Pensieve header: Implementing and testing the category of Gaussian Differential Operators.

Our morphisms are objects like $\mathbb{E}[Q, P]$ where Q is a benign quadratic and P is a docile series. No effort is made to encapsulate the domains and ranges.

Goal. Implement and verify $\text{Compose}_{\{ts\},\{as\},\{xs\}}[\mathcal{E}1\_\mathbb{E},\ \mathcal{E}2\_\mathbb{E}]$.

```
In[]:= {t*, y*, a*, x*, z*} = {τ, η, α, ξ, ζ};
       {τ*, η*, α*, ξ*, ζ*} = {t, y, a, x, z}; (u_i_)* := (u*)_i;
```

```
In[]:= {x*, α3*, z2*}
```

```
In[]:= Kδ /: Kδ_{i_,j_} := If[i === j, 1, 0];
```

Zip

```
In[]:= Zip_{}[P_] := P; Zip_{ζ_,ζs___}[P_] := (Expand[P // Zip_{ζs}] /. f_. ζ^d_. :> ∂_{ζ*,d} f) /. ζ* → 0
```

```
In[]:= Zip_{ξ}[(ξ^2 + ξ + 3) (x^5 e^x + 7 x) + 99 a]
```

```
In[]:= Zip_{η2}[e^(δ x y2) ξ η2]
```

```
In[]:= Zip_{ξ,η2}[(ξ^6 + ξ + 3 + 2 ξ η2) (x^5 e^(b x) + 7 x) + 99 a + e^(δ x y2) ξ η2]
```

```
In[]:= E0 = 𝔼[Sum[a_{10 i+j} x_i ξ_j, {i, 3}, {j, 3}],
         1 + ϵ Sum[f_i[x1, x2, x3] ξ_i, {i, 3}] + ϵ Sum[f_{10 i+j}[x1, x2, x3] ξ_i ξ_j, {i, 3}, {j, 3}]]
```

```
𝔼 /: Zip_{ζs_List}@𝔼[Q_, P_] := Module[{ζ, z, zs, c, ys, ηs, qt, zrule, Q1, Q2},
   zs = Table[ζ*, {ζ, ζs}];
   c = Q /. Alternatives @@ (ζs ⋃ zs) → 0;
   ys = Table[∂_ζ (Q /. Alternatives @@ zs → 0), {ζ, ζs}];
   ηs = Table[∂_z (Q /. Alternatives @@ ζs → 0), {z, zs}];
   qt = Inverse@Table[Kδ_{z,ζ*} - ∂_{z,ζ} Q, {ζ, ζs}, {z, zs}];
   zrule = Thread[zs → qt.(zs + ys)];
   Q1 = c + ηs.zs /. zrule;
   Q2 = Q1 /. Alternatives @@ zs → 0;
   Simplify /@ 𝔼[Q2, Det[qt] e^-Q2 Zip_{ζs}[e^Q1 (P /. zrule)]] ];
```

```
Zip_{ξ1,ξ2}@E0
```

```
In[]:= lhs = Zip_{ξ1,ξ2}@E0
```

```
In[]:= rhs12 = Zip_{ξ1}@Zip_{ξ2}@E0
```

```
In[]:= lhs == rhs12
```

```
In[]:= rhs21 = Zip_{ξ2}@Zip_{ξ1}@E0
```

```
In[]:= rhs12 == rhs21
```

```
In[]:= Eh = 𝔼[h Sum[a_{10 i+j} x_i ξ_j, {i, 3}, {j, 3}],
         1 + ϵ Sum[f_i[x1, x2, x3] ξ_i, {i, 3}] + ϵ Sum[f_{10 i+j}[x1, x2, x3] ξ_i ξ_j, {i, 3}, {j, 3}]]
```

```
In[∘]:= lhs = Normal[Eh /. 𝔼[Q_, P_] ⧴ Series[P e^Q, {h, 0, 1}]] // Zip_{ξ1,ξ2}

In[∘]:= rhs0 = Zip_{ξ1,ξ2}[Eh];
        rhs1 = Normal[rhs0 /. 𝔼[Q_, P_] ⧴ Series[P e^Q, {h, 0, 1}]]

In[∘]:= Simplify[lhs == rhs1]
```

Bind

E

```
In[∘]:= 𝔼 /: 𝔼[Q1_, P1_] 𝔼[Q2_, P2_] := 𝔼[Q1 + Q2, P1 * P2];

In[∘]:= Bind_{ξs_List}[L_𝔼, R_𝔼] := Module[{n, hideξs, hidezs},
          hideξs = Table[ξs[[i]] → ξ_{n@i}, {i, Length@ξs}];
          hidezs = Table[ξs[[i]]* → z_{n@i}, {i, Length@ξs}];
          Zip_{ξs/.hideξs}[(L /. hidezs) (R /. hideξs)] ];

In[∘]:= Bind_{ξ2}[𝔼[ξ (x1 + x2), 1], 𝔼[ξ2 (x2 + x3), 1]]

Out[∘]= 𝔼[ξ (x1 + x2 + x3), 1]

In[∘]:= Bind_{ξ2}[𝔼[(ξ2 + ξ3) x2, 1], 𝔼[(ξ1 + ξ2) x, 1]]

Out[∘]= 𝔼[x (ξ1 + ξ2 + ξ3), 1]

In[∘]:= Bind_{ξ1,ξ2}[𝔼[(ξ2 + ξ3) x2 + ξ1 x1, 1], 𝔼[(ξ1 + ξ2) x, 1]]

Out[∘]= 𝔼[x (ξ1 + ξ2 + ξ3), 1]
```