11a. Routine for products in the enveloping algebra

The routine **R** can evaluate differentiation by higher degree elements of U(**g**), but this runs slowly. Moreover if **Rna**(**HHr**) is substituted as differentiation unexpected results follow. It is better to make a modified routine **eR** that at each step uses simplifying substitution rules instead only applying the substitution at the end.

In[ ]:=
```
Clear[eR, eR0]
eR[aa_, ff_, sub_] :=
 eR0[Expand[aa //. XWtoZsub /. HHi → -(1/2) CKi + (3/2) WW0], Expand[ff], sub] // Simplify
eR0[aa_ + bb_, ff_, sub_] := eR0[aa, ff, sub] + eR0[bb, ff, sub]
eR0[pp_ aa_, ff_, sub_] := pp eR0[aa, ff, sub] /; FreeLie[pp]
eR0[nul, ff_, sub_] := 0
eR0[nul aa_, ff_, sub_] := 0
eR0[xx_, ff_, sub_] := (R[xx, ff] // sub // mult) /; inlist[xx, Zlist]
eR0[aa_ ** bb_, ff_, sub_] := Module[{x}, x = R[bb, ff] // sub // mult;
   eR0[aa, x, sub] // mult] /; FreeQ[bb, NonCommutativeMultiply ]
```

The following routine is useful for a sum over r in which various terms with Φ[h,p,r',q] occur. The routine compr carries out the substitution r->2r- r' in each term with Φ[h,p,r',q].

In[ ]:=
```
Clear[compr, cpro, cpr, h, p, r]
cpro[a_ + b_] := cpro[a] + cpro[b]
cpr[y_ Phi[hh_, pp_, rr_, qq_]] := (y /. r → 2 r - rr) Phi[hh, pp, r, qq];
compr[xx_] := Module[{x}, x = cpro[Expand[xx]];
  x = x /. cpro[y_] :→ cpr[y];
  x // Simplify]
```

In[ ]:=

Checks

Check by comparison with R

In[ ]:=
```
Clear[chbt, f, h, p, r]
F = chbt f[t] × Phi[h, p, r, p]
Do[Print[Zlist[[j]], "   ",
  eR[Zlist[[j]], F, subab] == R[Zlist[[j]], F] // subab // Simplify], {j, 1, 8}]
```

Out[ ]=
```
chbt f[t] × Phi[h, p, r, p]
```

```
Z13   True
Z23   True
Z12   True
CKi   True
WW0   True
Z21   True
Z32   True
Z31   True
```

In[∘]:=
```
Clear[tht, f, h, p, r, m, eqn]
F = tht[m] × f[t] × Phi[h, p, r, p]
Do[eqn = eR[Zlist⟦j⟧, F, subnab] == R[Zlist⟦j⟧, F] // subnab // Simplify;
 Print[Zlist⟦j⟧, "  eps=", epps, "  ", eqn /. eps → epps // Simplify],
 {j, 1, 8}, {epps, 1, -1, -2}]
```

Out[∘]=
```
f[t] × Phi[h, p, r, p] × tht[m]
```

```
Z13   eps=1   True
Z13   eps=-1   True
Z23   eps=1   True
Z23   eps=-1   True
Z12   eps=1   True
Z12   eps=-1   True
CKi   eps=1   True
CKi   eps=-1   True
WW0   eps=1   True
WW0   eps=-1   True
Z21   eps=1   True
Z21   eps=-1   True
Z32   eps=1   True
Z32   eps=-1   True
Z31   eps=1   True
Z31   eps=-1   True
```

Comparison with description of downward shift operators

```
In[ ]:= Clear[chbt, f, h, p, r]
        F = chbt f[t] × Phi[h, p, r, p]
        sh[3, -1, F, subab]
        % == eR[Z32 + (2 (p + 1))^(-1) Z12 ** Z31, F, subab] // Simplify
```

Out[ ]= chbt f[t] × Phi[h, p, r, p]

Out[ ]= $\frac{1}{4\times(1+p)}$ chbt p (f[t] (4 i betac π t Phi[3 + h, -1 + p, -1 + r, -1 + p] + (-4 + h - 2 p - r) Phi[3 + h, -1 + p, 1 + r, -1 + p]) + 2 t Phi[3 + h, -1 + p, 1 + r, -1 + p] f′[t])

Out[ ]= True

```
In[ ]:= sh[-3, -1, F, subab]
        % == eR[Z13 - (2 (p + 1))^(-1) Z12 ** Z23, F, subab] // Simplify
```

Out[ ]= $-\frac{1}{4\times(1+p)}$ chbt p (f[t] ((4 + h + 2 p - r) Phi[-3 + h, -1 + p, -1 + r, -1 + p] + 4 i beta π t Phi[-3 + h, -1 + p, 1 + r, -1 + p]) - 2 t Phi[-3 + h, -1 + p, -1 + r, -1 + p] f′[t])

Out[ ]= True

```
In[ ]:= Clear[tht, f, h, p, r, m]
        F = tht[m] × f[t] × Phi[h, p, r, p]
        sh[3, -1, F, subnab]
        % == eR[Z32 + (2 (p + 1))^(-1) Z12 ** Z31, F, subnab] // Simplify
```

Out[ ]= f[t] × Phi[h, p, r, p] × tht[m]

Out[ ]= $-\frac{1}{4\times(1+p)}$ p (4 i eps $\sqrt{2\pi}$ t $\sqrt{\text{Abs[ell]}}$ $\sqrt{\text{m + dtt[-eps]}}$ f[t] × Phi[3 + h, -1 + p, -1 + r, -1 + p] × tht[-eps + m] + Phi[3 + h, -1 + p, 1 + r, -1 + p] × tht[m] ((4 - h + 2 p + r - 4 ell π t^2) f[t] - 2 t f′[t]))

Out[ ]= True

```
In[ ]:= sh[-3, -1, F, subnab]
        % == eR[Z13 - (2 (p + 1))^(-1) Z12 ** Z23, F, subnab] // Simplify
```

Out[ ]= $\frac{1}{4\times(1+p)}$ p (4 i eps $\sqrt{2\pi}$ t $\sqrt{\text{Abs[ell]}}$ $\sqrt{\text{m + dtt[eps]}}$ f[t] × Phi[-3 + h, -1 + p, 1 + r, -1 + p] × tht[eps + m] - Phi[-3 + h, -1 + p, -1 + r, -1 + p] × tht[m] ((4 + h + 2 p - r + 4 ell π t^2) f[t] - 2 t f′[t]))

Out[ ]= True

Casimir operator of K

```
In[ ]:= Clear[h, p, r, q]
        F = Phi[h, p, r, q]
        eR[WW0 ** WW0 - 2 I WW0 + Z12 ** Z21, F, ident] / F // Simplify

Out[ ]= Phi[h, p, r, q]

Out[ ]= - p (2 + p)
```