Hidden $\TeX$

A Timing Game

Headers

pdf

Pensieve header: Verifying 2-parameter type A_n quantum groups following Ian Martin, Alexander Tsymbaliuk, Two-Parameter Quantum Groups and R-Matrices: Classical Types and Orthogonal bases for two-parameter quantum groups.

pdf

In[ ]:=
```
SetDirectory@"C:\\drorbn\\AcademicPensieve\\Projects\\SolvablePBW";
<< KnotTheory`
```

pdf

```
Loading KnotTheory` version of October 29, 2024, 10:29:52.1301.
Read more at http://katlas.org/wiki/KnotTheory.
```

nb2tex

In[ ]:=
```
Get["../nb2tex/nb2tex.m"];
$NotebookName = FileBaseName@FileNameTake@NotebookFileName[EvaluationNotebook[]]
nb2tex[$NotebookName, PDFWidth → 7.5, PDFFolder → "Snips" <> $NotebookName];
texfile = OpenAppend[$NotebookName <> ".tex"];
WriteLine[texfile, "\\end{document}"]; Close[texfile];
Run@
  ("\"C:\\Users\\drorb\\AppData\\Local\\Programs\\MiKTeX\\miktex\\bin\\x64\\pdflatex.exe\"
      " <> $NotebookName <> ".tex");
```

Time: 5.31 seconds Out[ ]=

```
TwoParameterQuantumGroups
```

pdf

Initialization / Utilities

pdf

In[◦]:=

```
$k = 1; $U = QU; $E := {$k};
$trim := {ϵ^k_. /; k > $k → 0};
SetAttributes[{SS, SST}, HoldAll];
q_ħ_ := ⅇ^(γ ϵ ħ);
(* Upper to lower and lower to Upper: *)
U2l = {B_i_^p_. → ⅇ^(-p ħ γ b_i), B^p_. → ⅇ^(-p ħ γ b), T_i_^p_. → ⅇ^(p ħ t_i), T^p_. → ⅇ^(p ħ t), 𝒜_i_^p_. → ⅇ^(p γ α_i), 𝒜^p_. → ⅇ^(p γ α)};
l2U = {ⅇ^(c_. b_i_ + d_.) :→ B_i^(-c/(ħ γ)) ⅇ^d, ⅇ^(c_. b + d_.) :→ B^(-c/(ħ γ)) ⅇ^d,
    ⅇ^(c_. t_i_ + d_.) :→ T_i^(c/ħ) ⅇ^d, ⅇ^(c_. t + d_.) :→ T^(c/ħ) ⅇ^d,
    ⅇ^(c_. α_i_ + d_.) :→ 𝒜_i^(c/γ) ⅇ^d, ⅇ^(c_. α + d_.) :→ 𝒜^(c/γ) ⅇ^d,
    ⅇ^ℰ_ :→ ⅇ^(Expand@ℰ)};
SS[ℰ_, op_] := Collect[Normal@Series[ℰ, {ϵ, 0, $k}], ϵ, op];
SS[ℰ_] := SS[ℰ, Together];
SST[ℰ_, op___] := SS[ℰ /. U2l, op];
Simp[ℰ_, op_] := Collect[ℰ, _CU | _QU, op];
Simp[ℰ_] := Simp[ℰ, SS[#, Expand] &];
SimpT[ℰ_] := Collect[ℰ, _CU | _QU, SST[#, Expand] &];
Kδ /: Kδ_{i_,j_} := If[i === j, 1, 0];
c_Integer_k_Integer := c + O[ϵ]^(k+1);
```

pdf

DeclareAlgebra

pdf

In[◦]:=

```
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];
```

pdf

In[◦]:=

```
DeclareAlgebra[U_Symbol, opts__Rule] := Module[
  {gp, sr, g, cp, M, pow, k = 0,
   gs = Generators /. {opts},
   cs = Centrals /. {opts} /. Centrals → {}
  },
  U[Generators] = gs;
  (#_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] /. ⅇ^x_ :> ⅇ^Expand[x] /. $trim;
U_i_[ℰ_] := ℰ /. {t : cp :> t_i, u_U :> (#_i &) /@ u};
U_i_[NCM[]] = pow[ℰ_, 0] = 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)]];
pow[ℰ_, n_] := pow[ℰ, n - 1] ⊠ ℰ;
𝕊_U[ℰ_, ss___Rule] := CE@Total[
   CoefficientRules[ℰ, First /@ {ss}] /.
    (p_ → c_) :> c NCM @@ MapThread[pow, {Last /@ {ss}, p}]];
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[])]; ]
```

pdf

Implementing QU = $U_{r,s}[A_n]$

pdf

Conventions:

$\{e_i, f_i, \omega_i^{\pm 1}, (\omega_i')^{\pm 1}, r, s\} \leftrightarrow \{x_{i,i+1}, y_{i,i+1}, e^{\pm v_i}, e^{\pm w_i}, e^r, e^s\}$

We are assuming that $r_i = r$ and $s_i = s$.

In[•]:=
```
$n = 3;
Flatten@{
  Table[x_{i,j}, {i, 1, $n}, {j, i + 1, $n + 1}],
  Table[y_{i,j}, {i, 1, $n}, {j, i + 1, $n + 1}],
  Table[{v_i, w_i}, {i, $n}]
 }
```

Time: 0.00 seconds Out[•]=

$\{x_{1,2}, x_{1,3}, x_{1,4}, x_{2,3}, x_{2,4}, x_{3,4}, y_{1,2}, y_{1,3}, y_{1,4}, y_{2,3}, y_{2,4}, y_{3,4}, v_1, w_1, v_2, w_2, v_3, w_3\}$

pdf

In[•]:=
```
$n = 3;
DeclareAlgebra[QU,
  Generators → Flatten@{
     Table[x_{i,j}, {i, 1, $n}, {j, i + 1, $n + 1}],
     Table[y_{i,j}, {i, 1, $n}, {j, i + 1, $n + 1}],
     Table[{v_i, w_i}, {i, $n}]
    },
  Centrals → {}
 ];
```

pdf

In[•]:=
```
Ringel[i_, j_] := Which[
  i + 1 < j ∨ i > j, 0,
  i + 1 == j, -1,
  i == j, 1
 ]
```

pdf

```
Do[
  B[QU@v_i, QU@v_j] = B[QU@v_i, QU@w_j] = B[QU@w_i, QU@w_j] = 0;
  B[QU@v_i, QU@x_{j,j+1}] = (Ringel[j, i] r - Ringel[i, j] s) QU[x_{j,j+1}, v_i]
  ,
  {i, 1, $n}, {j, 1, $n}
 ];
```

pdf

```
B[a_QU, x_QU] = -2 QU@x;
B[b_QU, x_QU] = QU@x;
B[b_QU, y_QU] = -2 QU@y;
B[a_QU, y_QU] = QU@y;
B[a_QU, z_QU] = -QU@z;
B[b_QU, z_QU] = -QU@z;
B[y_QU, x_QU] = -z_QU + ∈ ℏ QU[y, x];
B[z_QU, x_QU] = -∈ ℏ QU[z, x];
B[b_QU, a_QU] = 0;
B[z_QU, y_QU] = ∈ ℏ QU[z, y];
B[a_QU, X_QU] = 2 X_QU;
B[b_QU, X_QU] = -X_QU;
B[a_QU, Y_QU] = -Y_QU;
B[b_QU, Y_QU] = 2 Y_QU;
B[a_QU, Z_QU] = Z_QU;
B[b_QU, Z_QU] = Z_QU;
B[x_QU, X_QU] = -2 ∈ ℏ QU@{X, x} + ((1 - s) QU[])/ℏ + 2 s ∈ QU[a];
B[y_QU, Y_QU] = -2 ∈ ℏ QU@{Y, y} + ((1 - t) QU[])/ℏ + 2 t ∈ QU[b];
B[z_QU, Z_QU] = -2 ∈ ℏ QU@{Z, z} + ((1 - s t) QU[])/ℏ + 2 s t ∈ QU[a] + 2 s t ∈ QU[b];
B[y_QU, Z_QU] = X_QU - ∈ ℏ QU@{Z, y};
B[x_QU, Z_QU] = -∈ ℏ QU@{Z, x} - (s + s ∈ ℏ) QU[Y] + 2 s ∈ ℏ QU[Y, a];
B[z_QU, X_QU] = 2 ∈ y_QU - ∈ ℏ QU@{X, z};
B[z_QU, Y_QU] = -2 t ∈ QU[x] - ∈ ℏ QU[Y, z];
B[y_QU, X_QU] = ∈ ℏ QU[X, y];
B[x_QU, Y_QU] = ∈ ℏ QU[Y, x];
B[X_QU, Y_QU] = 2 ∈ Z_QU - ∈ ℏ QU[X, Y];
B[X_QU, Z_QU] = ∈ ℏ QU[X, Z];
B[Z_QU, Y_QU] = ∈ ℏ QU[Y, Z];
```

In[∘]:= `QU@Generators`

Time: 0.02 seconds Out[∘]=

{X, Y, Z, b, a, z, y, x}

```
In[ ]:= $Basis = QU /@ QU@Generators;
        Module[{a, b}, Table[{a, b} → CE[B[a, b]], {a, $Basis}, {b, $Basis}]] // MatrixForm
```

Time: 0.02 seconds Out[ ]=

```
( {QU[X], QU[X]} → 0                                      {QU[X], QU[Y]} → 2 ∈ QU[Z] - ∈ QU
  {QU[Y], QU[X]} → -2 ∈ QU[Z] + ∈ QU[X, Y]                {QU[Y], QU[Y]} → 0
  {QU[Z], QU[X]} → -∈ QU[X, Z]                            {QU[Z], QU[Y]} → ∈ QU[Y, Z
  {QU[b], QU[X]} → -QU[X]                                 {QU[b], QU[Y]} → 2 QU[Y]
  {QU[a], QU[X]} → 2 QU[X]                                {QU[a], QU[Y]} → -QU[Y]
  {QU[z], QU[X]} → 2 ∈ QU[y] - ∈ QU[X, z]                 {QU[z], QU[Y]} → -2 t ∈ QU[x] - ∈
  {QU[y], QU[X]} → ∈ QU[X, y]                             {QU[y], QU[Y]} → (1 - t) QU[] + 2 t ∈ QU[b
  {QU[x], QU[X]} → (1 - s) QU[] + 2 s ∈ QU[a] - 2 ∈ QU[X, x]  {QU[x], QU[Y]} → ∈ QU[Y, x
```

```
In[ ]:= Module[{a, b, c},
         DeleteCases[Flatten[Table[{a, b, c} → CE[B[a, B[b, c]] + B[b, B[c, a]] + B[c, B[a, b]]],
            {a, $Basis}, {b, $Basis}, {c, $Basis}]], _ → 0]
        ]
```

Time: 0.80 seconds Out[ ]=

```
{}
```