Pensieve Header: The programs running the playground - a minimalistic Lie quotient of A^w, with a BCH product law.

The cell below is copied from BCH.nb:

```
BCH[b_, x_, y_] := Module[
   {BCH8, B, X, Y, Ev},
   BCH8 = B[X, Y] / 2 + B[X, B[X, Y]] / 12 -
     B[X, B[X, B[X, B[X, Y]]]] / 720 + B[X, B[X, B[X, B[X, B[X, B[X, Y]]]]]] / 30 240 -
     B[X, B[X, B[X, B[X, B[X, B[Y, B[X, Y]]]]]]] / 20 160 -
     B[X, B[X, B[X, B[X, B[Y, B[X, Y]]]]]] / 10 080 +
     B[X, B[X, B[X, B[X, B[Y, B[Y, B[X, Y]]]]]]] / 20 160 -
     B[X, B[X, B[X, B[Y, B[X, Y]]]]] / 1440 + B[X, B[X, B[X, B[Y, B[X, B[X, B[X, Y]]]]]]] /
      12 096 - B[X, B[X, B[X, B[Y, B[X, B[Y, B[X, Y]]]]]]] / 5040 -
     B[X, B[X, B[X, B[Y, B[Y, B[X, Y]]]]]] / 7560 -
     (5 * B[X, B[X, B[X, B[Y, B[Y, B[Y, B[X, Y]]]]]]]) / 24 192 -
     B[X, B[X, B[Y, B[X, Y]]]] / 120 + B[X, B[X, B[Y, B[X, B[X, B[X, Y]]]]]] / 2520 +
     B[X, B[X, B[Y, B[X, B[X, B[Y, B[X, Y]]]]]]] / 3360 +
     B[X, B[X, B[Y, B[X, B[Y, B[X, Y]]]]]] / 3360 +
     B[X, B[X, B[Y, B[X, B[Y, B[Y, B[X, Y]]]]]]] / 2520 -
     B[X, B[X, B[Y, B[Y, B[X, Y]]]]] / 720 - B[X, B[X, B[Y, B[Y, B[Y, B[X, Y]]]]]] / 3360 +
     B[X, B[X, B[Y, B[Y, B[Y, B[Y, B[X, Y]]]]]]] / 20 160 -
     B[X, B[Y, B[X, Y]]] / 24 + B[X, B[Y, B[X, B[X, B[X, Y]]]]] / 720 -
     B[X, B[Y, B[X, B[X, B[X, B[X, B[X, Y]]]]]]] / 20 160 -
     B[X, B[Y, B[X, B[X, B[X, B[Y, B[X, Y]]]]]]] / 6720 +
     B[X, B[Y, B[X, B[X, B[Y, B[X, Y]]]]]] / 1680 +
     B[X, B[Y, B[X, B[X, B[Y, B[Y, B[X, Y]]]]]]] / 15 120 +
     B[X, B[Y, B[X, B[Y, B[X, Y]]]]] / 240 - B[X, B[Y, B[X, B[Y, B[X, B[X, B[X, Y]]]]]]] /
      10 080 - B[X, B[Y, B[X, B[Y, B[X, B[Y, B[X, Y]]]]]]] / 2016 +
     B[X, B[Y, B[X, B[Y, B[Y, B[X, Y]]]]]] / 1260 -
     B[X, B[Y, B[X, B[Y, B[Y, B[Y, B[X, Y]]]]]]] / 20 160 -
     B[X, B[Y, B[Y, B[X, Y]]]] / 360 + B[X, B[Y, B[Y, B[X, B[X, B[Y, B[X, Y]]]]]]] / 20 160 -
     B[X, B[Y, B[Y, B[X, B[Y, B[Y, B[X, Y]]]]]]] / 10 080 +
     B[X, B[Y, B[Y, B[Y, B[X, Y]]]]] / 1440 + B[X, B[Y, B[Y, B[Y, B[Y, B[X, Y]]]]]] / 10 080 -
     B[X, B[Y, B[Y, B[Y, B[Y, B[Y, B[X, Y]]]]]]] / 60 480 - B[Y, B[X, Y]] / 12 +
     B[Y, B[X, B[X, B[X, Y]]]] / 360 - B[Y, B[X, B[X, B[X, B[X, B[X, Y]]]]]] / 10 080 -
     B[Y, B[X, B[X, B[X, B[Y, B[X, Y]]]]]] / 3360 +
     B[Y, B[X, B[X, B[Y, B[Y, B[X, Y]]]]]] / 7560 +
     B[Y, B[X, B[Y, B[X, Y]]]] / 120 - B[Y, B[X, B[Y, B[X, B[X, B[X, Y]]]]]] / 5040 -
     B[Y, B[X, B[Y, B[X, B[Y, B[X, B[Y, B[X, Y]]]]]]] / 1008 - B[Y, B[X, B[Y, B[Y, B[Y, B[X, Y]]]]]] /
      10 080 + B[Y, B[Y, B[X, B[X, B[Y, B[X, Y]]]]]] / 10 080 -
     B[Y, B[Y, B[X, B[Y, B[Y, B[X, Y]]]]]] / 5040 + B[Y, B[Y, B[Y, B[X, Y]]]] / 720 -
     B[Y, B[Y, B[Y, B[Y, B[Y, B[X, Y]]]]]] / 30 240;
   Ev[X] = x; Ev[Y] = y;
   Ev[B[u_, v_]] := Ev[B[u, v]] = b[Ev[u], Ev[v]];
   Expand[x + y + (BCH8 /. B[u_, v_] :> Ev[B[u, v]])]
  ];
```

```
If[Head[$DegreeStack] =!= List, $DegreeStack = {Infinity}];
$ToDegree = First[$DegreeStack];
SetAttributes[ToDegree, HoldRest];
ToDegree[m_, expr_] := Module[{res},
   PrependTo[$DegreeStack, $ToDegree = m];
   Clear[h]; h /: h^p_Integer /; p > $ToDegree := 0;
   res = expr;
   $DegreeStack = Rest[$DegreeStack];
   $ToDegree = First[$DegreeStack];
   Clear[h]; h /: h^p_Integer /; p > $ToDegree := 0;
   res
];
PCollect[expr_] := Collect[expr, _ar, Expand];
b[ar[i_, j_], ar[k_, l_]] := Which[
   i == k || i == j || k == l, 0,
   j == l, h[i] ar[k, l] - h[k] ar[i, j],
   j == k && i == l, -h[i] ar[j, i] + h[j] ar[i, j] - h[i] ar[j, j] + h[j] ar[i, i],
   j == k, -h[i] ar[k, l] + h[k] ar[i, l],
   i == l, -h[i] ar[k, j] + h[k] ar[i, j],
   True, 0
  ];
b[0, y_] = 0; b[x_, 0] = 0;
b[x_Plus, y_] := PCollect[b[#, y] & /@ x];
b[x_, y_Plus] := PCollect[b[x, #] & /@ y];
b[c_ * x_ar, y_] := PCollect[c * b[x, y]];
b[x_, c_ * y_ar] := PCollect[c * b[x, y]];
Unprotect[NonCommutativeMultiply];
x_ ** y_ := BCH[b, x, y];
R[i_, j_] := h ar[i, j];
Unprotect[Up];
expr_ ~Up~ ind_List := Module[
   {ind1},
   ind1 = Flatten[{#}] & /@ ind;
   PCollect[expr /. {
      h[i_] :> Total[h /@ ind1[[i]]],
      ar[i_, j_] :> Total[Outer[ar, ind1[[i]], ind1[[j]]], 2]
     }]
  ];
R4Eqn[V_] := V ** (-R[1, 3] - R[2, 3]) - (-R[1, 3]) ** (-R[2, 3]) ** V;
TwistEqn[V_] := (-V) ** R[1, 2] ** (V ~Up~ {2, 1}) - (R[1, 2] / 2 + R[2, 1] / 2);
TwistMinusEqn[V_] := (-V) ** (-R[2, 1]) ** (V ~Up~ {2, 1}) + (R[1, 2] / 2 + R[2, 1] / 2);
Clear[Phi];
Phi[d_Integer, V_] := Phi[d, V] = PCollect[ToDegree[d,
     (-V) ~Up~ {{1, 2}, 3} ** (-V) ** V ~Up~ {2, 3} ** V ~Up~ {1, {2, 3}}
    ]];
Phi[V_] := Phi[$ToDegree, V];
```

```
Pentagon[Phi_] := Phi ** (Phi ~Up~ {1, {2, 3}, 4}) ** (Phi ~Up~ {2, 3, 4}) -
    (Phi ~Up~ {{1, 2}, 3, 4}) ** (Phi ~Up~ {1, 2, {3, 4}});
RR[i_, j_] := R[i, j] / 2 + R[j, i] / 2;
Hexagon[Phi_] := RR[1, 2] ~Up~ {{1, 2}, 3} -
    Phi ** RR[2, 3] ** ((-Phi) ~Up~ {1, 3, 2}) ** RR[1, 3] ** (Phi ~Up~ {3, 1, 2});
HexagonMinus[Phi_] := (-RR[1, 2]) ~Up~ {{1, 2}, 3} -
    Phi ** (-RR[2, 3]) ** ((-Phi) ~Up~ {1, 3, 2}) ** (-RR[1, 3]) ** (Phi ~Up~ {3, 1, 2});
uUnitarity[Phi_] := Phi ** (Phi ~Up~ {3, 2, 1});

Puncture[inds___Integer, expr_] :=
  expr /. Flatten[{h[#] → 0, ar[#, _] → 0} & /@ {inds}];

b[ar[1, 2], ar[1, 3]]
b[ar[1, 3], ar[2, 3]] /. h → c // MatrixForm
b[ar[1, 2], ar[2, 3]]
```

$$\begin{pmatrix} 0 \\ \text{ar}[2, 3]\, \text{c}[1] - \text{ar}[1, 3]\, \text{c}[2] \\ -\text{ar}[2, 3]\, \text{c}[1] + \text{ar}[1, 3]\, \text{c}[2] \end{pmatrix}$$