

```

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[_] := Collect[_] /. $trim;
  U_i[_] := _ /. {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[_NonCommutativeMultiply] := U /@ _;
  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 -> x_s));
    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^{+Q}]];
  sigma_rs__[c_. * u_U] :=
  (c /. (t : cp)_j -> t_j /. {rs}) U[List@@(u /. v_j -> v_j /. {rs})];
  m_j_to_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] ]];
  Delta_i_to_j_k_[c_. * u_U] :=
  CE[ ((c /. Delta_i_to_j_k_[U, Centrals]) DeleteCases[u, _i]) **
  (NCM@@Cases[u, x_i -> sigma_{1->j, 2->k}@Delta@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[_] := Simp[_] /. oncs /. u_U -> m[u] /. $trim; )

```

## Meta-Operations

```

sigma_rs__[E_Plus] := sigma_rs /@ E;
m_j_to_j_ = Identity; m_j_to_k_[0] = 0;
m_j_to_k_[E_Plus] := Simp[m_j_to_k_ /@ E];
m_is___, i_, j_to_k_[E_] := m_j_to_k_@m_is, i -> j @ E;
S_i[E_Plus] := Simp[S_i /@ E];
Delta_is__[E_Plus] := Simp[Delta_is /@ E];

```

## Implementing CU = U(sl\_2^E)

```

DeclareAlgebra[CU, Generators -> {y, a, x}, Centrals -> {t}];
B[a_CU, y_CU] = -y y_CU; B[x_CU, a_CU] = -y x_CU;
B[x_CU, y_CU] = 2 e 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};
Delta@y_CU = CU@y_1 + CU@y_2; Delta@a_CU = CU@a_1 + CU@a_2;
Delta@x_CU = CU@x_1 + CU@x_2;
Delta_i_to_j_k_[CU, Centrals] = {t_i -> t_j + t_k};

```

## Implementing QU = U\_q(sl\_2^E)

```

DeclareAlgebra[QU, Generators -> {y, a, x},
  Centrals -> {t, T}];
B[a_QU, y_QU] = -y y_QU; B[x_QU, a_QU] = -y QU@x;
B[x_QU, y_QU] := SS[q_h - 1] QU@{y, x} +
  O_QU[{a}, SS[(1 - T e^{-2 e a h}) / h]];
(S@y_QU := O_QU[{a, y}, SS[-T^{-1} e^{h e a} y]]; S@a_QU = -a_QU;
  S@x_QU := O_QU[{a, x}, SS[-e^{h e a} x]]);
S_i[QU, Centrals] = {t_i -> -t_i, T_i -> T_i^{-1}};
Delta@y_QU := O_QU[{y_1, a_1}_1, {y_2}_2, SS[y_1 + T_1 e^{-h e a_1} y_2]];
Delta@a_QU = QU@a_1 + QU@a_2;
Delta@x_QU := O_QU[{a_1, x_1}_1, {x_2}_2, SS[x_1 + e^{-h e a_1} x_2]];
Delta_i_to_j_k_[QU, Centrals] = {t_i -> t_j + t_k, T_i -> T_j T_k};

```

## The representation rho

```

rho@y_CU = rho@y_QU = ( 0 0
                      e 0 ); rho@a_CU = rho@a_QU = ( y 0
                                                       0 0 );
rho@x_CU = ( 0 y
             0 0 ); rho@x_QU = ( 0 (1 - e^{-y e a h}) / (e h)
                                 0 0 );
rho[e^E] := MatrixExp[rho[E]];
rho[E_] :=
  (E /. T2t /. t -> y e /.
  (U : CU | QU) [u___] -> Fold[Dot, ( 1 0
                                       0 1 ), rho /@ 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: