16.6.2023 L sequence - Algorithm

INPUT:

Two polynomials F and G defined as sums of their holynomials (= homogeneous polynomials):

$$F = F_m + F_{m+1} + F_{m+2} + \cdots + F_M$$
$$G = G_n + G_{n+1} + G_{n+2} + \cdots + G_N$$

(F_i, G_i is homogeneous of degree i) While this is an algorithm for polynomials, all the individual steps are done on holynomials. Its very important to watch the degrees.
We treat F and G as lists of holynomials, i.e. $F[k] = F_k$.

ALGORITHM:

Algorithm is a while cycle.

Step(0)
$i = 1$
while (not stopping condition) **do**
  Step(i)
  $i++$
end while

Each step adds a new element to the following lists:

- $L = \mathcal{L}$-sequence (list of the holynomials ${}_iL$)
- $A = \mathcal{A}$-sequence (list of integers $a_i = \deg {}_iL$)
- KA = list of lists of holynomials, of which the first coordinate of $\ker \psi_i$ can be constructed
- KB = list of lists of holynomials, of which the second coordinate of $\ker \psi_i$ can be constructed
- U = list of holynomials ${}_i\Upsilon$
- Z = list of holynomials $Z_i = \frac{{}_i\Upsilon}{{}_iL}$
- D = list of holynomials $D_i = \frac{{}_iL}{{}_{i+1}L}$

METHODS

- $A = \gcd(U, V)$ - greatest common divisor of two holynomials
- $A = U/V$ - quotient of two holynomials
- $A, B = \text{OneSol}(U, V, W)$ - returns one solution (pair A, B) of the homogeneous equation

$$A_{p-i}U_i + B_{p-j}V_j = W_p$$

  where $\gcd(U_i, V_j) = 1$ and $i + j < p$

Remark: both KA and KB will be lists of lists of structure:

$$KA = \big[[\circ], [\circ, \circ], [\circ, \circ, \circ], [\circ, \circ, \circ, \circ], \cdots\big]$$

where each circle represents one holynomial.

STEP 0:

$L[0] = \gcd(F_m, G_n)$
$A[0] = \deg({}_0L)$
$U[0] = 0$
$Z[0] = 0$
$D[0] = 0$

$ka = G_n / {}_0L$
$kb = F_m / {}_0L$
$KA[0] = [ka]$ (as list of one holynomial)
$KB[0] = [kb]$ (as list of one holynomial)

STEP k:

$U[k] = \sum_{i=0}^{k-1} \Big(KB[k-1][i] \cdot G[n+i+1] - KA[k-1][i] \cdot F[m+i+1] \Big)$
$L[k] = \gcd(L[k-1], U[k])$
$A[k] = \deg(L[k])$
$Z[k] = U[k]/L[k]$
$D[k] = L[k-1]/L[k]$

Now we need the list C of k holynomial coefficients.
$C[0] = D[k]$
$v = -Z[k]$

for $(i = 0, \cdots, k-1)$ **do**:
$C[i+1], v = \text{OneSol}(Z[k-i-1], -D[k-i-1], v)$
end for
$a, b = \text{OneSol}(KB[0][0], KA[0][0], v)$

Now we can construct the new kernel (As a lists ka and kb of $k+1$ holynomials):
$ka[0] = a$
$kb[0] = b$
for $(i = 1, \cdots, k)$ **do**
$ka[i] = \sum_{j=0}^{k-i} C[j] \cdot KA[k-1-j][i-1]$
$kb[i] = \sum_{j=0}^{k-i} C[j] \cdot KB[k-1-j][i-1]$
end for

$KA[k] = ka$ (as list of $k+1$ holynomials)
$KB[k] = kb$ (as list of $k+1$ holynomials)

STOPPING CONDITIONS:

Algorithm stops if one of the conditions is true:

- We reach step $k+1$ where $A[k+1] = 0$. Then we have found the whole sequences $L = [{}_0L, \cdots {}_kL]$ and $A = [a_0, \cdots, a_k]$.
- We reach the limit of Bezout theorem, which means that $\sum a_i > \deg(F) \cdot \deg(G)$. Then F and G have a common component passing through 0 and the intersection multiplicity is ∞.

OUTPUT:

Output consists of

- $\mathcal{L}$-sequence (list $L = [{}_0L, {}_1L, \cdots {}_kL]$)
- $\mathcal{A}$-sequence (list $A = [a_0, a_1, \cdots, a_k]$)

Then the intersection multiplicity of F and G at $O = (0, 0)$ equals

$$I_O(F, G) = mn + \sum a_i \tag{1}$$