Pensieve header: A fresh implementation of baby DoPeGDO. Continues pensieve://2020-09/, pensieve://2020-03/Testing123.nb, and pensieve://People/VanDerVeen/TimidHeisenbergRGeneralForm@.nb.

𝔼[ω, Q, P_ϵSeries] represents ωe^{Q+P}, where ω is a scalar, Q is an ϵ-free quadratic, and $P=\sum_{k=0}^{\$k} P[\![k]\!]\,\epsilon^k$ is a perturbation (it is ill-advised to include ω in P because then it will have log terms).

Scheme: 𝔼_[_] // 𝔼_[_] calls FZip or Zip, which are functionally the same. Zip works by handling the quadratic part and calling PZip for the perturbation-only part. PZip works by iteratively solving the synthesis equation. FZip works by encapsulating coefficients, calling Zip, and back-substituting.

Minor utilities

```
In[∘]:=
CCF[ℰ_] := PP_CCF@ExpandDenominator@ExpandNumerator@Together[ℰ];
(*CoefficientCanonical Form *)
CF[ℰ_List] := CF /@ ℰ;
CF[ℰ_ϵSeries] := CF /@ ℰ;
CF[ℰ_] := PP_CF@Module[
    {vs = Cases[ℰ, (y | x | η | ξ)_, ∞] ⋃ {y | x | η | ξ}},
    Total[(CCF[#[[2]]] (Times @@ vs^#[[1]])) & /@ CoefficientRules[Expand[ℰ], vs]]
   ];
(*CF[ℰ_] := PP_CF@CCF[ℰ];*)
CF[ℰ_𝔼] := CF /@ ℰ;
CF[𝔼_sp___[ℰs___]] := CF /@ 𝔼_sp[ℰs];
```

```
ϵSeries /: S1_ϵSeries ≡ S2_ϵSeries :=
  Length[S1] == Length[S2] ∧ Inner[CF[#1] == CF[#2] &, S1, S2, And];
ϵSeries[0] /; $k > 0 := ϵSeries @@ Table[0, $k + 1];
ϵSeries /: S1_ϵSeries + S2_ϵSeries :=
  ϵSeries @@ Table[S1[[k]] + S2[[k]], {k, Min[Length@S1, Length@S2]}];
ϵSeries /: S1_ϵSeries * S2_ϵSeries := ϵSeries @@
   Table[Sum[S1[[j + 1]] * S2[[k - j + 1]], {j, 0, k}], {k, 0, Min[Length@S1, Length@S2] - 1}];
ϵSeries /: c_ * S_ϵSeries := (c #) & /@ S;
ϵSeries /: ∂_vs___ S_ϵSeries := (s ↦ ∂_vs s) /@ S;
```

The Main Program

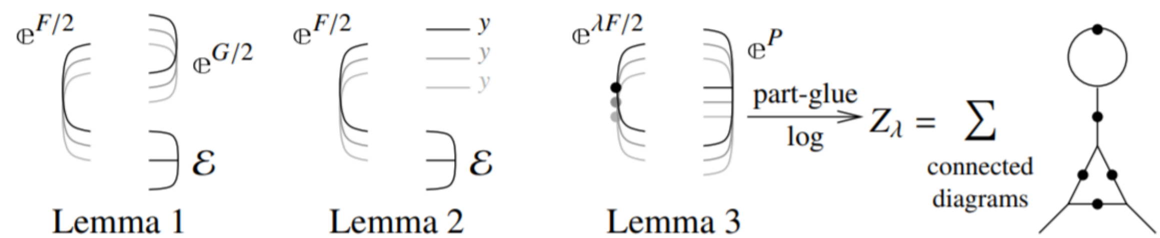
Variables and their duals:

```
In[∘]:=
{y*, x*, η*, ξ*} = {η, ξ, y, x};
(vs_List)* := (v ↦ v*) /@ vs;
(u_i_)* := (u*)_i;
```

𝔼 operations:

In[∘]:=
```
𝔼 /: 𝔼[ω1_, Q1_, P1_] ≡ 𝔼[ω2_, Q2_, P2_] := CF[ω1 == ω2] ∧ CF[Q1 == Q2] ∧ (P1 ≡ P2);
𝔼 /: 𝔼[ω1_, Q1_, P1_] 𝔼[ω2_, Q2_, P2_] := 𝔼[ω1 ω2, Q1 + Q2, P1 + P2];
𝔼_{d1_→r1_}[ℰ1s___] ≡ 𝔼_{d2_→r2_}[ℰ2s___] ^:= (d1 == d2) ∧ (r1 == r2) ∧ (𝔼[ℰ1s] ≡ 𝔼[ℰ2s]);
𝔼_{d1_→r1_}[ℰ1s___] 𝔼_{d2_→r2_}[ℰ2s___] ^:= 𝔼_{(d1⋃d2)→(r1⋃r2)} @@ (𝔼[ℰ1s] 𝔼[ℰ2s]);
𝔼_{dr_}[ℰs___]_{$k_} := 𝔼_{dr} @@ 𝔼[ℰs]_{$k};
```

In[∘]:=
```
𝔼_{d1_→r1_}[ℰ1s___] // 𝔼_{d2_→r2_}[ℰ2s___] := Module[{is = r1 ⋂ d2, lvs},
  lvs = Flatten@Table[{x_{$@i}, y_{$@i}}, {i, is}];
  𝔼_{(d1⋃Complement[d2,is])→(r2⋃Complement[r1,is])} @@ (Zip_{lvs⋃lvs*}[lvs*.lvs, Times[
      𝔼[ℰ1s] /. Table[(v : x | y)_i → v_{$@i}, {i, is}],
      𝔼[ℰ2s] /. Table[(v : ξ | η)_i → v_{$@i}, {i, is}]
    ]])
 ]
```



In[∘]:=
```
Zip_{vs_}[ℱ_, ℰ_] := ⟨ℱ, ℰ⟩ // Zip1_{vs} // Zip2_{vs} // Zip3_{vs};
Zip_{vs_}[ℱ_, ℰ_] := ⟨ℱ, ℰ⟩ // Zip1_{vs} // EZip23_{vs};
```

Getting rid of the quadratic.

Lemma 1. With convergences left to the reader,

$$\left\langle F:\, \mathcal{E}\,\mathbb{e}^{\frac{1}{2}\sum_{i,j\in B} G_{ij}z_i z_j}\right\rangle_B = \det(1-GF)^{-1/2}\left\langle F(1-GF)^{-1}:\,\mathcal{E}\right\rangle_B$$

In[∘]:=
```
Zip1_{} = Identity;
Zip1_{vs_}@⟨ℱ_, 𝔼[ω_, Q_, P_]⟩ := PP_{Zip1}@Module[{ℐ, F, G, u, v},
   ℐ = IdentityMatrix@Length@vs;
   F = Table[∂_{u,v}ℱ, {u, vs*}, {v, vs*}];
   G = Table[∂_{u,v}Q, {u, vs}, {v, vs}];
   ⟨CF[vs*.F.Inverse[ℐ - G.F].vs* / 2],
    𝔼[CF@PowerExpand@Factor[ω Det[ℐ - G.F]^{-1/2}], CF[Q - vs.G.vs / 2], P]⟩
  ]
```

Getting rid of linear terms.

Lemma 2. $\left\langle F:\, \mathcal{E}\,\mathbb{e}^{\sum_{i\in B} y_i z_i}\right\rangle_B = \mathbb{e}^{\frac{1}{2}\sum_{i,j\in B} F_{ij}y_i y_j}\left\langle F:\; \mathcal{E}|_{z_B\to z_B+Fy_B}\right\rangle_B.$

In[∘]:=
```
Zip2{} = Identity;
Zip2vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PPZip2@Module[{F, Y, u, v},
   F = Table[∂u,v 𝓕, {u, vs*}, {v, vs*}];
   Y = Table[∂v Q, {v, vs}];
   CF /@ ⟨𝓕, 𝔼[ω, Q - Y.vs + Y.F.Y / 2, P /. Thread[vs → vs + F.Y]]⟩
  ]
```

Dealing with Feynman diagrams.

Lemma 3. With an extra variable λ, $Z_\lambda := \log[\lambda F : \mathbb{e}^P]_B$ satisfies and is determined by the following PDE / IVP:

$$Z_0 = P \quad \text{and} \quad \partial_\lambda Z_\lambda = \frac{1}{2}\sum_{i,j\in B} F_{ij}\left(\partial_{z_i}\partial_{z_j}Z_\lambda + (\partial_{z_i}Z_\lambda)(\partial_{z_j}Z_\lambda)\right).$$

Note that the power m of λ is at most $k-1+\frac{2k+2}{2}=2k$. We write $Z_\lambda = \sum Z[m]\,\lambda^m$.

In[∘]:=
```
Zip3vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PPZip3@Module[{Z, u, v, m, j},
   Z[0] = P;
   For[m = 0, m < 2 $k, ++m,
    Z[m + 1] = CF[1/(2 (m + 1))
       Sum[∂u*,v* 𝓕 (∂u,v Z[m] + Sum[(∂u Z[j]) (∂v Z[m - j]), {j, 0, m}]), {u, vs}, {v, vs}]]
   ];
   𝔼[ω, Q, CF[Sum[Z[m], {m, 0, 2 $k}] /. Table[v → 0, {v, vs}]]]
  ]
```

```
EZip23vs_ @⟨𝓕_, 𝔼[ω_, Q_, P_]⟩ := PPEZip23@Module[
   {nP, n𝓕, nQ, j = 0, ps, c, t, rr = {(*release rules*)}},
   nP = Total[
       CoefficientRules[#, vs] /.
        (ps_ → c_) :→ (AppendTo[rr, t[++j] → CF@c]; t[j] (Times @@ vs^ps))
      ] & /@ P;
   nQ = Total[CoefficientRules[Q, vs] /.
      (ps_ → c_) :→ (AppendTo[rr, t[++j] → CF@c]; t[j] (Times @@ vs^ps))];
   n𝓕 = Total[CoefficientRules[𝓕, vs*] /. (ps_ → c_) :→
       (AppendTo[rr, t[++j] → CF@c]; t[j] (Times @@ (vs*)^ps))];
   CF[Expand[⟨n𝓕, 𝔼[ω, nQ, nP]⟩ // Zip2vs // Zip3vs] /. rr]
  ]
```