Implementing Classical g_0

```
(*Orig:  { X+1,  X+2,   X+3, H+1,   H+2,  H-2,  H-1,   X-3,  X-2,   X-1};
          {X,    Y,     Z,   A,     B,    b,    a,     z,    y,     x};
trivial cobracket, W is by definition the dual of w.
  lowercase letters represent the lower Borel
 A,B central. *)
PBWBasis = {X, Y, Z, b, a, z, y, x};
PBWRule = {X → 1, Y → 2, Z → 3, b → 4, a → 5, z → 6, y → 7, x → 8};
Br[U@X, U@Y] = 0;
Br[U@X, U@Z] = 0;
Br[U@X, U@b] = U@X;
Br[U@X, U@a] = -2 U@X;
Br[U@X, U@z] = 0;
Br[U@X, U@y] = 0;
Br[U@X, U@x] = (2 A - B) U[];
Br[U@Y, U@Z] = 0;
Br[U@Y, U@b] = -2 U@Y;
Br[U@Y, U@a] = U@Y;
Br[U@Y, U@z] = 0;
Br[U@Y, U@y] = (-A + 2 B) U[];
Br[U@Y, U@x] = 0;
Br[U@Z, U@b] = -U@Z;
Br[U@Z, U@a] = -U@Z;
Br[U@Z, U@z] = (A + B) U[];
Br[U@Z, U@y] = -U@X;
Br[U@Z, U@x] = U@Y;
Br[U@b, U@a] = 0;
Br[U@b, U@z] = -U@z;
Br[U@b, U@y] = -2 U@y;
Br[U@b, U@x] = U@x;
Br[U@a, U@z] = -U@z;
Br[U@a, U@y] = U@y;
Br[U@a, U@x] = -2 U@x;
Br[U@z, U@y] = 0;
Br[U@z, U@x] = 0;
Br[U@y, U@x] = -U@z;
```

co-bracket:

δ(X) =X^(2A-B) (wedge)

δ(Y) =Y^(2B-A)

δ(Z) = Z^(A+B)+X^Y

```
CheckJac[a_, b_, c_] := Br[U@a, Br[U@b, U@c]] + Br[U@b, Br[U@c, U@a]] + Br[U@c, Br[U@a, U@b]] // Simp

CheckJac[x, y, a]
```

```
0
```

```
x_ ≤ y_ := OrderedQ[{x, y} /. PBWRule];
x_ ≺ y_ := ! OrderedQ[{y, x} /. PBWRule];
Simp[ℰ_] := Collect[ℰ, _U, Expand];
```

```
U_i_[ℰ_] := ℰ /. {A → A_i, s → s_i, B → B_i, t → t_i, u_U :> Replace[u, x_ :> x_i, 1]};
Br[U[(x_)_i_], U[(y_)_i_]] := Br[U[x_i], U[y_i]] = U_i[Br[U@x, U@y]];
Br[U[(x_)_i_], U[(y_)_j_]] /; i =!= j := 0;
Br[x_, x_] = 0;
Br[U[y_], U[x_]] := Br[U[y], U[x]] = Simp[-Br[U[x], U[y]]];
Br[x_, y_] := x ** y - y ** x;
```

```
Unprotect[NonCommutativeMultiply];
NonCommutativeMultiply[x_] := x;
0 ** _ = _ ** 0 = 0;
x_ ** U[] := x; U[] ** x_ := x;
(a_ * x_U) ** (b_ * y_U) := If[a b === 0, 0, Simp[a b (x ** y)]];
(a_ * x_U) ** y_ := Simp[a (x ** y)]; x_ ** (a_ * y_U) := Simp[a (x ** y)];
(x_Plus) ** y_ := (# ** y) & /@ x; x_ ** (y_Plus) := (x ** #) & /@ y;
```

```
U[xx___, x_] ** U[y_, yy___] := If[x ≤ y, U[xx, x, y, yy], U@xx ** (U@y ** U@x + Br[U@x, U@y]) ** U@yy];
```

```
UU[l___, x_^n_, r___] := UU[l, Sequence @@ Table[x, {n}], r];
UU[l___, 1, r___] := UU[l, r];
UU[] = U[];
UU[l_, r___] := U[l] ** UU[r];
```