Implementing g_0 for sl_3 (Quantum algebra)

```
(*Orig:  { X⁺1,  X⁺2,   X⁺3, H⁺1,   H⁺2,  H⁻2,  H⁻1,   X⁻3,  X⁻2,   X⁻1};
          {X,    Y,     Z,   A,     B,    b,    a,     z,    y,     x};
A,B central. e^(2A-B)=s, e^(-A+2B)=t*)
PBWBasis = {X, Y, Z, b, a, z, y, x};
PBWRule = {X → 1, Y → 2, Z → 3, b → 4, a → 5, z → 6, y → 7, x → 8};
Br[U@X, U@Y] = 0;
Br[U@X, U@Z] = 0;
Br[U@X, U@b] = U@X;
Br[U@X, U@a] = -2 U@X;
Br[U@X, U@z] = 0;
Br[U@X, U@y] = 0;
Br[U@X, U@x] = (s - 1) U[];
Br[U@Y, U@Z] = 0;
Br[U@Y, U@b] = -2 U@Y;
Br[U@Y, U@a] = U@Y;
Br[U@Y, U@z] = 0;
Br[U@Y, U@y] = (t - 1) U[];
Br[U@Y, U@x] = 0;
Br[U@Z, U@b] = -U@Z;
Br[U@Z, U@a] = -U@Z;
Br[U@Z, U@z] = (s t - 1) U[];
Br[U@Z, U@y] = -U@X;
Br[U@Z, U@x] = s U@Y;
Br[U@b, U@a] = 0;
Br[U@b, U@z] = -U@z;
Br[U@b, U@y] = -2 U@y;
Br[U@b, U@x] = U@x;
Br[U@a, U@z] = -U@z;
Br[U@a, U@y] = U@y;
Br[U@a, U@x] = -2 U@x;
Br[U@z, U@y] = 0;
Br[U@z, U@x] = 0;
Br[U@y, U@x] = -U@z;
```

```
x_ ≤ y_ := OrderedQ[{x, y} /. PBWRule];
x_ < y_ := ! OrderedQ[{y, x} /. PBWRule];
Simp[ℰ_] := Collect[ℰ, _U, Expand];
```

```
Ui_[ℰ_] := ℰ /. {A → Ai, s → si, B → Bi, t → ti, u_U ⧴ Replace[u, x_ ⧴ xi, 1]};
Br[U[(x_)i_], U[(y_)i_]] := Br[U[xi], U[yi]] = Ui[Br[U@x, U@y]];
Br[U[(x_)i_], U[(y_)j_]] /; i =!= j := 0;
Br[x_, x_] = 0;
Br[U[y_], U[x_]] := Br[U[y], U[x]] = Simp[-Br[U[x], U[y]]];
Br[x_, y_] := x ** y - y ** x;
```

```
Unprotect[NonCommutativeMultiply];
NonCommutativeMultiply[x_] := x;
0 ** _ = _ ** 0 = 0;
x_ ** U[] := x; U[] ** x_ := x;
(a_ * x_U) ** (b_ * y_U) := If[a b === 0, 0, Simp[a b (x ** y)]];
(a_ * x_U) ** y_ := Simp[a (x ** y)]; x_ ** (a_ * y_U) := Simp[a (x ** y)];
(x_Plus) ** y_ := (# ** y) & /@ x; x_ ** (y_Plus) := (x ** #) & /@ y;
```

```
U[xx___, x_] ** U[y_, yy___] := If[x ≤ y, U[xx, x, y, yy], U@xx ** (U@y ** U@x + Br[U@x, U@y]) ** U@yy];
```

```
UU[l___, x_^n_, r___] := UU[l, Sequence @@ Table[x, {n}], r];
UU[l___, 1, r___] := UU[l, r];
UU[] = U[];
UU[l_, r___] := U[l] ** UU[r];
```

```
UProducts[{}, 0] = {UU[]};
UProducts[{}, n_Integer] /; n > 0 = {};
UProducts[{x_, xs___}, n_Integer] :=
  Sort@Flatten@Table[UU[x^k] ** u, {k, 0, n}, {u, UProducts[{xs}, n - k]}];
UProducts[xs_List, k_Integer, n_Integer] := UProducts[Flatten@Table[x_j, {x, xs}, {j, k}], n];
UProducts[any__, {n_}] := Flatten@Table[UProducts[any, k], {k, 0, n}];
```

```
Δ[i_, j_, k_][ℰ_] := Simp[ℰ /. {
    w_. U[] :→ (w /. {A_i → A_j + A_k, B_i → B_j + B_k, s_i → s_j s_k, t_i → t_j t_k}) U[],
    w_. v_U :→ (w /. {A_i → A_j + A_k, B_i → B_j + B_k, s_i → s_j s_k, t_i → t_j t_k}) NonCommutativeMultiply @@ (v /. {
          X_i → U@X_j s_k + U@X_k,
          Y_i → U@Y_j t_k + U@Y_k,
          Z_i → U@Z_j s_k t_k + U@Z_k + s_k UU[X_j, Y_k] ×
            b_i → U@b_j + U@b_k,
          a_i → U@a_j + U@a_k,
          z_i → U@z_j + U@z_k,
          y_i → U@y_j + U@y_k,
          x_i → U@x_j + U@x_k,
          u_l_ :→ U@u_l

        })
  }]
```

```
S[i_][ℰ_] := Simp[ℰ /. {z_. x_U :→ (z /. {A_i → -A_i, B_i → -B_i, s_i → s_i^-1, t_i → t_i^-1}) S[i][x]}];
S[i_][U[]] = U[];
S[i_][U[w_j_, more___]] /; i ≠ j := U[w_j] ** S[i][U[more]];
(*Careful! if mathematica cannot decide i==j or not then we get an error*)
S[i_][U[X_i_, more___]] := S[i][U[more]] ** (-s_i^-1 U@X_i);
S[i_][U[Y_i_, more___]] := S[i][U[more]] ** (-t_i^-1 U@Y_i);
S[i_][U[Z_i_, more___]] := S[i][U[more]] ** (-t_i^-1 s_i^-1 U@Z_i + U[X, Y] t_i^-1 s_i^-1);
S[i_][U[b_i_, more___]] := S[i][U[more]] ** (-U@b_i);
S[i_][U[a_i_, more___]] := S[i][U[more]] ** (-U@a_i);
S[i_][U[z_i_, more___]] := S[i][U[more]] ** (-U@z_i);
S[i_][U[y_i_, more___]] := S[i][U[more]] ** (-U@y_i);
S[i_][U[x_i_, more___]] := S[i][U[more]] ** (-U@x_i);
```

```
σ[i_, j_][ℰ_] := ℰ /. {i → k, j → i} /. k → j;
```

```
mul[i_, j_][ℰ_] :=
  Simp[ℰ /. x_U :→ DeleteCases[x, _j] ** U @@ Cases[x, y_j :→ y_i] /. {A_j → A_i, B_j → B_i, s_j → s_i, t_j → t_i}];
mul[i_, j_, k_][ℰ_] := ℰ // mul[i, j] // σ[i, k];
```

```
CoUnit[i_][ℰ_] := Simp[ℰ /. {z_. x_U ⧴ (z /. {t → 1, b → 0, b_i → 0, t_i → 1}) CoUnit[i][x]}];
CoUnit[i_][U[]] = U[];
CoUnit[i_][U[y_j_, more___]] /; i ≠ j := U[y_j] ** CoUnit[i][U[more]];
CoUnit[i_][U[y_i_, more___]] := 0;
```

```
ToDegree[n_][ℰ_] :=
 (Simp[ℰ] /. {A_i_ ⧴ ℏ A_i, B_i_ ⧴ ℏ B_i, s_i_ ⧴ e^(ℏ A_i), t_i_ ⧴ e^(ℏ B_i), A ⧴ ℏ A, B ⧴ ℏ B, s ⧴ e^(ℏ A), t ⧴ e^(ℏ B),
     w_U ⧴ ℏ^(Count[w,X|X_]+Count[w,Y|Y_]+Count[w,Z|Z_]) w} /. v_. w_U ⧴ Normal[Series[v, {ℏ, 0, n}]] * w) /. ℏ → 1
```

Check anti-symmetry and Jacobi and associativity.

```
bas = UProducts[PBWBasis, 1, {3}];
Table[Br[c, d] + Br[d, c] // Simp, {c, bas}, {d, bas}] // Flatten // Union
```

{0}

```
bas = UProducts[Take[PBWBasis, 5], 1, {3}];
Table[
   {c, d, e} = cde;
   Simp[Br[Br[c, d], e] + Br[Br[d, e], c] + Br[Br[e, c], d]],
   {cde, Subsets[bas, {3}]}
  ] // Flatten // Union
```

{0}

```
bas = UProducts[Take[PBWBasis, 4], 1, {3}];
Table[
   {c, d, e} = cde;
   Simp[c ** (d ** e) - (c ** d) ** e],
   {cde, Subsets[bas, {3}]}
  ] // Flatten // Union
```

{0}

Check co-associativity

```
Table[c, {c, bas}]
```

$\{U[], U[a_1], U[b_1], U[x_1], U[X_1], U[y_1], U[Y_1], U[z_1], U[Z_1]\}$

```
bas = UProducts[PBWBasis, 1, {1}];
Table[Δ[1, 4, 3][c] // Simp, {c, bas}]
```

$\{U[], U[a_3] + U[a_4], U[b_3] + U[b_4], U[x_3] + U[x_4], s_4 U[X_3] + U[X_4],$
$U[y_3] + U[y_4], t_4 U[Y_3] + U[Y_4], U[z_3] + U[z_4], s_4 t_4 U[Z_3] + U[Z_4]\}$

```
bas = UProducts[PBWBasis, 1, {1}];
Table[Δ[4, 1, 2][Δ[1, 4, 3][c]] -
    Δ[4, 2, 3][Δ[1, 1, 4][c]] // Simp, {c, bas}] // Union
```

{0}

S-Delta-axiom

```
bas = UProducts[PBWBasis, 1, {1}];
Table[mul[2, 3, 1][S[2][Δ[1, 2, 3][c]]] // Simp, {c, bas}] // Union
```

{0, U[]}

```
Sum[F[i, j], {i, 0, 3}, {j, 0, 3 - i}]
```

F[0, 0] + F[0, 1] + F[0, 2] + F[0, 3] + F[1, 0] + F[1, 1] + F[1, 2] + F[2, 0] + F[2, 1] + F[3, 0]

```
R[i_, j_, d_] := Sum[((A_i^n4 B_i^n5) / (n1 ! n2 ! n3 ! n4 ! n5 !)) UU[X_i^n1, Y_i^n2, Z_i^n3, b_j^n5, a_j^n4, z_j^n3, y_j^n2, x_j^n1], {n1, 0, d},
   {n2, 0, d - n1}, {n3, 0, d - n1 - n2}, {n4, 0, d - n1 - n2 - n3}, {n5, 0, d - n1 - n2 - n3 - n4}] // ToDegree[d]

R[1, 2, 2]
```

$U[] + A_1 U[a_2] + B_1 U[b_2] + \frac{1}{2} A_1^2 U[a_2, a_2] + A_1 B_1 U[b_2, a_2] + \frac{1}{2} B_1^2 U[b_2, b_2] + U[X_1, x_2] + U[Y_1, y_2] + U[Z_1, z_2] + A_1 U[X_1, a_2, x_2] + B_1 U[X_1, b_2, x_2] + A_1 U[Y_1, a_2, y_2] + B_1 U[Y_1, b_2, y_2] + A_1 U[Z_1, a_2, z_2] + B_1 U[Z_1, b_2, z_2] + \frac{1}{2} U[X_1, X_1, x_2, x_2] + U[X_1, Y_1, y_2, x_2] + U[X_1, Z_1, z_2, x_2] + \frac{1}{2} U[Y_1, Y_1, y_2, y_2] + U[Y_1, Z_1, z_2, y_2] + \frac{1}{2} U[Z_1, Z_1, z_2, z_2]$

```
(*Reidemeister 3*)
R3[d_] := ToDegree[d][
  (ToDegree[d][R[1, 2, d] ** R[1, 3, d]] ** R[2, 3, d]) - (ToDegree[d][R[2, 3, d] ** R[1, 3, d]] ** R[1, 2, d])]

R3[2] // ToDegree[1]
```

0

```
R3[3] // ToDegree[2]
```