

Pensieve header: A unified verification notebook for the \$sl_3\$ project.

Prolog

```
HL[ $\mathcal{E}_-$ ] := Style[ $\mathcal{E}$ , Background  $\rightarrow$  Yellow];
```

Initialization / Utilities

The “degree carrier / filtration parameter” is $\hbar$, and all “coupling constants” are proportional to it.

TD

```
$p = 2; $k = 1;
If[$k == 0,  $\epsilon = 0$ ,  $\epsilon$  /:  $\epsilon^k$  /;  $k > $k$  := 0];
SetAttributes[{SS, SST}, HoldAll];
TRule = { $T_{i_-} \rightarrow e^{\hbar t_i}$ ,  $T \rightarrow e^{\hbar t}$ }; qrule =  $q \rightarrow e^{2e\hbar}$ ;
SS[ $\mathcal{E}_-$ ] := Block[{ $\hbar$ ,  $\epsilon$ }, (* Shielded Series *)
  Collect[Normal@Series[ $\mathcal{E}$ , { $\hbar$ , 0, $p}],  $\hbar$ , Together] ];
SST[ $\mathcal{E}_-$ ] :=
  Block[{ $\hbar$ ,  $\epsilon$ }, Collect[Normal@Series[ $\mathcal{E}$  /. TRule, { $\hbar$ , 0, $p}],  $\hbar$ , Together] ];
Simp[ $\mathcal{E}_-$ , op_] := Collect[ $\mathcal{E}$ , _CU | _QU, op];
Simp[ $\mathcal{E}_-$ ] := Simp[ $\mathcal{E}$ , Collect[Normal@Series[#, { $\hbar$ , 0, $p}],  $\hbar$ , Expand] &];
SimpT[ $\mathcal{E}_-$ ] :=
  Collect[ $\mathcal{E}$ , _CU | _QU, Collect[Normal@Series[#, { $\hbar$ , 0, $p}],  $\hbar$ , Expand] &];
```

Differential polynomials (DP):

Utils

```
DP[ $\alpha \rightarrow D_x, \beta \rightarrow D_y$ ][ $P_-$ ][ $\lambda_-$ ] :=
  Total[CoefficientRules[P, { $\alpha$ ,  $\beta$ }] /. ({ $m_-$ ,  $n_-$ }  $\rightarrow$   $c_-$ )  $\Rightarrow$  c D[ $\lambda$ , { $x$ ,  $m$ }, { $y$ ,  $n$ }]]
```

DeclareAlgebra

QLImplementation

```
Unprotect[NonCommutativeMultiply]; Attributes[NonCommutativeMultiply] = {};
(NCM = NonCommutativeMultiply)[ $x_-$ ] :=  $x$ ;
NCM[ $x_-$ ,  $y_-$ ,  $z_-$ ] := ( $x \otimes y$ )  $\otimes$   $z$ ;
0  $\otimes$  _ = _  $\otimes$  0 = 0;
( $x\_Plus$ )  $\otimes$   $y_-$  := ( $\# \otimes y$ ) & /@  $x$ ;  $x_- \otimes$  ( $y\_Plus$ ) := ( $x \otimes \#$ ) & /@  $y$ ;
B[ $x_-$ ,  $x_-$ ] = 0;
B[ $x_-$ ,  $y_-$ ] :=  $x \otimes y - y \otimes x$ ;
```

QLImplementation

```
DeclareAlgebra[U_Symbol, opts__Rule] := Module[{gp, sr, g, cp, M, CE, pow, k = 0,
  gs = Generators /. {opts}, cs = Centrals /. {opts}},
  (#i = 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[ε_] := Collect[ε, _U, (Expand[#] /. h^d_ /; d > $p &⇒ 0) &];
Ui_ [ε_] := ε /. {t : cp &⇒ ti, u_U &⇒ Replace[u, x_ &⇒ xi, 1]};
Ui_ [NCM[]] = pow[ε_, 0] = U@{} = 1_U = U[];
B[U@(x_)i_, U@(y_)i_] := B[U@xi, U@yi] = 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_. 1_U) := CE[c x]; (c_. 1_U) ⓧ x_ := CE[c x];
(* (a_. * x_U) ** (b_. * y_U) := CE[M[a b, x ** y]]]; *)
(a_. U[xx___, x_]) ⓧ (b_. U[y_, yy___]) := If[OrderedQ[{x, y} /. sr],
  CE@M[a b, U[xx, x, y, yy]],
  U@xx ⓧ CE@M[a b, U@y ⓧ U@x + B[U@x, U@y]] ⓧ 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[ε_NonCommutativeMultiply] := U /@ ε;
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_ &⇒ xs));
  CE[Total[
    CoefficientRules[poly, vs] /. (p_ → c_) &⇒ c U@(us^p)
  ]] /. x_null &⇒ x];
pow[ε_, n_] := pow[ε, n - 1] ⓧ ε;
S_U[ε_, ss___Rule] := CE@Total[
  CoefficientRules[ε, First /@ {ss}] /.
    (p_ → c_) &⇒ c NCM@@MapThread[pow, {Last /@ {ss}, p}]];
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];
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 &⇒ DDj,k@U@x];
Pi_,j_ [c_. * u_U] := CE[(P[U@Cases[c U[], _i] ⓧ (U@@Cases[u, _i]),
  (U@Cases[c U[], _j]) ⓧ (U@@Cases[u, _j])]
  DeleteCases[c U[], _i|j] ⓧ U@@DeleteCases[u, _i|j])] /. U[] → 1];

```

DeclareMorphism

QLImplementation

```

DeclareMorphism[m_, U_ → V_, ongs_List, oncs_List:{}] := (
  Replace[ongs, (g_ → img_) ⇒ (m[U[g]] = img), {1}];
  m[1_U] = 1_V;
  m[U[g_i_]] := V_i[m[U@g]];
  m[U[vs_]] := NCM@@(m/@U/@{vs});
  m[ε_] := Simp[ε /. oncs /. u_U ⇒ m[u]]; )

```

Meta-Operations

QLImplementation

```

m_{j→j_} = Identity;
m_{j→k_}[ε_Plus] := Simp[m_{j→k_}/@ε];
m_{is_..., i_→j_→k_}[ε_] := m_{j→k_}@m_{is, i→j}@ε;
S_{i_}[ε_Plus] := Simp[S_i/@ε];
Δ_{i→j_→k_}[ε_Plus] := Simp[Δ_{i→j, k_}/@ε];
P_{i_→j_}[ε_Plus] := Simp[P_{i, j_}/@ε];
Δ_{i→j_→k_}[c_ * QU[]] :=
  Collect[(c /. Δ_{i→j, k_}[QU, Centrals]) QU[], _U, (Expand[#] /. h^{d-} /. d > $p ⇒ 0) &];
Δ_{i→j_→k_}[c_ * CU[]] :=
  Collect[(c /. Δ_{i→j, k_}[CU, Centrals]) CU[], _U, (Expand[#] /. h^{d-} /. d > $p ⇒ 0) &];

```

QLImplementation

Rule::rhs: Pattern j_ appears on the right-hand side of rule j_ → j_. >>

Implementing $CU = \mathcal{U}(sl_3^\epsilon)$

CU

```

DeclareAlgebra[CU, Generators → {X, Y, Z, b, a, z, y, x}, Centrals → {ss, tt}];
B[XCU, aCU] = 2 CU@x; B[XCU, bCU] = -CU@x;
B[YCU, bCU] = 2 CU@y; B[YCU, aCU] = -CU@y;
B[ZCU, aCU] = CU@z; B[ZCU, bCU] = CU@z;
B[XCU, YCU] = ZCU; B[XCU, ZCU] = 0; B[aCU, bCU] = 0; B[YCU, ZCU] = 0;
B[aCU, XCU] = 2 XCU; B[bCU, XCU] = -XCU;
B[aCU, YCU] = -YCU; B[bCU, YCU] = 2 YCU;
B[aCU, ZCU] = ZCU; B[bCU, ZCU] = ZCU;
B[XCU, XCU] = ss 1CU + 2 ∈ CU[a];
B[YCU, YCU] = tt 1CU + 2 ∈ CU[b];
B[ZCU, ZCU] = (ss + tt) 1CU + 2 ∈ (aCU + bCU);
B[YCU, ZCU] = XCU; B[XCU, ZCU] = -CU[Y];
B[ZCU, XCU] = 2 ∈ YCU;
B[ZCU, YCU] = -2 ∈ CU[X]; B[YCU, XCU] = 0;
B[XCU, YCU] = 0;
B[XCU, YCU] = 2 ∈ ZCU;
B[XCU, ZCU] = 0;
B[ZCU, YCU] = 0;

```

Verifying associativity on triples of generators:

```

With[{bas = CU /@ {X, Y, Z, b, a, z, y, x}},
  Table[z1 ⓧ (z2 ⓧ z3) - (z1 ⓧ z2) ⓧ z3 // Simp // HL,
    {z1, bas}, {z2, bas}, {z3, bas}] // Flatten // Union]
{0}

With[{bas = CU /@ {X, Y, Z, b, a, z, y, x}},
  Table[B[aa, B[bb, cc]] // Expand + B[bb, B[cc, aa]] // Expand + B[cc, B[aa, bb]] // Expand //
    Simp, {aa, bas}, {bb, bas}, {cc, bas}] // Flatten // Union]
{0}

```

Verifying associativity on a “random” triple:

```

With[{z1 = CU[y, y, a, a, x, x], z2 = CU[y, a, x], z3 = CU[y, y, a, x]}, {
  (rhs = (z1 ⓧ z2) ⓧ z3 // Simp) // Short,
  z1 ⓧ (z2 ⓧ z3) - rhs // Simp // HL
}] // Timing
{0.078125, {36 CU[y, y, a, a, z, z, z, x] + <<11>>, 0}}

```

Implementing $QU = \mathcal{U}_q(sl_3^\epsilon)$

```
(*s=exp[2A-B+εa], t=exp[2B-A+εb]*)
```

QU

```

DeclareAlgebra[QU, Generators → {X, Y, Z, b, a, z, y, x}, Centrals → {s, t}];
B[aQU, xQU] = -2 QU@x; B[bQU, xQU] = QU@x;
B[bQU, yQU] = -2 QU@y; B[aQU, yQU] = QU@y;
B[aQU, zQU] = -QU@z; B[bQU, zQU] = -QU@z;
B[yQU, xQU] = -zQU + ε ħ QU[y, x];
B[zQU, xQU] = -ε ħ QU[z, x];
B[bQU, aQU] = 0;
B[zQU, yQU] = ε ħ QU[z, y];
B[aQU, xQU] = 2 XQU; B[bQU, xQU] = -XQU;
B[aQU, yQU] = -YQU; B[bQU, yQU] = 2 YQU;
B[aQU, zQU] = ZQU; B[bQU, zQU] = ZQU;
B[xQU, xQU] = -2 ε ħ QU@{X, x} +  $\frac{(1-s) QU[]}{\hbar}$  + 2 s ε QU[a];
B[yQU, yQU] = -2 ε ħ QU@{Y, y} +  $\frac{(1-t) QU[]}{\hbar}$  + 2 t ε QU[b];
B[zQU, zQU] = -2 ε ħ QU@{Z, z} +  $\frac{(1-s t) QU[]}{\hbar}$  + 2 s t ε QU[a] + 2 s t ε QU[b];
B[yQU, zQU] = XQU - ε ħ QU@{Z, y};
B[xQU, zQU] = -ε ħ QU@{Z, x} - (s + s ε ħ) QU[Y] + 2 s ε ħ QU[Y, a];
B[zQU, xQU] = 2 ε yQU - ε ħ QU@{X, z};
B[zQU, yQU] = -2 t ε QU[x] - ε ħ QU[Y, z]; B[yQU, xQU] = ε ħ QU[X, y];
B[xQU, yQU] = ε ħ QU[Y, x];
B[xQU, yQU] = 2 ε zQU - ε ħ QU[X, Y];
B[xQU, zQU] = ε ħ QU[X, Z];
B[zQU, yQU] = ε ħ QU[Y, Z];

```

```

With[{bas = QU /@ {X, Y, Z, b, a, x, y, z}},
  Table[{z1, z2} → Simp[z1 ⌗ z2 - z2 ⌗ z1], {z1, bas}, {z2, bas}] ]
{
  {QU[X], QU[X]} → 0, {QU[X], QU[Y]} → -2 ∈ QU[Z] + ∈ ħ QU[X, Y],
  {QU[X], QU[Z]} → ∈ ħ QU[X, Z], {QU[X], QU[b]} → QU[X], {QU[X], QU[a]} → -2 QU[X],
  {QU[X], QU[x]} →  $\frac{(-1+s) QU[]}{\hbar} - 2 s \in QU[a] + 2 \in \hbar QU[X, x]$ ,
  {QU[X], QU[y]} → 0, {QU[X], QU[z]} → -2 ∈ QU[y] + ∈ ħ QU[X, z]},
  {
    {QU[Y], QU[X]} → 2 ∈ QU[Z] - ∈ ħ QU[X, Y], {QU[Y], QU[Y]} → 0,
    {QU[Y], QU[Z]} → ∈ ħ QU[Y, Z], {QU[Y], QU[b]} → -2 QU[Y], {QU[Y], QU[a]} → QU[Y],
    {QU[Y], QU[x]} → 0, {QU[Y], QU[y]} →  $\frac{(-1+t) QU[]}{\hbar} - 2 t \in QU[b] + 2 \in \hbar QU[Y, y]$ ,
    {QU[Y], QU[z]} → 2 t ∈ QU[x] + ∈ ħ QU[Y, z]},
    {
      {QU[Z], QU[X]} → - ∈ ħ QU[X, Z], {QU[Z], QU[Y]} → - ∈ ħ QU[Y, Z],
      {QU[Z], QU[Z]} → 0, {QU[Z], QU[b]} → -QU[Z], {QU[Z], QU[a]} → -QU[Z],
      {QU[Z], QU[x]} → (s + s ∈ ħ) QU[Y] + ∈ ħ QU[Z, x], {QU[Z], QU[y]} → -QU[X] + ∈ ħ QU[Z, y],
      {QU[Z], QU[z]} →  $\frac{(-1+s t) QU[]}{\hbar} - 2 s t \in QU[a] - 2 s t \in QU[b] + 2 \in \hbar QU[Z, z]$ },
      {
        {QU[b], QU[X]} → -QU[X], {QU[b], QU[Y]} → 2 QU[Y], {QU[b], QU[Z]} → QU[Z],
        {QU[b], QU[b]} → 0, {QU[b], QU[a]} → 0, {QU[b], QU[x]} → QU[x],
        {QU[b], QU[y]} → -2 QU[y], {QU[b], QU[z]} → -QU[z]},
        {
          {QU[a], QU[X]} → 2 QU[X], {QU[a], QU[Y]} → -QU[Y], {QU[a], QU[Z]} → QU[Z],
          {QU[a], QU[b]} → 0, {QU[a], QU[a]} → 0, {QU[a], QU[x]} → -2 QU[x],
          {QU[a], QU[y]} → QU[y], {QU[a], QU[z]} → -QU[z]},
          {
            {QU[x], QU[X]} →  $\frac{(1-s) QU[]}{\hbar} + 2 s \in QU[a] - 2 \in \hbar QU[X, x]$ ,
            {QU[x], QU[Y]} → 0, {QU[x], QU[Z]} → (-s - s ∈ ħ) QU[Y] - ∈ ħ QU[Z, x],
            {QU[x], QU[b]} → -QU[x], {QU[x], QU[a]} → 2 QU[x], {QU[x], QU[x]} → 0,
            {QU[x], QU[y]} → QU[z] - ∈ ħ QU[y, x], {QU[x], QU[z]} → ∈ ħ QU[z, x]},
            {
              {QU[y], QU[X]} → 0, {QU[y], QU[Y]} →  $\frac{(1-t) QU[]}{\hbar} + 2 t \in QU[b] - 2 \in \hbar QU[Y, y]$ ,
              {QU[y], QU[Z]} → QU[X] - ∈ ħ QU[Z, y], {QU[y], QU[b]} → 2 QU[y], {QU[y], QU[a]} → -QU[y],
              {QU[y], QU[x]} → -QU[z] + ∈ ħ QU[y, x], {QU[y], QU[y]} → 0, {QU[y], QU[z]} → - ∈ ħ QU[z, y]},
              {
                {QU[z], QU[X]} → 2 ∈ QU[y] - ∈ ħ QU[X, z], {QU[z], QU[Y]} → -2 t ∈ QU[x] - ∈ ħ QU[Y, z],
                {QU[z], QU[Z]} →  $\frac{(1-s t) QU[]}{\hbar} + 2 s t \in QU[a] + 2 s t \in QU[b] - 2 \in \hbar QU[Z, z]$ ,
                {QU[z], QU[b]} → QU[z], {QU[z], QU[a]} → QU[z], {QU[z], QU[x]} → - ∈ ħ QU[z, x],
                {QU[z], QU[y]} → ∈ ħ QU[z, y], {QU[z], QU[z]} → 0}
              }
            }
          }
        }
      }
    }
  }

```

Verifying associativity on triples of generators:

```
With[{bas = QU /@ {Z, X, Y, a, b, y, x, z}},  
  Table[HL[z1  $\boxtimes$  (z2  $\boxtimes$  z3) - (z1  $\boxtimes$  z2)  $\boxtimes$  z3 // Simp],  
    {z1, bas}, {z2, bas}, {z3, bas} ] // Flatten // Union]  
{0}
```