0,1) & \mathtt{P}(1,0\,|\,0,1) & \mathtt{P}(1,1\,|\,0,1) \\ \mathtt{P}(0,0\,|\,1,0) & \mathtt{P}(0,1\,|\,1,0) & \mathtt{P}(1,0\,|\,1,0) & \mathtt{P}(1,1\,|\,1,0) \\ \mathtt{P}(0,0\,|\,1,1) & \mathtt{P}(0,1\,|\,1,1) & \mathtt{P}(1,0\,|\,1,1) & \mathtt{P}(1,1\,|\,1,1) \end{bmatrix} \in \mathcal{M}_{4\times 4}([0,1]) \\ \mathtt{P}(a,b\,|x,y) & \longleftrightarrow & M_{\mathtt{P}}[i,j] \qquad \text{with } i = 2x+y \text{ and } j = 2a+b\,. \end{array}$$

$\rightsquigarrow$ The non-signalling condition in $\mathcal{F}(\{0,1\}^4,[0,1])$ is:

$$\begin{aligned} &\forall a,x \in \{0,1\}, && \sum_{b\in\{0,1\}} \mathtt{P}(a,b\,|\,x,0) = \sum_{b\in\{0,1\}} \mathtt{P}(a,b\,|\,x,1) \\ &\forall b,y \in \{0,1\}, && \sum_{a\in\{0,1\}} \mathtt{P}(a,b\,|\,0,y) = \sum_{a\in\{0,1\}} \mathtt{P}(a,b\,|\,1,y)\,, \end{aligned}$$

and it becomes in $\mathcal{M}_{4\times 4}([0,1])$ as follows:

$$\left\{\begin{array}{lcl} M_{\mathtt{P}}[0,0] + M_{\mathtt{P}}[0,1] &=& M_{\mathtt{P}}[1,0] + M_{\mathtt{P}}[1,1] \\ M_{\mathtt{P}}[2,0] + M_{\mathtt{P}}[2,1] &=& M_{\mathtt{P}}[3,0] + M_{\mathtt{P}}[3,1] \\ M_{\mathtt{P}}[0,2] + M_{\mathtt{P}}[0,3] &=& M_{\mathtt{P}}[1,2] + M_{\mathtt{P}}[1,3] \\ M_{\mathtt{P}}[2,2] + M_{\mathtt{P}}[2,3] &=& M_{\mathtt{P}}[3,2] + M_{\mathtt{P}}[3,3] \end{array}\right. \qquad \left\{\begin{array}{lcl} M_{\mathtt{P}}[0,0] + M_{\mathtt{P}}[0,2] &=& M_{\mathtt{P}}[2,0] + M_{\mathtt{P}}[2,2] \\ M_{\mathtt{P}}[1,0] + M_{\mathtt{P}}[1,2] &=& M_{\mathtt{P}}[3,0] + M_{\mathtt{P}}[3,2] \\ M_{\mathtt{P}}[0,1] + M_{\mathtt{P}}[0,3] &=& M_{\mathtt{P}}[2,1] + M_{\mathtt{P}}[2,3] \\ M_{\mathtt{P}}[1,1] + M_{\mathtt{P}}[1,3] &=& M_{\mathtt{P}}[3,1] + M_{\mathtt{P}}[3,3] \end{array}\right. \tag{1}$$

$\rightsquigarrow$ The condition of being a probability distribution in $\mathcal{F}(\{0,1\}^4,[0,1])$ is:

$$\forall x,y \in \{0,1\}, \qquad \sum_{a,b} \mathtt{P}(a,b\,|\,x,y) = 1\,,$$

and it becomes in $\mathcal{M}_{4\times 4}([0,1])$ that the sums of the lines make 1:

$$\forall i \in \{0,3\}, \qquad M_{\mathtt{P}}[i,0] + M_{\mathtt{P}}[i,1] + M_{\mathtt{P}}[i,2] + M_{\mathtt{P}}[i,3] = 1\,. \tag{2}$$

The set of non-signalling boxes $\mathcal{NS}$ is seen as the convex subset of $\mathcal{M}_{4\times 4}([0,1])$ that satisfies the equations (1) and (2). $\square$

II. $\mathcal{W}$ seen as $\subseteq [0,1]^{32}$

A wiring between two boxes is the data of six functions $\mathsf{W} = (f_1, g_1, f_2, g_2, f_3, g_3)$, see the drawing below. The functions f_i and g_j each have input bits, and the output set is either $\{0,1\}$ (when the wiring is *deterministic*), or the interval $[0,1]$ (when the wiring is *mixed*). For a wiring to be valid, we require that $f_1(x, a_2)$ and $f_2(x, a_1)$ do *not* both depend on the second variable a same value of x, and similarly for g_1 and g_2:

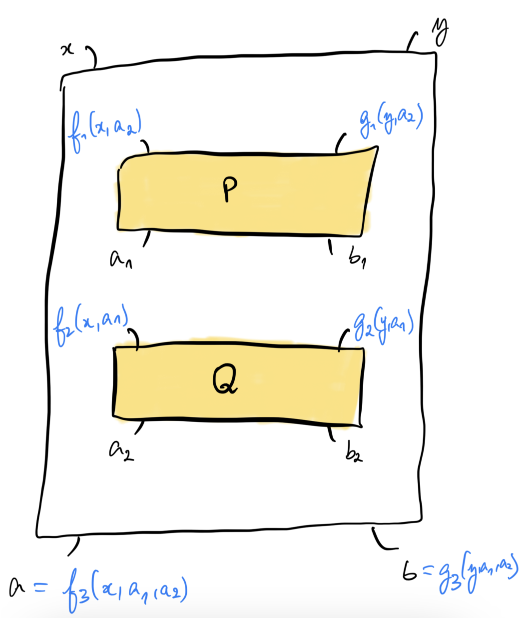
$$\forall x \in \{0,1\}, \quad \Big[f_1(x,0) = f_1(x,1) \quad \text{or} \quad f_2(x,0) = f_2(x,1) \Big],$$
$$\forall y \in \{0,1\}, \quad \Big[g_1(y,0) = g_1(y,1) \quad \text{or} \quad g_2(y,0) = g_2(y,1) \Big],$$

that is to say:

$$\forall x \in \{0,1\}, \quad \Big(f_1(x,0) - f_1(x,1)\Big)\Big(f_2(x,0) - f_2(x,1)\Big) \;=\; 0\,, \tag{3}$$
$$\forall y \in \{0,1\}, \quad \Big(g_1(y,0) - g_1(y,1)\Big)\Big(g_2(y,0) - g_2(y,1)\Big) \;=\; 0\,. \tag{4}$$

Those conditions may be seen as “causility" conditions for Alice and Bob. We then denote by $\mathcal{W}$ the set of all mixed wirings.

$$\mathcal{W} := \Big\{\text{mixed wirings}\Big\}\,,$$



Correspondence. We use the following correspondence:

$$\mathsf{W} = (f_1, g_1, f_2, g_2, f_3, g_3) \in \mathcal{W} \quad \longleftrightarrow \quad (w_0, \dots, w_{31}) \in [0,1]^{32}\,,$$

such that:

- $f_1(x, a_2) = w_{2x+a_2}$,
- $g_1(y, b_2) = w_{2y+b_2+4}$,
- $f_2(x, a_1) = w_{2x+a_1+8}$,
- $g_2(y, b_1) = w_{2y+b_1+12}$,
- $f_3(x, a_1, a_2) = w_{4x+2a_1+a_2+16}$,
- $g_3(y, b_1, b_2) = w_{4y+2b_1+b_2+24}$.

Conditions (3) and (4) of valid wiring are translated as follows:

$$\begin{cases} (w_0 - w_1)(w_8 - w_9) &= 0\,, \\ (w_2 - w_3)(w_{10} - w_{11}) &= 0\,, \end{cases} \qquad \begin{cases} (w_4 - w_5)(w_{12} - w_{13}) &= 0\,, \\ (w_6 - w_7)(w_{14} - w_{15}) &= 0\,. \end{cases} \tag{5}$$

$\mathcal{W}$ is seen as the subset of $[0,1]^{32}$ that satisfies equation (5). We will optimize on W_2. □

III. Maximize Φ

Given two non-signalling boxes $\mathtt{P}, \mathtt{Q} \in \mathcal{NS}$, we want to solve the following problem:

$$\begin{cases} \text{Maximize} & \Phi(\mathsf{W}) := \mathtt{CHSH}\Big(\mathtt{P} \boxtimes_{\mathsf{W}} \mathtt{Q}\Big)\,, \\ \text{subject to} & \mathsf{W} \in \mathcal{W}\,. \end{cases}$$

We do the Projected Gradient Descent algorithm: we project the wiring into the $\mathcal{W}$ after each iteration in order to always satisfy the constraint:

$$\mathsf{W}^{k+1} = \operatorname{proj}\Big(\mathsf{W}^k + \alpha\, \nabla\Phi(\mathsf{W}^k)\Big)\,.$$

Here are more details about the definition of Φ:

(1) Mixed bits in the box $\mathtt{P}$. According to the definition, a box $\mathtt{P}(a, b \,|\, x, y)$ is written with classical bits $a, b, x, y \in \{0, 1\}$. Here we need to generalize it to $\mathtt{P}(a, b \,|\, \alpha, \beta)$ with $a, b \in \{0, 1\}$ and *miwed bits* $\alpha, \beta \in [0, 1]$. It is the unique affine function in α and β that satisfies the good conditions on the classical bits:

$$\begin{aligned} \mathtt{P}\Big(a,\, b \,|\, \alpha,\, \beta\Big) \;:=\; & \mathtt{P}\Big(a,\, b \,|\, 0,\, 0\Big) \times [1-\alpha] \times [1-\beta] \;+\; \mathtt{P}\Big(a,\, b \,|\, 0,\, 1\Big) \times [1-\alpha] \times \beta \\ & +\; \mathtt{P}\Big(a,\, b \,|\, 1,\, 0\Big) \times \alpha \times [1-\beta] \;+\; \mathtt{P}\Big(a,\, b \,|\, 1,\, 1\Big) \times \alpha \times \beta\,. \end{aligned}$$

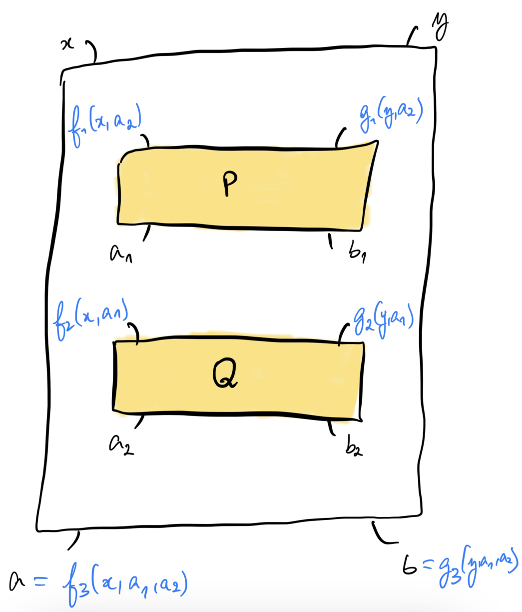
(2) Box product. Given two boxes $\mathtt{P}, \mathtt{Q} \in \mathcal{NS}$ and a mixed wiring $\mathsf{W} = (f_1, g_1, f_2, g_2, f_3, g_3) \in \mathcal{W}$, the product is defined as:

$$\begin{aligned} \mathtt{P} \boxtimes_{\mathsf{W}} \mathtt{Q}(a, b \,|\, x, y) \;=\; \sum_{a_1, a_2, b_1, b_2 \in \{0,1\}} & \mathtt{P}\Big(a_1,\, b_1 \,|\, f_1(x, a_2),\, g_1(y, b_2)\Big)\, \mathtt{Q}\Big(a_2,\, b_2 \,|\, f_2(x, a_1),\, g_2(y, b_1)\Big) \\ & \times \Big[(1 - f_3(x, a_1, a_2))\mathbb{1}_{a=0} + f_3(x, a_1, a_2)\mathbb{1}_{a=1}\Big] \\ & \times \Big[(1 - g_3(y, b_1, b_2))\mathbb{1}_{b=0} + g_3(y, b_1, b_2)\mathbb{1}_{b=1}\Big]\,, \end{aligned}$$

where in blue are represented the mixed bits (used point (1)). We get:

$$
\begin{aligned}
= \sum_{a_1,a_2,b_1,b_2\in\{0,1\}} & \Big[\mathtt{P}\big(a_1,\, b_1\,|\,0,\,0\big)(1-f_1)\,(1-g_1) + \mathtt{P}\big(a_1,\, b_1\,|\,0,\,1\big)(1-f_1)\,g_1 \\
& \quad + \mathtt{P}\big(a_1,\, b_1\,|\,1,\,0\big)\,f_1\,(1-g_1) + \mathtt{P}\big(a_1,\, b_1\,|\,1,\,1\big)\,f_1\,g_1\Big] \\
& \times\Big[\mathtt{Q}\big(a_2,\, b_2\,|\,0,\,0\big)(1-f_2)\,(1-g_2) + \mathtt{Q}\big(a_2,\, b_2\,|\,0,\,1\big)(1-f_2)\,g_2 \\
& \quad + \mathtt{Q}\big(a_2,\, b_2\,|\,1,\,0\big)\,f_2\,(1-g_2) + \mathtt{Q}\big(a_2,\, b_2\,|\,1,\,1\big)\,f_2\,g_2\Big] \\
& \times\Big[(1-f_3)\,\mathbb{1}_{a=0} + f_3\,\mathbb{1}_{a=1}\Big] \times \Big[(1-g_3)\,\mathbb{1}_{b=0} + g_3\,\mathbb{1}_{b=1}\Big]\,,
\end{aligned}
$$

Notice that the function $\mathsf{W} \mapsto \mathtt{P} \boxtimes_{\mathsf{W}} \mathtt{Q}$ is affine in each variable.



(3) $\mathtt{CHSH}$-value function. Given a box $\mathtt{P} \in \mathcal{NS}$, its $\mathtt{CHSH}$ is defined as:

$$
\mathtt{CHSH}(\mathtt{P}) := \mathbb{P}\Big(\mathtt{P} \text{ gagne à } \mathtt{CHSH}\Big) = \sum_{x,y\in\{0,1\}} \tfrac{1}{4} \sum_{a,b\in\{0,1\}} \mathtt{P}(a,b\,|\,x,y)\,\mathbb{1}_{a\oplus b = xy}\,. \tag{6}
$$

This function $\mathtt{CHSH}$ is linear in $\mathtt{P}$.

IV. Vectorization of Φ

We vectorize the function Φ in order to have many efficient gradient descents in parallel.

(1) Vectorization of $\mathtt{CHSH}(\cdot)$. Use Section I. to have the following expression:

$$
\boxed{\mathtt{CHSH}(\mathtt{P}) = \mathtt{sum\_coeff}(\mathtt{P} * M)}\,,
$$

where $\texttt{sum\_coeff}(A)$ is the sum of the coefficients of the matrix A, where $*$ is the component-wise multiplication, and where M is the following matrix:

$$M := \begin{bmatrix} 1/4 & 0 & 0 & 1/4 \\ 1/4 & 0 & 0 & 1/4 \\ 1/4 & 0 & 0 & 1/4 \\ 0 & 1/4 & 1/4 & 0 \end{bmatrix}.$$

(It is a scalar product.)

(2) Vectorization of $\mathtt{P} \boxtimes_W \mathtt{Q}$. We write $\mathtt{P}, \mathtt{Q} \in [0,1]^{16}$ and $W \in [0,1]^{32}$ as in previous sections. We have:

$$\begin{aligned}
\mathtt{P} \boxtimes_W \mathtt{Q}\Big(a, b \,|\, x, y\Big) = \sum_{a_1,a_2,b_1,b_2 \in \{0,1\}} & \Big[\mathtt{P}\Big(a_1,\, b_1 \,|\, 0,\, 0\Big)(1-f_1)\,(1-g_1) + \mathtt{P}\Big(a_1,\, b_1 \,|\, 0,\, 1\Big)(1-f_1)\, g_1 \\
& \quad + \mathtt{P}\Big(a_1,\, b_1 \,|\, 1,\, 0\Big)\, f_1\,(1-g_1) + \mathtt{P}\Big(a_1,\, b_1 \,|\, 1,\, 1\Big)\, f_1\, g_1\Big] \\
& \times \Big[\mathtt{Q}\Big(a_2,\, b_2 \,|\, 0,\, 0\Big)(1-f_2)\,(1-g_2) + \mathtt{Q}\Big(a_2,\, b_2 \,|\, 0,\, 1\Big)(1-f_2)\, g_2 \\
& \quad + \mathtt{Q}\Big(a_2,\, b_2 \,|\, 1,\, 0\Big)\, f_2\,(1-g_2) + \mathtt{Q}\Big(a_2,\, b_2 \,|\, 1,\, 1\Big)\, f_2\, g_2\Big] \\
& \times \Big[(1-f_3)\,\mathbb{1}_{a=0} + f_3\,\mathbb{1}_{a=1}\Big] \times \Big[(1-g_3)\,\mathbb{1}_{b=0} + g_3\,\mathbb{1}_{b=1}\Big],
\end{aligned}$$

where $f_1 = f_1(x, a_2)$, $f_2 = f_2(x, a_1)$, $f_3 = f_3(x, a_1, a_2)$ and $g_1 = g_1(y, b_2)$, $g_2 = g_2(y, b_1)$, $g_3 = g_3(y, b_1, b_2)$.

The elements in green and blue are the coefficients of the following respective 4×4 matrices:

$$M_{(f_1(x,.),g_1(y,.))} \cdot M_{\mathtt{P}} \qquad\qquad \Big(M_{(f_2(x,.),g_2(y,.))} \cdot M_{\mathtt{Q}}\Big)^{\top},$$

where $\cdot$ is the usual matrix product, and $M_{\mathtt{P}}$ is the matrix of the box $\mathtt{P}$:

$$M_{\mathtt{P}} := \begin{bmatrix} \mathtt{P}\Big(0,\,0\,|\,0,\,0\Big) & \mathtt{P}\Big(0,\,1\,|\,0,\,0\Big) & \mathtt{P}\Big(1,\,0\,|\,0,\,0\Big) & \mathtt{P}\Big(1,\,1\,|\,0,\,0\Big) \\ \mathtt{P}\Big(0,\,0\,|\,0,\,1\Big) & \mathtt{P}\Big(0,\,1\,|\,0,\,1\Big) & \mathtt{P}\Big(1,\,0\,|\,0,\,1\Big) & \mathtt{P}\Big(1,\,1\,|\,0,\,1\Big) \\ \mathtt{P}\Big(0,\,0\,|\,1,\,0\Big) & \mathtt{P}\Big(0,\,1\,|\,1,\,0\Big) & \mathtt{P}\Big(1,\,0\,|\,1,\,0\Big) & \mathtt{P}\Big(1,\,1\,|\,1,\,0\Big) \\ \mathtt{P}\Big(0,\,0\,|\,1,\,1\Big) & \mathtt{P}\Big(0,\,1\,|\,1,\,1\Big) & \mathtt{P}\Big(1,\,0\,|\,1,\,1\Big) & \mathtt{P}\Big(1,\,1\,|\,1,\,1\Big) \end{bmatrix},$$

(when we launch the gradient descent, we compute this matrix only once since it does not depend on the wiring W) (or maybe, use this notation in Annexe A instead of a vector of $[0,1]^{16}$), and $M_{(f,g)}$ is the following matrix:

$$\begin{aligned}
M_{(f,g)} &= \begin{bmatrix} (1-f(0))\,(1-g(0)) & (1-f(0))\,g(0) & f(0)\,(1-g(0)) & f(0)\,g(0) \\ (1-f(0))\,(1-g(1)) & (1-f(0))\,g(1) & f(0)\,(1-g(1)) & f(0)\,g(1) \\ (1-f(1))\,(1-g(0)) & (1-f(1))\,g(0) & f(1)\,(1-g(0)) & f(1)\,g(0) \\ (1-f(1))\,(1-g(1)) & (1-f(1))\,g(1) & f(1)\,(1-g(1)) & f(1)\,g(1) \end{bmatrix} \\
&= \begin{bmatrix} (1-f(0)) & (1-f(0)) & f(0) & f(0) \\ (1-f(0)) & (1-f(0)) & f(0) & f(0) \\ (1-f(1)) & (1-f(1)) & f(1) & f(1) \\ (1-f(1)) & (1-f(1)) & f(1) & f(1) \end{bmatrix} * \begin{bmatrix} (1-g(0)) & g(0) & (1-g(0)) & g(0) \\ (1-g(1)) & g(1) & (1-g(1)) & g(1) \\ (1-g(0)) & g(0) & (1-g(0)) & g(0) \\ (1-g(1)) & g(1) & (1-g(1)) & g(1) \end{bmatrix} \\
&= \texttt{vector_to_matrix}\left(\left(F \cdot \begin{bmatrix} f(0) \\ f(1) \\ 1 \\ 1 \end{bmatrix}\right) * \left(G \cdot \begin{bmatrix} g(0) \\ g(1) \\ 1 \\ 1 \end{bmatrix}\right)\right),
\end{aligned}$$

where F and G are constant matrices, see below, and where the map `vector_to_matrix` turns a 16-entry vector into a 4×4 matrix, column by column. We also define:

$$
\begin{aligned}
M_{f_3,x,a} &= (1-a)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^a \left[\begin{array}{cc|cc} f_3(x,0,0) & f_3(x,0,0) & f_3(x,1,0) & f_3(x,1,0) \\ f_3(x,0,0) & f_3(x,0,0) & f_3(x,1,0) & f_3(x,1,0) \\ \hline f_3(x,0,1) & f_3(x,0,1) & f_3(x,1,1) & f_3(x,1,1) \\ f_3(x,0,1) & f_3(x,0,1) & f_3(x,1,1) & f_3(x,1,1) \end{array}\right] \\
&= (1-a)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^a \ \texttt{vector\_to\_matrix}\left(\widetilde{F} \cdot \begin{bmatrix} f_3(x,0,0) \\ f_3(x,0,1) \\ f_3(x,1,0) \\ f_3(x,1,1) \end{bmatrix}\right), \\
M_{g_3,y,b} &= (1-b)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^b \left[\begin{array}{cc|cc} g_3(y,0,0) & g_3(y,1,0) & g_3(y,0,0) & g_3(y,1,0) \\ g_3(y,0,1) & g_3(y,1,1) & g_3(y,0,1) & g_3(y,1,1) \\ \hline g_3(y,0,0) & g_3(y,1,0) & g_3(y,0,0) & g_3(y,1,0) \\ g_3(y,0,1) & g_3(y,1,1) & g_3(y,0,1) & g_3(y,1,1) \end{array}\right] \\
&= (1-b)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^b \ \texttt{vector\_to\_matrix}\left(\widetilde{G} \cdot \begin{bmatrix} g_3(y,0,0) \\ g_3(y,0,1) \\ g_3(y,1,0) \\ g_3(y,1,1) \end{bmatrix}\right).
\end{aligned}
$$

The constant matrices used above are:

$$
F = \begin{bmatrix} -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\qquad
G = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\qquad
\widetilde{F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
\widetilde{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

These notations yield:

$$
\boxed{\mathtt{P} \boxtimes_W \mathtt{Q}\big(a, b \,|\, x, y\big) \;=\; \texttt{sum\_coeff}\Big(\big(M_{(f_1(x,.),g_1(y,.))} \cdot M_{\mathtt{P}}\big) * \big(M_{(f_2(x,.),g_2(y,.))} \cdot M_{\mathtt{Q}}\big)^{\top} * M_{f_3,x,a} * M_{g_3,y,b}\Big)},
$$

with the entry-wise matrix multiplication $*$, and where the function $\texttt{sum\_coeff}(A)$ sums all the coefficients of the matrix A (*i.e.* we mutltiply by $\begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$ on the right and by $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ on the left).

Notations in the Python package, in "evaluate.py"

Here W is no longer a 32-vector, it is a $32 \times 10\,000$-matrix (so that we do multiple gradient descents at the same time).

Using the Einstein notation (exponents mean there is a sum in this variable, so it means there is a hidden multiplication of tensors):

$$
\begin{aligned}
\mathsf{R}_{abxy\alpha}(\mathsf{W}, \mathtt{P}, \mathtt{Q}) &:= \mathtt{P} \boxtimes_{\mathsf{W}} \mathtt{Q}(a, b \,|\, x, y) \\
&= \mathbf{1}^i \Big[\Big(\big(A_{xy\alpha i}{}^{k}(\mathsf{W})\, \mathtt{P}_k{}^j\big)\, \big(B_{xy\alpha}{}^{j\ell}(\mathsf{W})\, \mathtt{Q}_{\ell i}\big)\Big)_{ab}\, C_{abxy\alpha i}{}^{j}(\mathsf{W})\, D_{abxy\alpha i}{}^{j}(\mathsf{W})\Big] \mathbf{1}_j \\
\textit{i.e.} \qquad \mathsf{R}(\mathsf{W}, \mathtt{P}, \mathtt{Q}) &= \boxed{\Big[\mathbf{1}_{2\times 2} \otimes \Big(\big(A(\mathsf{W}) \cdot \mathtt{P}\big) * \big(B(\mathsf{W}) \cdot \mathtt{Q}\big)^{\top(0,1,2,4,3)}\Big) * C(\mathsf{W}) * D(\mathsf{W})\Big] \cdot \mathbf{1}_4 \cdot \mathbf{1}_4},
\end{aligned}
$$

where $a, b, x, y \in \{0, 1\}$, and $i, j, k, \ell \in \{0, 1, 2, 3\}$, and $m, n \in \{0, \dots, 31\}$, and $\alpha \in \{0, 9\,999\}$, and where:

- $\boxed{*}$ is the component-wise multiplication ;
- By convention $\boxed{T \cdot U}$ is the tensor multiplication of T and U through the last entry of T and the first one of U ;
- $\boxed{T^{\top\sigma(0,1,\dots,n-1)}}$ (for some permutation $\sigma \in \mathfrak{S}(\{0,\dots,n-1\})$) is the tensor whose i-th entry is the σ_i-th entry of T (if $|\operatorname{supp}\sigma| = 2$, then $\top\sigma$ is simply the transpose) ;
- $\boxed{\otimes}$ is the Kronecker product. When we do $\mathbb{1} \otimes T$, we duplicate the tensor T so that we obtain a new tensor with more entries ;
- $A_{xy\alpha ik}(\mathsf{W}) = \left(A^{(1)}_{xik}{}^{m}\,\mathsf{W}_{m\alpha} + A^{(2)}_{xik\alpha}\right)_y \left(A^{(3)}_{yik}{}^{n}\,\mathsf{W}_{n\alpha} + A^{(4)}_{yik\alpha}\right)_x,$

  i.e. $\boxed{A(\mathsf{W}) = \left(\left[\mathbf{1}_2 \otimes \left(A^{(1)} \cdot \mathsf{W} + A^{(2)}\right)\right]^{\top(1,0,2,3,4)} * \left[\mathbf{1}_2 \otimes \left(A^{(3)} \cdot \mathsf{W} + A^{(4)}\right)\right]\right)^{\top(0,1,4,2,3)}}$;
- $B_{xy\alpha j\ell}(\mathsf{W}) = \left(B^{(1)}_{xj\ell}{}^{m}\,\mathsf{W}_{m\alpha} + B^{(2)}_{yj\ell\alpha}\right)_y \left(B^{(3)}_{yj\ell}{}^{n}\,\mathsf{W}_{n\alpha} + B^{(4)}_{yj\ell\alpha}\right)_x,$

  i.e. $\boxed{B(\mathsf{W}) = \left(\left[\mathbf{1}_2 \otimes \left(B^{(1)} \cdot \mathsf{W} + B^{(2)}\right)\right]^{\top(1,0,2,3,4)} * \left[\mathbf{1}_2 \otimes \left(B^{(3)} \cdot \mathsf{W} + B^{(4)}\right)\right]\right)^{\top(0,1,4,2,3)}}$;
- $C_{abxy\alpha ij}(\mathsf{W}) = \left(C^{(1)}_{axij}{}^{m}\,\mathsf{W}_{m\alpha} + C^{(2)}_{axij\alpha}\right)_{by},$

  i.e. $\boxed{C(\mathsf{W}) = \left(\mathbf{1}_{2\times 2} \otimes \left(C^{(1)} \cdot \mathsf{W} + C^{(2)}\right)\right)^{\top(2,0,3,1,6,4,5)}}$;
- $D_{abxy\alpha ij}(\mathsf{W}) = \left(D^{(1)}_{byij}{}^{m}\,\mathsf{W}_{m\alpha} + D^{(2)}_{byij\alpha}\right)_{ax},$

  i.e. $\boxed{D(\mathsf{W}) = \left(\mathbf{1}_{2\times 2} \otimes \left(D^{(1)} \cdot \mathsf{W} + D^{(2)}\right)\right)^{\top(0,2,1,3,6,4,5)}}$.

Computation of $A^{(1)}$ and $A^{(2)}$. We have :

$$\begin{bmatrix} (1-f_1(x,0)) & (1-f_1(x,0)) & f_1(x,0) & f_1(x,0) \\ (1-f_1(x,0)) & (1-f_1(x,0)) & f_1(x,0) & f_1(x,0) \\ (1-f_1(x,1)) & (1-f_1(x,1)) & f_1(x,1) & f_1(x,1) \\ (1-f_1(x,1)) & (1-f_1(x,1)) & f_1(x,1) & f_1(x,1) \end{bmatrix}_{ik} = \left(A^{(1)}_{xik}{}^{m}\,\mathsf{W}_{m\alpha} + A^{(2)}_{xik\alpha}\right)_{ik}$$

for $x \in \{0,1\}$, and $i,k \in \{0,\dots,3\}$, and $m \in \{0,\dots,31\}$, and $\alpha \in \{0, 9\,999\}$.
We have for all x and for $k = 0,1$ (1st and 2nd columns of the above matrix) and then $k = 2,3$ (3rd and 4th columns):

$$\left(A^{(1)}_{xi0}{}^{m}\right)_{im} = \left[\begin{matrix} x-1 & 0 & -x & 0 \\ x-1 & 0 & -x & 0 \\ 0 & x-1 & 0 & -x \\ 0 & x-1 & 0 & -x \end{matrix} \quad \mathbf{0}_{4\times 28}\right]_{im} = \left(A^{(1)}_{xi1}{}^{m}\right)_{im},$$
$$= -\left(A^{(1)}_{xi2}{}^{m}\right)_{im} = -\left(A^{(1)}_{xi3}{}^{m}\right)_{im}.$$

And for all x, α:

$$\left(A^{(2)}_{xik\alpha}\right)_{ik} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}_{ik} = \begin{bmatrix} \mathbf{1}_{4\times 2} & \mathbf{0}_{4\times 2} \end{bmatrix}_{ik}.$$

Computation of $A^{(3)}$ and $A^{(4)}$. We have :

$$\begin{bmatrix} (1-g_1(y,0)) & g_1(y,0) & (1-g_1(y,0)) & g_1(y,0) \\ (1-g_1(y,1)) & g_1(y,1) & (1-g_1(y,1)) & g_1(y,1) \\ (1-g_1(y,0)) & g_1(y,0) & (1-g_1(y,0)) & g_1(y,0) \\ (1-g_1(y,1)) & g_1(y,1) & (1-g_1(y,1)) & g_1(y,1) \end{bmatrix}_{ik} = \left(A^{(3)\,n}_{yik} \mathsf{W}_{n\alpha} + A^{(4)}_{yik\alpha} \right)_{ik}$$

for $y \in \{0,1\}$, and $i, k \in \{0, \ldots, 3\}$, and $n \in \{0, \ldots, 31\}$, and $\alpha \in \{0, 9\,999\}$.
We have for all y and $k = 0, 2$ (1st and 3rd columns of the above matrix) and then for $k = 1, 3$ (2nd and 4th columns):

$$\begin{aligned} \left(A^{(3)\,n}_{yi0}\right)_{i,n} &= \left[\begin{array}{ccccccc} \mathbf{0}_{4\times 4} & \begin{smallmatrix} y-1 & 0 & -y & 0 \\ 0 & y-1 & 0 & -y \\ y-1 & 0 & -y & 0 \\ 0 & y-1 & 0 & -y \end{smallmatrix} & \mathbf{0}_{4\times 24} \end{array} \right]_{i,n} = \left(A^{(3)\,n}_{yi2}\right)_{i,n}, \\ &= -\left(A^{(3)\,n}_{yi1}\right)_{i,n} = -\left(A^{(3)\,n}_{yi3}\right)_{i,n}. \end{aligned}$$

And for all y, α:

$$\left(A^{(4)}_{yik\alpha}\right)_{ik} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}_{ik} = \begin{bmatrix} \mathbf{1}_{4\times 1} & \mathbf{0}_{4\times 1} & \mathbf{1}_{4\times 1} & \mathbf{0}_{4\times 1} \end{bmatrix}_{ik}.$$

Computation of $B^{(1)}$ to $B^{(4)}$. After doing as before, we obtain:

$$B^{(2)}_{yj\ell\alpha} = A^{(2)}_{yj\ell\alpha} \qquad \text{and} \qquad B^{(4)}_{yj\ell\alpha} = A^{(4)}_{yj\ell\alpha}.$$

As we just change f_1 into f_2 and g_1 into g_2, we have for all x:

$$\begin{aligned} \left(B^{(1)\,m}_{xj0}\right)_{j,m} &= \left[\begin{array}{ccc} \mathbf{0}_{4\times 8} & \begin{smallmatrix} x-1 & 0 & -x & 0 \\ x-1 & 0 & -x & 0 \\ 0 & x-1 & 0 & -x \\ 0 & x-1 & 0 & -x \end{smallmatrix} & \mathbf{0}_{4\times 20} \end{array} \right]_{j,m} \\ &= \left(B^{(1)\,m}_{xj1}\right)_{j,m} = -\left(B^{(1)\,m}_{xj2}\right)_{j,m} = -\left(B^{(1)\,m}_{xj3}\right)_{j,m}, \end{aligned}$$

and for all y:

$$\begin{aligned} \left(B^{(3)\,n}_{yj0}\right)_{j,n} &= \left[\begin{array}{ccc} \mathbf{0}_{4\times 12} & \begin{smallmatrix} y-1 & 0 & -y & 0 \\ 0 & y-1 & 0 & -y \\ y-1 & 0 & -y & 0 \\ 0 & y-1 & 0 & -y \end{smallmatrix} & \mathbf{0}_{4\times 16} \end{array} \right]_{j,n} \\ &= -\left(B^{(3)\,n}_{yj1}\right)_{j,n} = \left(B^{(3)\,n}_{yj2}\right)_{j,n} = -\left(B^{(3)\,n}_{yj3}\right)_{j,n}. \end{aligned}$$

Computation of $C^{(1)}$ and $C^{(2)}$. We have:

$$-(-1)^a \left[\begin{array}{cc|cc} f_3(x,0,0) & f_3(x,0,0) & f_3(x,1,0) & f_3(x,1,0) \\ f_3(x,0,0) & f_3(x,0,0) & f_3(x,1,0) & f_3(x,1,0) \\ \hline f_3(x,0,1) & f_3(x,0,1) & f_3(x,1,1) & f_3(x,1,1) \\ f_3(x,0,1) & f_3(x,0,1) & f_3(x,1,1) & f_3(x,1,1) \end{array} \right]_{ij} = \left(C^{(1)\;m}_{axij} \mathsf{W}_{m\alpha} \right)_{ij}$$

For all a, x and for $j \in \{0,1\}$ (1st and 2nd columns):

$$\left(C^{(1)\;m}_{axi0}\right)_{i,m} = -(-1)^a \left[\begin{array}{ccc} \mathbf{0}_{4\times 16} & \begin{smallmatrix} 1-x & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 1-x & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & 1-x & 0 & 0 & 0 & x & 0 & 0 \\ 0 & 1-x & 0 & 0 & 0 & x & 0 & 0 \end{smallmatrix} & \mathbf{0}_{4\times 8} \end{array} \right]_{i,m} = \left(C^{(1)\;m}_{axi1}\right)_{i,m},$$

and for $j \in \{2,3\}$ (3rd and 4th columns):

$$\left(C^{(1)\ m}_{axi2}\right)_{i,m} = -(-1)^a \left[\begin{array}{cccccccccc} & 0 & 0 & 1-x & 0 & 0 & 0 & x & 0 & \\ \mathbf{0}_{4\times16} & 0 & 0 & 1-x & 0 & 0 & 0 & x & 0 & \mathbf{0}_{4\times8} \\ & 0 & 0 & 0 & 1-x & 0 & 0 & 0 & x & \\ & 0 & 0 & 0 & 1-x & 0 & 0 & 0 & x & \end{array}\right]_{i,m} = \left(C^{(1)\ m}_{axi3}\right)_{i,m}.$$

And we have $C^{(2)}_{axij\alpha} = (1-a)\,\mathbf{1}_{xij\alpha}$.

Computation of $D^{(1)}$ and $D^{(2)}$. We have:

$$-(-1)^b \left[\begin{array}{cc|cc} g_3(y,0,0) & g_3(y,1,0) & g_3(y,0,0) & g_3(y,1,0) \\ g_3(y,0,1) & g_3(y,1,1) & g_3(y,0,1) & g_3(y,1,1) \\ \hline g_3(y,0,0) & g_3(y,1,0) & g_3(y,0,0) & g_3(y,1,0) \\ g_3(y,0,1) & g_3(y,1,1) & g_3(y,0,1) & g_3(y,1,1) \end{array}\right]_{ij} = \left(D^{(1)\ m}_{byij}\,\mathsf{W}_{m\alpha}\right)_{ij}.$$

For all y and $j \in \{0,2\}$ (1st and 3rd columns):

$$\left(D^{(1)\ m}_{byi0}\right)_{i,m} = -(-1)^b \left[\begin{array}{ccccccccc} & 1-y & 0 & 0 & 0 & y & 0 & 0 & 0 \\ \mathbf{0}_{4\times24} & 0 & 1-y & 0 & 0 & 0 & y & 0 & 0 \\ & 1-y & 0 & 0 & 0 & y & 0 & 0 & 0 \\ & 0 & 1-y & 0 & 0 & 0 & y & 0 & 0 \end{array}\right]_{i,m} = \left(D^{(1)\ m}_{byi2}\right)_{i,m},$$

and for $j \in \{1,3\}$ (2nd and 3rd columns):

$$\left(D^{(1)\ m}_{byi1}\right)_{i,m} = -(-1)^b \left[\begin{array}{ccccccccc} & 0 & 0 & 1-y & 0 & 0 & 0 & y & 0 \\ \mathbf{0}_{4\times24} & 0 & 0 & 0 & 1-y & 0 & 0 & 0 & y \\ & 0 & 0 & 1-y & 0 & 0 & 0 & y & 0 \\ & 0 & 0 & 0 & 1-y & 0 & 0 & 0 & y \end{array}\right]_{i,m} = \left(D^{(1)\ m}_{byi3}\right)_{i,m}.$$

And $D^{(2)}_{byij\alpha} = (1-b)\,\mathbf{1}_{yij\alpha}$.

V. Tangent hyperplane

Define:

$$F : \begin{cases} \mathbb{R}^{32} & \to & \mathbb{R}^4 \\ (w_0,\ldots,w_{31}) & \mapsto & \begin{pmatrix} (w_0-w_1)(w_8-w_9) \\ (w_2-w_3)(w_{10}-w_{11}) \\ (w_4-w_5)(w_{12}-w_{13}) \\ (w_6-w_7)(w_{14}-w_{15}) \end{pmatrix} \end{cases}.$$

The function is F polynomial in each component so differentiable, with:

$$DF(\mathsf{W}) = \left[\begin{array}{ccccccccccccccccc} w_8-w_9 & w_9-w_8 & 0 & 0 & 0 & 0 & 0 & 0 & w_0-w_1 & w_1-w_0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & w_{10}-w_{11} & w_{11}-w_{10} & 0 & 0 & 0 & 0 & 0 & 0 & w_2-w_3 & w_3-w_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{12}-w_{13} & w_{13}-w_{12} & 0 & 0 & 0 & 0 & 0 & 0 & w_4-w_5 & w_5-w_4 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & w_{14}-w_{15} & w_{15}-w_{14} & 0 & 0 & 0 & 0 & 0 & 0 & w_6-w_7 & w_7-w_6 & \end{array}\right],$$

where the 0 on the right is a 4×16 zero block. The tangent hyperplane to $V = \{F = 0\} \subseteq \mathbb{R}^{32}$ at W is:

$$T_{\mathsf{W}}V = \ker\left\{DF(\mathsf{W})\right\} = \left\{H \in \mathbb{R}^{32} : DF(\mathsf{W})H = 0\right\}.$$

Up to re-arranging the lines and columns, this matrix is in the Reduced Row Echelon Form. Looking at the free variables and using $F(\mathsf{W}) = 0$, it yields a following basis for $\ker(DF(\mathsf{W}))$:

$$
\begin{aligned}
\mathcal{B} := & \left\{ \begin{array}{ll} \frac{e_0+e_1}{\sqrt{2}} & \text{if } w_8 \neq w_9 \\ e_0, e_1 & \text{if } w_8 = w_9 \end{array} \right\} \cup \left\{ \begin{array}{ll} \frac{e_8+e_9}{\sqrt{2}} & \text{if } w_0 \neq w_1 \\ e_8, e_9 & \text{if } w_0 = w_1 \end{array} \right\} \\
& \cup \left\{ \begin{array}{ll} \frac{e_2+e_3}{\sqrt{2}} & \text{if } w_{10} \neq w_{11} \\ e_2, e_3 & \text{if } w_{10} = w_{11} \end{array} \right\} \cup \left\{ \begin{array}{ll} \frac{e_{10}+e_{11}}{\sqrt{2}} & \text{if } w_2 \neq w_3 \\ e_{10}, e_{11} & \text{if } w_2 = w_3 \end{array} \right\} \\
& \cup \left\{ \begin{array}{ll} \frac{e_4+e_5}{\sqrt{2}} & \text{if } w_{12} \neq w_{13} \\ e_4, e_5 & \text{if } w_{12} = w_{13} \end{array} \right\} \cup \left\{ \begin{array}{ll} \frac{e_{12}+e_{13}}{\sqrt{2}} & \text{if } w_4 \neq w_5 \\ e_{12}, e_{13} & \text{if } w_4 = w_5 \end{array} \right\} \\
& \cup \left\{ \begin{array}{ll} \frac{e_6+e_7}{\sqrt{2}} & \text{if } w_{14} \neq w_{15} \\ e_6, e_7 & \text{if } w_{14} = w_{15} \end{array} \right\} \cup \left\{ \begin{array}{ll} \frac{e_{14}+e_{15}}{\sqrt{2}} & \text{if } w_6 \neq w_7 \\ e_{14}, e_{15} & \text{if } w_6 = w_7 \end{array} \right\} \\
& \cup \{e_{16}, \dots, e_{31}\},
\end{aligned}
$$

where $(e_0, \dots, e_{31})$ denotes the canonical basis of $\mathbb{R}^{32}$. Note that $12 \leq \dim\Big(\ker DF(\mathsf{W})\Big) \leq 16$. Note as well that this is an ONB. Hence, the projection of W' onto $T_{\mathsf{W}}V$ amounts to simply compute the scalar product of these vectors with W':

$$\operatorname{proj}_{T_{\mathsf{W}}V}(\mathsf{W}') \;=\; \sum_{b \in \mathcal{B}} \langle \mathsf{W}', b \rangle \, b\,.$$

$\Rightarrow$ Conclusion: after coding it in Python, this method doesn't seem efficient: it takes a lot of time and does not give an interesting answer...

$M_{(f,g)}$ can be vectoriazlized as well:

$$
\begin{aligned}
M_{(f,g)} \;&=\; \begin{bmatrix} (1-f(0)) & (1-f(0)) & f(0) & f(0) \\ (1-f(0)) & (1-f(0)) & f(0) & f(0) \\ (1-f(1)) & (1-f(1)) & f(1) & f(1) \\ (1-f(1)) & (1-f(1)) & f(1) & f(1) \end{bmatrix} * \begin{bmatrix} (1-g(0)) & g(0) & (1-g(0)) & g(0) \\ (1-g(1)) & g(1) & (1-g(1)) & g(1) \\ (1-g(0)) & g(0) & (1-g(0)) & g(0) \\ (1-g(1)) & g(1) & (1-g(1)) & g(1) \end{bmatrix} \\
&=\; \texttt{vector\_to\_matrix}\left(\left(F \cdot \begin{bmatrix} f(0) \\ f(1) \\ 1 \\ 1 \end{bmatrix} \right) * \left(G \cdot \begin{bmatrix} g(0) \\ g(1) \\ 1 \\ 1 \end{bmatrix} \right) \right)
\end{aligned}
$$

where F and G are the following constant matrices:

$$
F \;=\; \begin{bmatrix} -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad\qquad G \;=\; \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.
$$

$M_{f_3,x,a}$ and $M_{g_3,y,b}$ can be vectoriazlized as well:

$$
\begin{aligned}
M_{f_3,x,a} &= (1-a)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^a\, \texttt{vector\_to\_matrix}\left(\widetilde{F} \cdot \begin{bmatrix} f_3(x,0,0) \\ f_3(x,0,1) \\ f_3(x,1,0) \\ f_3(x,1,1) \end{bmatrix} \right) \\
M_{g_3,y,b} &= (1-b)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - (-1)^b\, \texttt{vector\_to\_matrix}\left(\widetilde{G} \cdot \begin{bmatrix} g_3(y,0,0) \\ g_3(y,0,1) \\ g_3(y,1,0) \\ g_3(y,1,1) \end{bmatrix} \right),
\end{aligned}
$$

where $\widetilde{F}$ is the following matrix:

$$\widetilde{F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \widetilde{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

References

[**BC23**] Pierre Botteron and Réda Chhaibi. Github page, 2023. https://github.com/Pierre-Botteron/Algebra-of-Boxes-code [Accessed: November 2023].

[**FWW09**] Manuel Forster, Severin Winkler, and Stefan Wolf. Distilling nonlocality. *Phys. Rev. Lett.*, 102:120401, Mar 2009.