Multivariate Rational Function Interpolation

Input: Modular black box $B : \mathbb{Z}_p^n \longrightarrow \mathbb{Z}_p$ for rational function $\frac{ff(x_1,\ldots,x_n)}{gg(x_1,\ldots,x_n)}$ over $\mathbb{Z}_p$, where $ff, gg \in \mathbb{Z}_p[x_1,\ldots,x_n]$, $\gcd(ff,gg) = 1$, gg is monic, prime p,number of variables n, and list of variables $vars$.

Output: $ff(x_1,\ldots,x_n), gg(x_1,\ldots,x_n)$ or **FAIL**.

$Primes \leftarrow [2, 3, \ldots, p_n] \in \mathbb{Z}_p^n$

Pick random vector $\beta = [\beta_2, \ldots, \beta_n] \in \mathbb{Z}_p^{n-1}$

$num \leftarrow [\ ]$, $den \leftarrow [\ ]$

$num\_points\_mqrfr \leftarrow 0$

$numerator\_done \leftarrow$ false

$denominator\_done \leftarrow$ false

$T\_old \leftarrow 1$

$T \leftarrow 4$

$\sigma^0 \leftarrow [1, \ldots, 1] \in \mathbb{Z}_p^n$

$(f_0, g_0) \leftarrow \mathrm{NDSA}(B, \sigma^0, \beta, p, T)$

num.append($f_0(\sigma_1^0) \bmod p$)

den.append($g_0(\sigma_1^0) \bmod p$)

$num\_points\_mqrfr \leftarrow \deg(f) + \deg(g) + 2$

while true **do**

    for $j \leftarrow T\_old$ to $2T - 1$ **do**

        $\sigma^j \leftarrow [2^j, 3^j, \ldots, p_n^j] \mod p$

        $(f_j, g_j) \leftarrow \mathrm{NDSA}(B, \sigma^j, \beta, p, num\_points\_mqrfr)$

        if *not* $numerator\_done$ **then**

            num.append($f_j(\sigma_1^j) \bmod p$)

        end if

        if *not* $denominator\_done$ **then**

            den.append($g_j(\sigma_1^j) \bmod p$)

        end if

    end for

    Construct minimum characteristic polynomials using Berlekamp-Massey algorithm

    if *not* $numerator\_done$ **then**

        $\Lambda_{num}(z) \leftarrow \mathrm{Berlekamp\_Massey}(num, p, z) \in \mathbb{Z}_p[z]$

        $s \leftarrow \deg(\Lambda_{num}(z))$

        Let $roots_{num}$ be the distinct roots of $\Lambda_{num}(z) \in \mathbb{Z}_p[z]$.

    end if

    if *not* $denominator\_done$ **then**

        $\Lambda_{den}(z) \leftarrow \mathrm{Berlekamp\_Massey}(den, p, z) \in \mathbb{Z}_p[z]$

        $d \leftarrow \deg(\Lambda_{den}(z))$

        Let $roots_{den}$ be the distinct roots of $\Lambda_{den}(z) \in \mathbb{Z}_p[z]$.

    end if

    if $s = |roots_{num}|$ and $s < T$ **then**

        $numerator\_done \leftarrow$ true

    end if

if $d = |roots_{den}|$ and $d < T$ **then**
$denominator\_done \leftarrow$ true
end if
if $numerator\_done \wedge denominator\_done$ **then**
break
end if
$T\_old \leftarrow 2T$
$T \leftarrow 2T$
end while
Recover monomials from roots using trial division
$N \leftarrow$ get_monomial$(roots_{num}, Primes, n, vars)$
$D \leftarrow$ get_monomial$(roots_{den}, Primes, n, vars)$
if N = FAIL or D = FAIL **then**
return FAIL
end if
Recover coefficients via Zippel Vandermonde solver
$A \leftarrow$ Zippel_Vandermonde_solver$(num, s, roots_{num}, \Lambda_{num}(z), p)$
$B \leftarrow$ Zippel_Vandermonde_solver$(den, d, roots_{den}, \Lambda_{den}(z), p)$
$ff \leftarrow \sum_{m=1}^{s} A_m N_m, \quad gg \leftarrow \sum_{m=1}^{d} B_m D_m$
Let μ be the leading coefficient of gg in grlex order where $x_1 > \cdots > x_n$.
$ff \leftarrow \mu^{-1} ff$ mod p, $gg \leftarrow \mu^{-1} gg$ mod p.
return ff, gg.

Numerator Denominator Separation Algorithm (NDSA)

Input: Modular black box B for rational function $\frac{ff(x_1,\ldots,x_n)}{gg(x_1,\ldots,x_n)}$ over $\mathbb{Z}_p$, prime p, where $ff, gg \in K[x_1,\ldots,x_n]$, $\gcd(ff,gg)=1$, $\sigma \in \mathbb{Z}_p^n$, $\beta \in \mathbb{Z}_p^{n-1}$, $num\_points \in \mathbb{N}$.

Output: $f(x) = \frac{ff(x,\beta_2(x-\sigma_1)+\sigma_2,\ldots,\beta_n(x-\sigma_1)+\sigma_n)}{c}$, $g(x) = \frac{gg(x,\beta_2(x-\sigma_1)+\sigma_2,\ldots,\beta_n(x-\sigma_1)+\sigma_n)}{c}$, for some scalar $c \in \mathbb{Z}_p^n$.

```
t ← num_points
while true do
    Pick random vector α = [α_1, ..., α_t] ∈ Z_p^t
    m(x) ← ∏_{k=1}^{t} (x − α_k) ∈ Z_p[x]
    Φ ← [φ(α_1), ..., φ(α_t)] ∈ Z_p^{t×n} such that: φ(α_k) ← [α_k, β_2(α_k − σ_1) + σ_2, ..., β_n(α_k − σ_1) + σ_n] mod p ∀ 1 ≤ k ≤ t
    Y ← [B(Φ(α_1), p), ..., B(Φ(α_t), p)] ∈ Z_p^t
    u(x) ← Interpolate(α, Y, x) mod p
    (f(x), g(x), deg_q) ← MQRFR(m, u) mod p
    if deg_q > 1 then
        break
    else
        t ← 2t
    end if
end while
return f, g
```