Algorithm : reduce tensor product

input : formula "ijk = -jik = ikj"
and ireps of each index $\begin{cases} i = 1e \\ j = 1e \\ k = 1e \end{cases}$

output : change of basis Q
Q shape is
$(\#\text{actual deg. of freedom}, d_1, \dots, \underbrace{d_n}_{\text{rank}})$

Current implementation :

1. germinate formula : create a group from the input
2. create basis P from perm. symmetry
3. create basis Q from Clebsch-Gordan coeff.
4. find projectors into the intersection of P and Q (see next page)
5. choose a basis using orthonormalize
6. create the change of basis

1 is fast and memory cheap

2 is same size as the output

3 is memory expensive

4 is memory expensive and slow

5 is ok

6 is ok

2 and 6 create large objects but 3 is even larger

④ details

$$P \in \mathbb{R}^{p \times d} \qquad Q \in \mathbb{R}^{q \times d} \qquad \begin{matrix} p \leq d \\ q \leq d \end{matrix}$$

look for $(x, y) \in \mathbb{R}^p \times \mathbb{R}^q$ such that

$$x^T P = y^T Q$$

which can be rewritten in the form:

$$\begin{pmatrix} x^T & y^T \end{pmatrix} \begin{pmatrix} P \\ -Q \end{pmatrix} = 0 \qquad (*)$$

solutions of $(*)$ are also solutions of:

$$\begin{pmatrix} x^T & y^T \end{pmatrix} \underbrace{\begin{pmatrix} P \\ -Q \end{pmatrix} \begin{pmatrix} P^T & -Q^T \end{pmatrix}}_{\begin{pmatrix} PP^T & -PQ^T \\ -(PQ^T)^T & QQ^T \end{pmatrix}} = 0$$