We would like to calculate the following integral for two variables as:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(v=x_1)\mathbb{I}[\wedge_p\ x_2>\phi_p(x_1)]e_1e_2\ dx_2dx_1 \tag{1}$$

for $\mathcal{P}$ constraints where

$$e_i=\sum_{j=1}^{n_i}c_{i_j}x_i^j+c_{i_0} \tag{2}$$

which can be written as:

$$\int_{l_1}^{u_1}\delta(v=x_1)e_1\left(\int_{l_2}^{u_2}\mathbb{I}[x_2-\phi(x_1)>0]e_2\ dx_2\right)dx_1 \tag{3}$$

where $l_1\leq x_1\leq u_1,\ l_2\leq x_2\leq u_2$ which will lead to

$$\int_{l_1}^{u_1}\delta(v=x_1)e_1\left(\sum_{k=1}^{n_i}\frac{c_{i_k}}{i+1}(l_j-u_j)^{i+1}+c_{i_0}(u_j-l_j)\right)\Big|_{\max_p\phi_p(x_1)-u_2}^{\min_p\phi_p(x_1)-l_2}\ dx_1 \tag{4}$$

this can be written in following cases:

$$\begin{cases}
\phi_{\mathcal{P}}(x_i)>\phi_{\mathcal{P}-1}(x_i)\wedge\phi_{\mathcal{P}-1}(x_i)>\phi_{\mathcal{P}-2}(x_i)\wedge\ldots\wedge\phi_1(x_i)-l_j>0\wedge\phi_1(x_i)-u_j>0 \\
\qquad :\sum_{k=1}^{n_i}\frac{c_{i_k}}{i+1}(l_j-u_j)^{i+1}+c_{i_0}(u_j-l_j) \\
\phi_{\mathcal{P}}(x_i)>\phi_{\mathcal{P}-1}(x_i)\wedge\phi_{\mathcal{P}-1}(x_i)>\phi_{\mathcal{P}-2}(x_i)\wedge\ldots\wedge\phi_1(x_i)-l_j>0\wedge\phi_1(x_i)-u_j\leq 0 \\
\qquad :\sum_{k=1}^{n_i}\frac{c_{i_k}}{i+1}(l_j-x_i)^{i+1}+c_{i_0}(x_i-l_j) \\
\phi_{\mathcal{P}}(x_i)>\phi_{\mathcal{P}-1}(x_i)\wedge\phi_{\mathcal{P}-1}(x_i)>\phi_{\mathcal{P}-2}(x_i)\wedge\ldots\wedge\phi_1(x_i)-l_j\leq 0\wedge\phi_1(x_i)-u_j>0 \\
: \qquad \sum_{k=1}^{n_i}\frac{c_{i_k}}{i+1}(x_i-u_j)^{i+1}+c_{i_0}(u_j-x_i) \\
\phi_{\mathcal{P}}(x_i)>\phi_{\mathcal{P}-1}(x_i)\wedge\phi_{\mathcal{P}-1}(x_i)>\phi_{\mathcal{P}-2}(x_i)\wedge\ldots\wedge\phi_1(x_i)-l_j\leq 0\wedge\phi_1(x_i)-u_j\leq 0 \\
:0s \\
\phi_{\mathcal{P}}(x_i)>\phi_{\mathcal{P}-1}(x_i)\wedge\phi_{\mathcal{P}-1}(x_i)>\phi_{\mathcal{P}-2}(x_i)\wedge\ldots\wedge\phi_2(x_i)-l_j>0\wedge\phi_2(x_i)-u_j>0 \\
\qquad :\sum_{k=1}^{n_i}\frac{c_{i_k}}{i+1}(l_j-u_j)^{i+1}+c_{i_0}(u_j-l_j) \\
\\
\qquad \vdots
\end{cases} \tag{5}$$

Algorithm 1 Integral: This procedure computs the integral

Input: An expression whose integral is required.

for all case $\in$ cases **do**

    EXP(x_i):=arrangeExpression(case) {So that the overall statement is segmented}

    a := computeEXP(EXP(x_i))

    newcase:=XADD(a)

    Integral(newcase)

end for

Algorithm 2 computeEXP: Computes the given expression EXP(x_i)

for all $token(x_i) \in \mathrm{EXP}(x_i)$ **do**

    if $token(x_i) = \delta(v = f(x_i))$ **then**

        $v := f(x_i)$ {x_j is the variable not in the current integral}

    else

        append($\delta(v = f(x_j))$)

    end if

    if $token(x) = \mathbb{I}[\phi(x_j) - x > 0]$ **then**

        $z := \phi(x_j) - x_i \Rightarrow x_i = \phi(x_j) - z$

        $e_i = \sum_{k=1}^{n_i} c_{i_k} x_i^k + c_{i_0}$ {Replace z into e_i and change the bounds, then build a new case expression}

        append(Equation 5)

    end if

end for