```
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, g_{i_} → {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;
  U_{i_}[ℰ_] := ℰ /. {t : cp :→ t_{i}, u_U :→ (#_{i} &) /@ u};
  U_{i_}[NCM[]] = U@{} = 1_{U} = U[];
  B[U@(x_)_{i_}, U@(y_)_{i_}] := U_{i}@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_. 1_{U}) := CE[c x]; (c_. 1_{U}) ** 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;
  𝕆_{U}[specs___, poly_] := Module[{sp, null, vs, us},
    sp = Replace[{specs}, l_List :→ l_{null}, {1}];
    vs = Join @@ (First /@ sp);
    us = Join @@ (sp /. l_{s_} :→ (l /. x_{i_} :→ x_{s}));
    CE[Total[
       CoefficientRules[poly, vs] /. (p_ → c_) :→ c U@(us^p)
      ]] /. x_{null} :→ x];
  𝕆_{U}[specs___, 𝔼[L_, Q_, P_]] :=
   𝕆_{U}[specs, SS@Normal[P e^(L+Q)]];
  σ_{rs___}[c_. * u_U] :=
   (c /. (t : cp)_{j_} :→ t_{j /. {rs}}) U[List @@ (u /. v_{j_} :→ v_{j /. {rs}})];
  m_{j_→k_}[c_. * u_U] :=
   CE[((c /. (t : cp)_{j} → t_{k}) DeleteCases[u, _{j|k}]) **
     U @@ Cases[u, w_{j} :→ w_{k}] ** U @@ Cases[u, _{k}]];
  U /: c_. * u_U * v_U := CE[c u ** v];
  S_{i_}[c_. * u_U] :=
   CE[((c /. S_{i}[U, Centrals]) DeleteCases[u, _{i}]) **
     U_{i}[NCM @@ Reverse@Cases[u, x_{i} :→ S@U@x]]];
  Δ_{i_→j_,k_}[c_. * u_U] :=
   CE[((c /. Δ_{i→j,k}[U, Centrals]) DeleteCases[u, _{i}]) **
     (NCM @@ Cases[u, x_{i} :→ σ_{1→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[1_{U}] = 1_{V};
  m[U[g_{i_}]] := V_{i}[m[U@g]];
  m[U[vs__]] := NCM @@ (m /@ U /@ {vs});
  m[ℰ_] := Simp[ℰ /. oncs /. u_U :→ m[u]] /. $trim; )
```

Meta-Operations

```
σ_{rs___}[ℰ_Plus] := σ_{rs} /@ ℰ;
m_{j_→j_} = Identity; m_{j_→k_}[0] = 0;
m_{j_→k_}[ℰ_Plus] := Simp[m_{j→k} /@ ℰ];
m_{is___,i_,j_→k_}[ℰ_] := m_{j→k}@m_{is,i→j}@ℰ;
S_{i_}[ℰ_Plus] := Simp[S_{i} /@ ℰ];
Δ_{is___}[ℰ_Plus] := Simp[Δ_{is} /@ ℰ];
```

Implementing CU = $\mathcal{U}(sl_2^{\gamma\epsilon})$

```
DeclareAlgebra[CU, Generators → {y, a, x}, Centrals → {t}];
B[a_{CU}, y_{CU}] = -γ y_{CU}; B[x_{CU}, a_{CU}] = -γ x_{CU};
B[x_{CU}, y_{CU}] = 2 ϵ a_{CU} - t 1_{CU};
(S@y_{CU} = -y_{CU}; S@a_{CU} = -a_{CU}; S@x_{CU} = -x_{CU};)
S_{i_}[CU, Centrals] = {t_{i} → -t_{i}};
Δ@y_{CU} = CU@y_{1} + CU@y_{2}; Δ@a_{CU} = CU@a_{1} + CU@a_{2};
Δ@x_{CU} = CU@x_{1} + CU@x_{2};
Δ_{i_→j_,k_}[CU, Centrals] = {t_{i} → t_{j} + t_{k}};
```

Implementing QU = $\mathcal{U}_q(sl_2^{\gamma\epsilon})$

```
DeclareAlgebra[QU, Generators → {y, a, x},
  Centrals → {t, T}];
B[a_{QU}, y_{QU}] = -γ y_{QU}; B[x_{QU}, a_{QU}] = -γ QU@x;
B[x_{QU}, y_{QU}] := SS[q_{ℏ} - 1] QU@{y, x} +
   𝕆_{QU}[{a}, SS[(1 - T e^(-2 ϵ a ℏ)) / ℏ]];
(S@y_{QU} := 𝕆_{QU}[{a, y}, SS[-T^(-1) e^(ℏ ϵ a) y]]; S@a_{QU} = -a_{QU};
 S@x_{QU} := 𝕆_{QU}[{a, x}, SS[-e^(ℏ ϵ a) x]];)
S_{i_}[QU, Centrals] = {t_{i} → -t_{i}, T_{i} → T_{i}^(-1)};
Δ@y_{QU} := 𝕆_{QU}[{y_{1}, a_{1}}_{1}, {y_{2}}_{2}, SS[y_{1} + T_{1} e^(-ℏ ϵ a_{1}) y_{2}]];
Δ@a_{QU} = QU@a_{1} + QU@a_{2};
Δ@x_{QU} := 𝕆_{QU}[{a_{1}, x_{1}}_{1}, {x_{2}}_{2}, SS[x_{1} + e^(-ℏ ϵ a_{1}) x_{2}]];
Δ_{i_→j_,k_}[QU, Centrals] = {t_{i} → t_{j} + t_{k}, T_{i} → T_{j} T_{k}};
```

The representation ρ

```
ρ@y_{CU} = ρ@y_{QU} = {{0, 0}, {ϵ, 0}}; ρ@a_{CU} = ρ@a_{QU} = {{γ, 0}, {0, 0}};
ρ@x_{CU} = {{0, γ}, {0, 0}}; ρ@x_{QU} = {{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 $F = e^{-\eta y} e^{\xi x} e^{\eta y} e^{-\xi x}$. First compute $G = e^{\xi x} y e^{-\xi x}$, a finite sum. Now F satisfies the ODE $\partial_\eta F = \partial_\eta (e^{-\eta y} e^{\eta G}) = -yF + FG$ 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/