16 Evaluation at zero

See §3.5.3: (3.98) and Proposition 3.38

We assume that the component $f_r(t)$ is already written as $t^{2+\nu}\, h_r(t)$.

The operator **ev0** is evaluation at 0. The routine **ev** does the actual work: preserves addition and multiplication, equals one on the functions on N, and preserves $t^{2+\nu}$

In[ ]:=
```
Clear[ev, ev0, h]
ev0[ff_] := ev[Expand[ff]]
ev[ff_ + gg_] := ev[ff] + ev[gg]
ev[ff_ gg_] := ev[ff] × ev[gg]
ev[h[r_, t_]] := h[r, 0]
ev[t ^ (nu + cc_)] := 0 /; cc ≥ 3
ev[t ^ (2 + nu)] := t ^ (2 + nu)
ev[chbt] := 1
ev[tht[mm_]] := 1
ev[ff_] := ff /; FreeQ[ff, t] && FreeQ[ff, chbt] && FreeQ[ff, tht]
```

Check of shift operators in the abelian case

In[ ]:=
```
Clear[h, p, r]
F = chbt t ^ (2 + nu) h[r, t] × Phi[h, p, r, p]
```

Out[ ]= chbt t^{2+nu} h[r, t] × Phi[h, p, r, p]

In[ ]:=
```
ev0[sh[3, 1, F, subab]] ==
  sh[3, 1, ev0[F], subtriv] // Simplify
ev0[sh[3, -1, F, subab]] ==
  sh[3, -1, ev0[F], subtriv] // Simplify
ev0[sh[-3, 1, F, subab]] ==
  sh[-3, 1, ev0[F], subtriv] // Simplify
ev0[sh[-3, -1, F, subab]] ==
  sh[-3, -1, ev0[F], subtriv] // Simplify
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Check on shift operators in the non-abelian case

In[ ]:=
```
Clear[h, p, r, m]
F = tht[m[h, r]] t^(2 + nu) h[r, t] × Phi[h, p, r, p]
```

Out[ ]= t^{2+nu} h[r, t] × Phi[h, p, r, p] × tht[m[h, r]]

In[ ]:=
```
ev0[sh[3, 1, F, subnab]] ==
  sh[3, 1, ev0[F], subtriv] // Simplify
ev0[sh[3, -1, F, subnab]] ==
  sh[3, -1, ev0[F], subtriv] // Simplify
ev0[sh[-3, 1, F, subnab]] ==
  sh[-3, 1, ev0[F], subtriv] // Simplify
ev0[sh[-3, -1, F, subnab]] ==
  sh[-3, -1, ev0[F], subtriv] // Simplify
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True