```
QZipζs_List@𝔼[L_, Q_, P_] :=
  PPQZip@Module[{ζ, z, zs, c, ys, ηs, qt, zrule, ζrule, out},
    zs = Table[ζ*, {ζ, ζs}];
    c = CF[Q /. Alternatives @@ (ζs ⋃ zs) → 0];
    ys = CF@Table[∂ζ (Q /. Alternatives @@ zs → 0),
       {ζ, ζs}];
    ηs = CF@Table[∂z (Q /. Alternatives @@ ζs → 0), {z, zs}];
    qt = CF@Inverse@Table[Kδz,ζ* - ∂z,ζQ, {ζ, ζs}, {z, zs}];
    zrule = Thread[zs → CF[qt.(zs + ys)]];
    ζrule = Thread[ζs → ζs + ηs.qt];
    CF /@ 𝔼[L, c + ηs.qt.ys,
      Det[qt] Zipζs[P /. (zrule ⋃ ζrule)]] ];
```

LZip implements the "L-level zips" on $\mathbb{E}(L, Q, P) = P e^{L+Q}$. Such zips regard all of $P e^{Q}$ as a single"P". Here the z's are b and α and the ζ's are β and a.

```
LZipζs_List@𝔼[L_, Q_, P_] :=
  PPLZip@Module[{ζ, z, zs, Zs, c, ys, ηs, lt, zrule,
     Zrule, ζrule, Q1, EEQ, EQ},
    zs = Table[ζ*, {ζ, ζs}];
    Zs = zs /. {b → B, t → T, α → 𝒜};
    c = L /. Alternatives @@ (ζs ⋃ zs) → 0 /.
      Alternatives @@ Zs → 1;
    ys = Table[∂ζ (L /. Alternatives @@ zs → 0), {ζ, ζs}];
    ηs = Table[∂z (L /. Alternatives @@ ζs → 0), {z, zs}];
    lt = Inverse@Table[Kδz,ζ* - ∂z,ζL, {ζ, ζs}, {z, zs}];
    zrule = Thread[zs → lt.(zs + ys)];
    Zrule = Join[zrule,
      zrule /.
       r_Rule :> ((U = r〚1〛 /. {b → B, t → T, α → 𝒜}) →
          (U /. U2l /. r //. l2U))];
    ζrule = Thread[ζs → ζs + ηs.lt];
    Q1 = Q /. (Zrule ⋃ ζrule);
    EEQ[ps___] :=
     EEQ[ps] =
      PP"EEQ"@(CF[e^-Q1 DThread[{zs,{ps}}][e^Q1]] /.
         {Alternatives @@ zs → 0, Alternatives @@ Zs → 1});
    CF@𝔼[c + ηs.lt.ys,
      Q1 /. {Alternatives @@ zs → 0, Alternatives @@ Zs → 1},
      Det[lt]
       (Zipζs[(EQ @@ zs) (P /. (Zrule ⋃ ζrule))] /.
          Derivative[ps___][EQ][___] :> EEQ[ps] /.
         _EQ → 1) ] ];
```

```
B{}[L_, R_] := L R;
B{is__}[L_𝔼, R_𝔼] := PPB@Module[{n},
    Times[
      L /. Table[(v : b | B | t | T | a | x | y)i → vn@i,
        {i, {is}}],
      R /. Table[(v : β | τ | α | 𝒜 | ξ | η)i → vn@i, {i, {is}}]
     ] // LZipJoin@@Table[{βn@i,τn@i,an@i},{i,{is}}] //
    QZipJoin@@Table[{ξn@i,yn@i},{i,{is}}] ];
Bis__[L_, R_] := B{is}[L, R];
```

𝔼 morphisms with domain and range.

```
Bis_List[𝔼d1_→r1_[L1_, Q1_, P1_], 𝔼d2_→r2_[L2_, Q2_, P2_]] :=
  𝔼(d1⋃Complement[d2,is])→(r2⋃Complement[r1,is]) @@
   Bis[𝔼[L1, Q1, P1], 𝔼[L2, Q2, P2]];
𝔼d1_→r1_[L1_, Q1_, P1_] // 𝔼d2_→r2_[L2_, Q2_, P2_] :=
  Br1⋂d2[𝔼d1→r1[L1, Q1, P1], 𝔼d2→r2[L2, Q2, P2]];
𝔼d1_→r1_[L1_, Q1_, P1_] ≡ 𝔼d2_→r2_[L2_, Q2_, P2_] ^:=
  (d1 == d2) ∧ (r1 == r2) ∧ (𝔼[L1, Q1, P1] ≡ 𝔼[L2, Q2, P2]);
𝔼d1_→r1_[L1_, Q1_, P1_] 𝔼d2_→r2_[L2_, Q2_, P2_] ^:=
  𝔼(d1⋃d2)→(r1⋃r2) @@ (𝔼[L1, Q1, P1] × 𝔼[L2, Q2, P2]);
𝔼dr_[L_, Q_, P_]$k_ := 𝔼dr @@ 𝔼[L, Q, P]$k;
𝔼_[ℰ___][i_] := {ℰ}〚i〛;
```

𝔼[Λ]

```
𝔼dr_[Λ_] :=
 CF@Module[{L, Λ0 = Limit[Λ, ϵ → 0]},
   𝔼dr[L = Λ0 /. (η | y | ξ | x)_ → 0, Λ0 - L, e^(Λ-Λ0)]$k /. l2U]
```

Exponentials as needed.

Task. Define $\mathrm{Exp}_{m,i,k}[P]$ to compute $e^{\mathbb{O}(P)}$ to ϵ^k in the using the $m_{i,j\to i}$ multiplication, where P is an ϵ-dependent near-docile element, giving the answer in $\mathbb{E}$-form.

Methodology. If $P_0 := P_{\epsilon=0}$ and $e^{\lambda\,\mathbb{O}(P)} = \mathbb{O}\left(e^{\lambda P_0} F(\lambda)\right)$, then $F(\lambda=0)=1$ and we have:

$$\mathbb{O}\left(e^{\lambda P_0}(P_0 F(\lambda) + \partial_\lambda F)\right) = \mathbb{O}\left(\partial_\lambda e^{\lambda P_0} F(\lambda)\right) =$$

$$\partial_\lambda \mathbb{O}\left(e^{\lambda P_0} F(\lambda)\right) = \partial_\lambda e^{\lambda\,\mathbb{O}(P)} = e^{\lambda\,\mathbb{O}(P)}\,\mathbb{O}(P) = \mathbb{O}\left(e^{\lambda P_0} F(\lambda)\right)\mathbb{O}(P)$$

This is a linear ODE for F. Setting inductively $F_k = F_{k-1} + \epsilon^k \varphi$ we find that $F_0 = 1$ and solve for φ.

```
(* Bug: The first line is valid only if 𝕆(e^P0) == e^𝕆(P0). *)
Expm_,i_,0[P_] := Module[{LQ = Normal@P /. ϵ → 0},
   𝔼[LQ /. (x | y)i → 0, LQ /. (b | a | t)i → 0, 1] ];
Expm_,i_,k_[P_] := Block[{$k = k},
  Module[{P0, λ, φ, φs, F, j, rhs, eqn, pows, at0, atλ},
   P0 = Normal@P /. ϵ → 0;
   F = Normal@Last@Expm,i,k-1[λ P];
   While[
    rhs =
     mi,j→i[
       𝔼{}→{i}[λ P0 /. (x | y)i → 0, λ P0 /. (b | a | t)i → 0,
         F]k sσi→j@𝔼{}→{i}[0, 0, P]k] // Last // Normal;
    eqn = CF[(∂λF) + P0 F - rhs];
    eqn =!= 0, (*do*)
    pows = First /@ CoefficientRules[eqn, {yi, bi, ai, xi}];
    F += Sum[ϵ^k φjs[λ] Times @@ {yi, bi, ai, xi}^js,
      {js, pows}];
    rhs =
     mi,j→i[
       𝔼{}→{i}[λ P0 /. (x | y)i → 0, λ P0 /. (b | a | t)i → 0,
         F]k sσi→j@𝔼{}→{i}[0, 0, P]k] // Last // Normal;
    eqn = CF[(∂λF) + P0 F - rhs];
    φs = Table[φjs[λ], {js, pows}];
    at0 = Table[φjs[0] == 0, {js, pows}];
    atλ = (# == 0) & /@
      (pows /. CoefficientRules[eqn, {yi, bi, ai, xi}]);
    F = F /. DSolve[And @@ (at0 ⋃ atλ), φs, λ]〚1〛
   ];
   𝔼{}→{i}[P0 /. (x | y)i → 0, P0 /. (b | a | t)i → 0,
    F + O[ϵ]^(k+1) /. λ → 1]] ]
```

"Define" Code

Video and more: http://www.math.toronto.edu/~drorbn/Talks/CRM-1907, http://www.math.toronto.edu/~drorbn/Talks/UCLA-191101.