```
Unprotect[NonCommutativeMultiply];
Attributes[NonCommutativeMultiply] = {};
(NCM = NonCommutativeMultiply)[x_] := x;
NCM[x_, y_, z__] := (x ** y) ** z;
0 ** _ = _ ** 0 = 0;
(x_Plus) ** y_ := (# ** y) & /@ x;
x_ ** (y_Plus) := (x ** #) & /@ y;
B[x_, x_] = 0; B[x_, y_] := x ** y - y ** x;
B[x_, y_, e_] := B[x, y, e] = B[x, y];
DeclareAlgebra[U_Symbol, opts__Rule] :=
 Module[{gp, sr, g, cp, M, CE, k = 0,
   gs = Generators /. {opts},
   cs = Centrals /. {opts} /. Centrals → {} },
  (#U = U@#) & /@ gs;
  gp = Alternatives @@ gs; gp = gp | gp_; (* gens *)
  sr = Flatten@Table[{g → ++k, gi_ → {i, k}}, {g, gs}];
  (* sorting → *)
  cp = Alternatives @@ cs; (* cents *)
  SetAttributes[M, HoldRest]; M[0, _] = 0;
  M[a_, x_] := a x;
  CE[ℰ_] := Collect[ℰ, _U, Expand] /. $trim;
  Ui_[ℰ_] := ℰ /. {t : cp ⧴ ti, u_U ⧴ (#i &) /@ u};
  Ui_[NCM[]] = U@{} = 1U = U[];
  B[U@(x_)i_, U@(y_)i_] := Ui@B[U@x, U@y];
  B[U@(x_)i_, U@(y_)j_] /; i =!= j := 0;
  B[U@y_, U@x_] := CE[-B[U@x, U@y]];
  x_ ** (c_. 1U) := CE[c x]; (c_. 1U) ** x_ := CE[c x];
  (a_. U[xx___, x_]) ** (b_. U[y_, yy___]) :=
   If[OrderedQ[{x, y} /. sr],
    CE@M[a b /. $trim, U[xx, x, y, yy]],
    U@xx **
     CE@M[a b /. $trim, U@y ** U@x + B[U@x, U@y, $E]] **
     U@yy ];
  U@{c_. * (l : gp)^n_, r___} /; FreeQ[c, gp] :=
   CE[c U@Table[l, {n}] ** U@{r}];
  U@{c_. * l : gp, r___} := CE[c U[l] ** U@{r}];
  U@{c_, r___} /; FreeQ[c, gp] := CE[c U@{r}];
  U@{l_Plus, r___} := CE[U@{#, r} & /@ l];
  U@{l_, r___} := U@{Expand[l], r};
  U[ℰ_NonCommutativeMultiply] := U /@ ℰ;
  𝕆U[specs___, poly_] := Module[{sp, null, vs, us},
    sp = Replace[{specs}, l_List ⧴ lnull, {1}];
    vs = Join @@ (First /@ sp);
    us = Join @@ (sp /. l_s_ ⧴ (l /. x_i_ ⧴ xs));
    CE[Total[
       CoefficientRules[poly, vs] /. (p_ → c_) ⧴ c U@(us^p)
      ]] /. x_null ⧴ x];
  𝕆U[specs___, 𝔼[L_, Q_, P_]] :=
   𝕆U[specs, SS@Normal[P e^(L+Q)]];
  σrs_[c_. * u_U] :=
   (c /. (t : cp)j_ ⧴ tj/.{rs}) U[List @@ (u /. v_j_ ⧴ vj/.{rs})];
  mj_→k_[c_. * u_U] :=
   CE[((c /. (t : cp)j → tk) DeleteCases[u, _j|k]) **
     U @@ Cases[u, w_j ⧴ wk] ** U @@ Cases[u, _k]];
  U /: c_. * u_U * v_U := CE[c u ** v];
  Si_[c_. * u_U] :=
   CE[((c /. Si[U, Centrals]) DeleteCases[u, _i]) **
     Ui[NCM @@ Reverse@Cases[u, x_i ⧴ S@U@x]]];
  Δi_→j_,k_[c_. * u_U] :=
   CE[((c /. Δi→j,k[U, Centrals]) DeleteCases[u, _i]) **
     (NCM @@ Cases[u, x_i ⧴ σ1→j,2→k@Δ@U@x] /.
       NCM[] → U[])]; ]
```

DeclareMorphism

```
DeclareMorphism[m_, U_ → V_, ongs_List, oncs_List: {}] := (
  Replace[ongs, {(g_ → img_) ⧴ (m[U[g]] = img),
    (g_ ⧴ img_) ⧴ (m[U[g]] := img /. $trim)}, {1}];
  m[1U] = 1V;
  m[U[g_i_]] := Vi[m[U@g]];
  m[U[vs__]] := NCM @@ (m /@ U /@ {vs});
  m[ℰ_] := Simp[ℰ /. oncs /. u_U ⧴ m[u]] /. $trim; )
```

Meta-Operations

```
σrs__[ℰ_Plus] := σrs /@ ℰ;
mj_→j_ = Identity; mj_→k_[0] = 0;
mj_→k_[ℰ_Plus] := Simp[mj→k /@ ℰ];
mis__,i_,j_→k_[ℰ_] := mj→k@mis,i→j@ℰ;
Si_[ℰ_Plus] := Simp[Si /@ ℰ];
Δis__[ℰ_Plus] := Simp[Δis /@ ℰ];
```

Implementing CU $= \mathcal{U}(\mathrm{sl}_2^{\gamma\epsilon})$

```
DeclareAlgebra[CU, Generators → {y, a, x}, Centrals → {t}];
B[aCU, yCU] = -γ yCU; B[xCU, aCU] = -γ xCU;
B[xCU, yCU] = 2 ϵ aCU - t 1CU;
(S@yCU = -yCU; S@aCU = -aCU; S@xCU = -xCU;)
Si_[CU, Centrals] = {ti → -ti};
Δ@yCU = CU@y1 + CU@y2; Δ@aCU = CU@a1 + CU@a2;
Δ@xCU = CU@x1 + CU@x2;
Δi_→j_,k_[CU, Centrals] = {ti → tj + tk};
```

Implementing QU $= \mathcal{U}_q(\mathrm{sl}_2^{\gamma\epsilon})$

```
DeclareAlgebra[QU, Generators → {y, a, x},
  Centrals → {t, T}];
B[aQU, yQU] = -γ yQU; B[xQU, aQU] = -γ QU@x;
B[xQU, yQU] := SS[qħ - 1] QU@{y, x} +
   𝕆QU[{a}, SS[(1 - T e^(-2 ϵ a ħ)) / ħ]];
(S@yQU := 𝕆QU[{a, y}, SS[-T^-1 e^(ħ ϵ a) y]]; S@aQU = -aQU;
 S@xQU := 𝕆QU[{a, x}, SS[-e^(ħ ϵ a) x]];)
Si_[QU, Centrals] = {ti → -ti, Ti → Ti^-1};
Δ@yQU := 𝕆QU[{y1, a1}1, {y2}2, SS[y1 + T1 e^(-ħ ϵ a1) y2]];
Δ@aQU = QU@a1 + QU@a2;
Δ@xQU := 𝕆QU[{a1, x1}1, {x2}2, SS[x1 + e^(-ħ ϵ a1) x2]];
Δi_→j_,k_[QU, Centrals] = {ti → tj + tk, Ti → Tj Tk};
```

The representation ρ

```
ρ@yCU = ρ@yQU = (0 0
                 ϵ 0); ρ@aCU = ρ@aQU = (γ 0
                                        0 0);
ρ@xCU = (0 γ
         0 0); ρ@xQU = (0 (1 - e^(-γ ϵ ħ)) / (ϵ ħ)
                        0          0          );
ρ[e^ℰ_] := MatrixExp[ρ[ℰ]];
ρ[ℰ_] :=
 (ℰ /. T2t /. t → γ ϵ /.
   (U : CU | QU)[u___] ⧴ Fold[Dot, (1 0
                                    0 1), ρ /@ U /@ {u}])
```

tSW

Goal. In either U, compute $F=\mathbb{e}^{-\eta y}\,\mathbb{e}^{\xi x}\,\mathbb{e}^{\eta y}\,\mathbb{e}^{-\xi x}$. First compute $G = \mathbb{e}^{\xi x}\, y\, \mathbb{e}^{-\xi x}$, a finite sum. Now F satisfies the ODE $\partial_\eta F=\partial_\eta\left(\mathbb{e}^{-\eta y}\,\mathbb{e}^{\eta G}\right)=-yF+FG$ with initial conditions $F(\eta=0)=1$. So we set it up and solve:

Video and more at http://www.math.toronto.edu/~drorbn/Talks/Matemale-1804/