0.1 Representation

For some polynomial

$$p(q_1,...,q_n) = \sum_{(e_1,...,e_n)\in(\mathbb{N}_0)^n, \sum_i e_i \le d} p_{e_1,...,e_n} q_1^{e_1} ... q_n^{e_n}$$

we need to store the coefficients $p_{e_1,...,e_n}$ (in an array). We choose the ordering: $p_{e_1,...,e_n}$ comes before $p_{\bar{e}_1,...,\bar{e}_n}$ exactly if

- $\sum_i e_i < \sum_i \bar{e}_i$ or
- $\sum_i e_i = \sum_i \bar{e}_i$ and $e_{i_{\min}} > \bar{e}_{i_{\min}}$ with $i_{\min} := \min_i(e_i \neq \bar{e}_i)$

so for example if $n = 2, d = 2$, p is represented as

$$p_{0,0} \quad p_{1,0} \quad p_{0,1} \quad p_{2,0} \quad p_{1,1} \quad p_{0,2}$$

for given $\bar{e}_1, ..., \bar{e}_n$ the index into this array can be calculated as follows:

$$I_n(\bar{e}_1,...,\bar{e}_n) = \#\{(e_1,...,e_n) \text{ that come before}\}$$

$$= \left(\sum_{\sigma=0}^{<\sum_i \bar{e}_i} \#\left\{(e_1,...,e_n), \sum_i e_i = \sigma\right\}\right) + \#\left\{(e_1,...,e_n), \sum_i e_i = \sum_i \bar{e}_i, \text{ that come before}\right\}$$

$$= \left(\sum_{\sigma=0}^{<\sum_i \bar{e}_i} \left(\!\binom{n}{\sigma}\!\right)\right) + \sum_{i_{\min}=1}^{n-1} \#\left\{(\bar{e}_1,...,\bar{e}_{i_{\min}-1}, e_{i_{\min}},...,e_n), \sum = \sum_i \bar{e}_i, e_{i_{\min}} > \bar{e}_{i_{\min}}\right\}$$

$$= \left(\sum_{\sigma=0}^{<\sum_i \bar{e}_i} \left(\!\binom{n}{\sigma}\!\right)\right) + \sum_{i_{\min}=1}^{n-1} \sum_{e_{i_{\min}}=\bar{e}_{i_{\min}}+1}^{\sum_{i\ge i_{\min}} \bar{e}_i} \#\left\{(e_{i_{\min}+1},...,e_n), \sum = \left(\sum_{i \ge i_{\min}} \bar{e}_i\right) - e_{i_{\min}}\right\}$$

$$= \left(\sum_{\sigma=0}^{<\sum_i \bar{e}_i} \left(\!\binom{n}{\sigma}\!\right)\right) + \sum_{i_{\min}=1}^{n-1} \sum_{e_{i_{\min}}=\bar{e}_{i_{\min}}+1}^{\sum_{i\ge i_{\min}} \bar{e}_i} \left(\!\binom{n-i_{\min}}{\left(\sum_{i\ge i_{\min}} \bar{e}_i\right) - e_{i_{\min}}}\!\right)$$

$$= \left(\!\binom{\sum_i \bar{e}_i}{n}\!\right) + \sum_{i_{\min}=1}^{n-1} \left(\!\binom{\sum_{i>i_{\min}} \bar{e}_i}{n-i_{\min}}\!\right)$$

$$= \sum_{i_{\min}=0}^{n-1} \left(\!\binom{\sum_{i=i_{\min}+1}^{n} \bar{e}_i}{n-i_{\min}}\!\right)$$

let $E_n(i)$ be the function to get back the exponents such that always:

$$i = I_n(E_n(i))$$

0.2 Algorithm

let p be a polynomial of degree d in n variables and $q_j, j = 1...n$ be polynomials of degrees d_j in n_j variables. we want to calculate the polynomial

$$(p \circ_{(X)} q)(x_1,...,x_m) := p\left(q_1\left(x_{X_1(1)},...,x_{X_1(n_1)}\right),...,q_n\left(x_{X_n(1)},...,x_{X_n(n_n)}\right)\right)$$

where the $X_j(i)$ decide which x is used as which argument to each q_j. with the additional constraints:

- $\forall k = 1 ... m \; \exists j, i : \; X_j(i) = k$
- $\forall j, i_1 < i_2 : \; X_j(i_1) < X_j(i_2)$

so every x is used somewhere, no x is used as multiple arguments in one q_j and the x are used in the same order in each q_j as they are given in. this also means all $n_j \le n$.

$$(p \circ_{(X)} q)(x_1, ..., x_m) = \sum_{(e_1,...,e_n) \in (\mathbb{N}_0)^n, \sum_j e_j \le d} p_{e_1,...,e_n} \prod_{j=1}^{n} \left(\sum_{(e_{j,1},...,e_{j,n_j}) \in (\mathbb{N}_0)^{n_j}, \sum_i e_{j,i} \le d_j} q_{j;e_{j,1},...,e_{j,n_j}} \prod_{i=1}^{n_j} x_{X_j(i)}^{e_{j,i}} \right)^{e_j}$$

$$= \sum_{\substack{(e_1, ..., e_n) \in (\mathbb{N}_0)^n, \sum_j e_j \le d \\ \text{for } (1 \le j \le n, 1 \le k \le e_j) : \\ (e_{j,1,k}, ..., e_{j,n_1,k}) \in (\mathbb{N}_0)^{n_j}, \sum_i e_{j,i,k} \le d_j}} p_{e_1,...,e_n} \prod_{\substack{1 \le j \le n, \\ 1 \le k \le e_j}} q_{j;e_{j,1,k},...,e_{j,n_j,k}} \prod_{i=1}^{n_j} x_{X_j(i)}^{e_{j,i,k}}$$

$$= \sum_{\substack{(e_1, ..., e_n) \in (\mathbb{N}_0)^n, \sum_j e_j \le d \\ \text{for } (1 \le j \le n, 1 \le k \le e_j) : \; 0 \le I_{j,k} < \left(\!\binom{d_j + 1}{n_j}\!\right)}} p_{e_1,...,e_n} \prod_{\substack{1 \le j \le n, \\ 1 \le k \le e_j}} q_{j;E_{n_j}(I_{j,k})} \prod_{i=1}^{n_j} x_{X_j(i)}^{E_{n_j}(I_{j,k})_i}$$

$$= \sum_{\substack{(e_1, ..., e_n) \in (\mathbb{N}_0)^n, \sum_j e_j \le d \\ \text{for } (1 \le j \le n) : \; 0 \le I_{j,1} \le ... \le I_{j,e_j} < \left(\!\binom{d_j + 1}{n_j}\!\right)}} f_{(...)} p_{e_1,...,e_n} \prod_{\substack{1 \le j \le n, \\ 1 \le k \le e_j}} q_{j;E_{n_j}(I_{j,k})} \prod_{i=1}^{n_j} x_{X_j(i)}^{E_{n_j}(I_{j,k})_i}$$

where ordering the I_j, k in the last step is compensated with the factor

$$f_{(...)} = \prod_{j=1}^{n} \binom{e_j}{\#\{k, I_{j,k} = ...\}, ...}$$

when comparing with the target form

$$(p \circ_{(X)} q)(x_1, ..., x_m) = r\left(x_1, ..., x_m\right) = \sum_{(e_1,...,e_m) \in (\mathbb{N}_0)^m} r_{e_1,...,e_m} x_1^{e_1} ... x_m^{e_m}$$

we find that the resulting polynomial can be calculated as:

$$\text{for} \begin{pmatrix} (e_1, ..., e_n) \in (\mathbb{N}_0)^n, \sum_j e_j \le d \\ \text{for } (1 \le j \le n) : \; 0 \le I_{j,1} \le ... \le I_{j,e_j} < \left(\!\binom{d_j + 1}{n_j}\!\right) \end{pmatrix} : r\left[I_{(...)}\right] \mathrel{+}= f_{(...)} p[I(e_1, ..., e_n)] \prod_{\substack{1 \le j \le n, \\ 1 \le k \le e_j}} q_j\left[I_{j,k}\right]$$

where

$$I_{(...)} = I_m\left(\left(\sum_{i,j,k: \, X_j(i)=1} E\left(I_{j,k}\right)_i\right), ..., \left(\sum_{i,j,k: \, X_j(i)=m} E\left(I_{j,k}\right)_i\right)\right)$$

the result can be rewritten by combining the numbers j and k into l:

$$\text{for}\begin{pmatrix} 1 \leq D \leq d, 1 \leq j_1 \leq \ldots \leq j_D \leq n, (I_1, \ldots, I_D) \\ \forall l : 0 \leq I_l < \left(\!\!\binom{d_{j_l}+1}{n_{j_l}}\!\!\right), \ \forall l : j_l < j_{l+1} \vee I_l < I_{l+1} \end{pmatrix} : r\left[I_{(\ldots)}\right] \mathrel{+}= f_{(\ldots)} p[I(\#\{l, j_l = 1\}, \ldots, \#\{l, j_l = n\})] \prod_l q_{j_l}[I_l]$$

in order to minimize the number of times the same $q_{j_l}\left[I_l\right]$ are multiplied, the iteration is done in the order given by the following recursive function because the outer $\prod_l q_{j_l}[I_l]$ can be reused inside:

```
iterate( J: array(int), I: array(int), q_product: float) {
  r[...] += f(...) * p[...] * q_product
  for next_I in last(I)..ms_coeff(d[last( J)]+1, n[last( J)])-1 {
    iterate(append( J, last( J)), append(I, next_I, q_product * q[last( j)][next_I]))
  }
  for next_J in last( J)+1..n {
    for next_I in 0..ms_coeff(d[next_J]+1, n[next_J])-1 {
      iterate(append( J, next_J), append(I, next_I), q_product * q[next_J][next_I])
    }
  }
}
for first_J in 0..n {
  for first_I in 0..ms_coeff(d[first_J]+1,n[first_J])-1 {
    iterate(first_J,first_I, q[first_J][first_I])
  }
}
```

the numbers that have to be calculated in every step can be calculated from previous values if we know how to change them for the following cases:

1. the last j_l and I_l are copied onto the end
2. the last I_l is incremented
3. some j_l that wasn't there before with $I_l = 0$ is appended
4. the last j_l is incremented keeping the last $I_l = 0$

the simplest is the index for p: $I_p := I_n\left(\#\{l, j_l = 1\}, \ldots, \#\{l, j_l = n\}\right)$

$$I_p = \sum_{j=0}^{n-1} \left(\!\!\binom{\#\{l, j_l > j\}}{n-j}\!\!\right) = \sum_{j=0}^{n-1} \sum_{l, j_l > j} \left(\!\!\binom{D+1-l}{n-j-1}\!\!\right) = \sum_{l=1}^{D} \left(\!\!\binom{n}{D+1-l}\!\!\right) - \left(\!\!\binom{n-j_l}{D+1-l}\!\!\right) = \left(\!\!\binom{n+1}{D}\!\!\right) - 1 - \sum_{l=1}^{D} \left(\!\!\binom{n-j_l}{D+1-l}\!\!\right)$$

this form is better especially for $n >> d$. for cases (1.) and (3.) not much can be gained over just recalculating I_p, for case (2.), I_p does not change but for case (4.) it changes to:

$$I_p(j_1, \ldots, j_D + 1) = \left(\!\!\binom{n+1}{D}\!\!\right) - 1 - \left(\sum_{l=1}^{D} \left(\!\!\binom{n-j_l}{D+1-l}\!\!\right)\right) - \left(\!\!\binom{n-j_D-1}{1}\!\!\right) + \left(\!\!\binom{n-j_D}{1}\!\!\right) = I_p(j_1, \ldots, j_D) + 1$$

overall, we only have to recalculate I_p once per call to `iterate` when copying the last j_l, and then incrementing it whenever incrementing the (new) last j_l.

the second number that has to be calculated is the repetition factor:

$$f_{(\ldots)} = \prod_{j=1}^{n} \begin{pmatrix} \#\{l, j_l = j\} \\ \#\left\{l, j_l = j, I_l = 0\right\}, \ldots, \#\left\{l, j_l = j, I_l = \left(\!\!\binom{d_j + 1}{n_j}\!\!\right) - 1\right\} \end{pmatrix}$$

in case (3.), it does not change:

$$f(j_1, \ldots, j_D, j_+; I_1, \ldots, I_D, 0) = f(j_1, \ldots, j_D; I_1, \ldots, I_D) \cdot \binom{1}{1, 0, \ldots, 0} = f(j_1, \ldots, j_D; I_1, \ldots, I_D)$$

therefore we also don't need to calculate case (4.). in case (2.), $f_{(\dots)}$ changes according to:

$$
\begin{aligned}
f(j_1,\dots,j_D;I_1,\dots,I_{D-1},I_D+1) &= f(j_1,\dots,j_D;I_1,\dots,I_D)\cdot\frac{\#\{l,j_l=j_D,I_l=I_D\}!}{(\#\{l,j_l=j_D,I_l=I_D\}-1)!\cdot 1}\\
&= f(j_1,\dots,j_D;I_1,\dots,I_D)\cdot\#\{l,j_l=j_D,I_l=I_D\}
\end{aligned}
$$

lastly, in case (1.) $f_{(\dots)}$ changes according to:

$$
\begin{aligned}
f(j_1,\dots,j_D,j_D;I_1,\dots,I_D,I_D) &= f(j_1,\dots,j_D;I_1,\dots,I_D)\cdot\binom{\#\{l,j_l=j_D\}+1}{\dots,(\#\{l,j_l=j_D,I_l=I_D\}+1),\dots}\Big/\binom{\#\{l,j_l=j_D\}}{\dots}\\
&= f(j_1,\dots,j_D;I_1,\dots,I_D)\cdot\frac{\#\{l,j_l=j_D\}+1}{\#\{l,j_l=j_D,I_l=I_D\}+1}
\end{aligned}
$$

the last number we need is

$$
\begin{aligned}
I_{(\dots)} &= I_m\left(\left(\sum_{i,j,k:\ X_j(i)=1}E(I_{j,k})_i\right),\dots,\left(\sum_{i,j,k:\ X_j(i)=m}E(I_{j,k})_i\right)\right)\\
&= \sum_{X_{\min}=0}^{m-1}\left(\!\!\binom{\sum_{X=X_{\min}+1}^{m}\sum_{i,l:X_{j_l}(i)=X}E(I_l)_i}{m-X_{\min}}\!\!\right) = \sum_{X_{\min}=0}^{m-1}\left(\!\!\binom{\sum_{i,l:X_{j_l}(i)>X_{\min}}E(I_l)_i}{m-X_{\min}}\!\!\right)
\end{aligned}
$$

in case (3.) the change is simple:

$$
\begin{aligned}
I(j_1,\dots,j_D,j_+;I_1,\dots,I_D,0) &= I(j_1,\dots,j_D;I_1,\dots,I_D)+\sum_{X_{\min}=0}^{<X_{j_+}(0)}+\left(\!\!\binom{\left(\sum_{i,l:X_{j_l}(i)>X_{\min}}E(I_l)_i\right)+1}{m-X_{\min}}\!\!\right)-\left(\!\!\binom{\sum_{i,l:X_{j_l}(i)>X_{\min}}E(I_l)_i}{m-X_{\min}}\!\!\right)\\
&= I(j_1,\dots,j_D;I_1,\dots,I_D)+\sum_{X_{\min}=0}^{<X_{j_+}(0)}\left(\!\!\binom{\left(\sum_{i,l:X_{j_l}(i)>X_{\min}}E(I_l)_i\right)+1}{m-X_{\min}-1}\!\!\right)
\end{aligned}
$$

we can also write:

$$
\begin{aligned}
I_{(\dots)} &= \sum_{X_{\min}=0}^{m-1}\left(\!\!\binom{\sum_{i,l:X_{j_l}(i)>X_{\min}}E(I_l)_i}{m-X_{\min}}\!\!\right) = \sum_{X_{\min}=0}^{m-1}\sum_{\lambda,S_\lambda>X_{\min}}\left(\!\!\binom{L+1-\lambda}{m-S_{\min}-1}\!\!\right) = \sum_{\lambda,X=S_\lambda}\left(\!\!\binom{m}{L+1-\lambda}\!\!\right)-\left(\!\!\binom{m-X}{L+1-\lambda}\!\!\right)\\
&= \left(\!\!\binom{m+1}{L}\!\!\right)-1-\sum_{\lambda,X=S_\lambda}\left(\!\!\binom{m-X}{L+1-\lambda}\!\!\right)
\end{aligned}
$$

where

$$
S_\lambda = \max_X\left(\sum_{i,l,X_{j_l}(i)<X}E(I_{j_l})_i<\lambda\right)
$$

S is the sorted 1-indexed array containing each X $\sum_{i,l,X_{j_l}(i)=X}E(I_{j_l})_i$ times and L is it's length $\sum_{i,l}E(I_{j_l})_i$. the possible changes then have the effects on S:

1. each $X_{j_D}(\star)$ is sorted in $E(I_D)_\star$ times
2. let X be the sorted list where each $X_{j_D}(\star)$ is contained $E(I_D)_\star$ times. remove each element of X from S, then sort in each element of the list:

- if $X[0] = \mathrm{last}(X_{j_D}(\star))$ then $X_{j_D}(0)$ repeated $\mathrm{len}(X) + 1$ times
- else: let $\iota := \max_i(X[i] \neq \mathrm{last}(X_{j_D}(\star)))$ and $X_{j_D}(\kappa) = X[\iota]$ in $(X[0], \ldots, X[\iota-1], X_{j_D}(\kappa+1), \ldots, X_{j_D}(\kappa+1))$ with the same length as X

3. $X_{j_l}(0)$ is sorted into S

so the operations we have to be able to do are:

- insert each element of some list X into S
- remove each element of some list X from S, then $\mathrm{len}(X)$ times insert some X_+.

in the first case, $I_{(\ldots)}$ changes to:

$$I_{(\ldots)} = \left(\!\!\binom{m+1}{\mathrm{len}(S)+\mathrm{len}(X)}\!\!\right) - 1 - \sum_{\lambda=1}^{\mathrm{len}(S)+\mathrm{len}(X)} \left(\!\!\binom{m-\mathrm{sort}(S\oplus X)[\lambda-1]}{\mathrm{len}(S)+\mathrm{len}(X)+1-\lambda}\!\!\right)$$

$$= \left(\!\!\binom{m+1}{L_+}\!\!\right) - 1 - \sum_{\lambda=1}^{\mathrm{len}(S)} \left(\!\!\binom{m-S[\lambda-1]}{L_+ + 1 - \lambda - \#\{i, X[i] < S[\lambda-1]\}}\!\!\right) - \sum_{\lambda=1}^{\mathrm{len}(X)} \left(\!\!\binom{m-X[\lambda-1]}{L_+ + 1 - \lambda - \#\{i, S[i] \leq X[\lambda-1]\}}\!\!\right)$$

$$I_{(\ldots)} - I_{(\ldots),\mathrm{prev.}} = \left(\!\!\binom{m+1}{L_+}\!\!\right) - \left(\!\!\binom{m+1}{\mathrm{len}(S)}\!\!\right) - \sum_{\lambda=1}^{\mathrm{len}(X)} \left(\!\!\binom{m-X[\lambda-1]}{L_+ + 1 - \lambda - \#\{i, S[i] \leq X[\lambda-1]\}}\!\!\right)$$

$$+ \sum_{\lambda=1}^{\mathrm{len}(S)} \left(\!\!\binom{m-S[\lambda-1]}{\mathrm{len}(S)+1-\lambda}\!\!\right) - \left(\!\!\binom{m-S[\lambda-1]}{\mathrm{len}(S)+1-\lambda+\#\{i, X[i] \geq S[\lambda-1]\}}\!\!\right)$$

$$= \left(\!\!\binom{m+1}{L_+}\!\!\right) - \left(\!\!\binom{m+1}{\mathrm{len}(S)}\!\!\right) - \sum_{\lambda=1}^{\mathrm{len}(X)} \left(\!\!\binom{m-X[\lambda-1]}{L_+ + 1 - \lambda - \#\{i, S[i] \leq X[\lambda-1]\}}\!\!\right) - \sum_{\lambda=1}^{\mathrm{len}(S)} \sum_{i, X[i] \geq S[\lambda-1]} \left(\!\!\binom{m-S[\lambda-1]-1}{\mathrm{len}(S)+1-\lambda+\mathrm{len}(X)-i}\!\!\right)$$

$$= \left(\!\!\binom{m+1}{L_+}\!\!\right) - \left(\!\!\binom{m+1}{\mathrm{len}(S)}\!\!\right) - \sum_{\lambda=1}^{\mathrm{len}(X)} \left(\left(\!\!\binom{m-X[\lambda-1]}{L_+ + 1 - \lambda - \#\{i, S[i] \leq X[\lambda-1]\}}\!\!\right) + \sum_{\mu, S[\mu-1] \leq X[\lambda-1]} \left(\!\!\binom{m-S[\mu-1]-1}{L_+ + 2 - \mu - \lambda}\!\!\right)\right)$$