

```

QZip $\zeta_s$ _List@E[L_, Q_, P_] :=
  PPQZip@Module[{ $\xi$ , z, zs, c, ys,  $\eta_s$ , qt, zrule,  $\xi$ rule, out},
    zs = Table[ $\xi^*$ , { $\xi$ ,  $\xi_s$ }];
    c = CF[Q /. Alternatives@@ ( $\xi_s \cup zs$ )  $\rightarrow$  0];
    ys = CF@Table[ $\partial_{\xi}$ (Q /. Alternatives@@ zs  $\rightarrow$  0),
      { $\xi$ ,  $\xi_s$ };
     $\eta_s$  = CF@Table[ $\partial_z$ (Q /. Alternatives@@  $\xi_s \rightarrow$  0), {z, zs}];
    qt = CF@Inverse@Table[K $\delta_{z,\xi^*} - \partial_{z,\xi}Q$ , { $\xi$ ,  $\xi_s$ }, {z, zs}];
    zrule = Thread[zs  $\rightarrow$  CF[qt.(zs + ys)]];
     $\xi$ rule = Thread[ $\xi_s \rightarrow \xi_s + \eta_s.qt$ ];
    CF /@ E[L, c +  $\eta_s.qt.ys$ ,
      Det[qt] Zip $\zeta_s$ [P /. (zrule  $\cup$   $\xi$ rule)]];

```

LZip implements the “L-level zips” on $E(L, Q, P) = Pe^{L+Q}$. Such zips regard all of Pe^Q as a single “P”. Here the z’s are b and α and the ξ ’s are β and a .

```

LZip $\zeta_s$ _List@E[L_, Q_, P_] :=
  PPLZip@Module[{ $\xi$ , z, zs, Zs, c, ys,  $\eta_s$ , lt, zrule,
    Zrule,  $\xi$ rule, Q1, EEQ, EQ},
    zs = Table[ $\xi^*$ , { $\xi$ ,  $\xi_s$ }];
    Zs = zs /. {b  $\rightarrow$  B, t  $\rightarrow$  T,  $\alpha \rightarrow$  A};
    c = L /. Alternatives@@ ( $\xi_s \cup Zs$ )  $\rightarrow$  0 /.
      Alternatives@@ Zs  $\rightarrow$  1;
    ys = Table[ $\partial_{\xi}$ (L /. Alternatives@@ zs  $\rightarrow$  0), { $\xi$ ,  $\xi_s$ };
     $\eta_s$  = Table[ $\partial_z$ (L /. Alternatives@@  $\xi_s \rightarrow$  0), {z, zs}];
    lt = Inverse@Table[K $\delta_{z,\xi^*} - \partial_{z,\xi}L$ , { $\xi$ ,  $\xi_s$ }, {z, zs}];
    zrule = Thread[zs  $\rightarrow$  lt.(zs + ys)];
    Zrule = Join[zrule,
      zrule /.
        r_Rule  $\Rightarrow$  ((U = r[[1]] /. {b  $\rightarrow$  B, t  $\rightarrow$  T,  $\alpha \rightarrow$  A})  $\rightarrow$ 
          (U /. U21 /. r /. 12U));
     $\xi$ rule = Thread[ $\xi_s \rightarrow \xi_s + \eta_s.lt$ ];
    Q1 = Q /. (Zrule  $\cup$   $\xi$ rule);
    EEQ[ps___] :=
      EEQ[ps] =
        PPEEQ@(CF[e-Q1 DThread[{zs, {ps}}] [eQ1]] /.
          {Alternatives@@ zs  $\rightarrow$  0, Alternatives@@ Zs  $\rightarrow$  1});
    CF@E[c +  $\eta_s.lt.ys$ ,
      Q1 /. {Alternatives@@ zs  $\rightarrow$  0, Alternatives@@ Zs  $\rightarrow$  1},
      Det[lt]
        (Zip $\zeta_s$ [(EQ@@ zs) (P /. (Zrule  $\cup$   $\xi$ rule))] /.
          Derivative[ps___][EQ][___]  $\Rightarrow$  EEQ[ps] /.
            _EQ  $\rightarrow$  1) ]];

```

```

B_{i} [L_, R_] := LR;
B_{is___} [L_E, R_E] := PPB@Module[{n},
  Times[
    L /. Table[{v : b | B | t | T | a | x | y}_i  $\rightarrow$  vnei,
      {i, {is}}],
    R /. Table[{v :  $\beta$  |  $\tau$  |  $\alpha$  | A |  $\xi$  |  $\eta$ }_i  $\rightarrow$  vnei, {i, {is}}]
  ] // LZipJoinTable[{ $\beta_{nei}, \tau_{nei}, a_{nei}$ }, {i, {is}}] //
    QZipJoinTable[{ $\xi_{nei}, \eta_{nei}$ }, {i, {is}}];
B_{is___} [L_, R_] := B_{is} [L, R];

```

E morphisms with domain and range.

```

B_{is_List} [Ed1 $\rightarrow$ r1 [L1_, Q1_, P1_], Ed2 $\rightarrow$ r2 [L2_, Q2_, P2_]] :=
  E (d1  $\cup$  Complement[d2, is])  $\rightarrow$  (r2  $\cup$  Complement[r1, is]) @@
  B_{is} [E [L1, Q1, P1], E [L2, Q2, P2]];
Ed1 $\rightarrow$ r1 [L1_, Q1_, P1_] // Ed2 $\rightarrow$ r2 [L2_, Q2_, P2_] :=
  Br1  $\cap$  d2 [Ed1 $\rightarrow$ r1 [L1, Q1, P1], Ed2 $\rightarrow$ r2 [L2, Q2, P2]];
Ed1 $\rightarrow$ r1 [L1_, Q1_, P1_]  $\equiv$  Ed2 $\rightarrow$ r2 [L2_, Q2_, P2_]  $\wedge$  :=
  (d1 == d2)  $\wedge$  (r1 == r2)  $\wedge$  (E [L1, Q1, P1]  $\equiv$  E [L2, Q2, P2]);
Ed1 $\rightarrow$ r1 [L1_, Q1_, P1_] Ed2 $\rightarrow$ r2 [L2_, Q2_, P2_]  $\wedge$  :=
  E (d1  $\cup$  d2)  $\rightarrow$  (r1  $\cup$  r2) @@ (E [L1, Q1, P1]  $\times$  E [L2, Q2, P2]);
Edr [L_, Q_, P_]  $\xi_k$  := Edr @@ E [L, Q, P]  $\xi_k$ ;
E_ [E___] [i_] := {E} [i];

```

E[\wedge]

```

Edr [A_] :=
  CF@Module[{L,  $\Delta$ 0 = Limit[A,  $\epsilon \rightarrow$  0]},
    Edr [L =  $\Delta$ 0 /. ( $\eta$  | y |  $\xi$  | x)  $\rightarrow$  0,  $\Delta$ 0 - L, eA- $\Delta$ 0]  $\xi_k$  /. 12U]

```

Exponentials as needed.

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

Methodology. If $P_0 := P_{\epsilon=0}$ and $e^{\mathcal{O}(P)} = \mathcal{O}(e^{\lambda P_0} F(\lambda))$, then

$F(\lambda = 0) = 1$ and we have:

$$\mathcal{O}(e^{\lambda P_0}(P_0 F(\lambda) + \partial_{\lambda} F)) = \mathcal{O}(\partial_{\lambda} e^{\lambda P_0} F(\lambda)) =$$

$$\partial_{\lambda} \mathcal{O}(e^{\lambda P_0} F(\lambda)) = \partial_{\lambda} e^{\mathcal{O}(P)} = e^{\mathcal{O}(P)} \mathcal{O}(P) = \mathcal{O}(e^{\lambda P_0} F(\lambda)) \mathcal{O}(P)$$

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 $\mathcal{O}(e^{P_0}) = e^{\mathcal{O}(P_0)}$.)

```

Expm,i,0 [P_] := Module[{LQ = Normal@P /.  $\epsilon \rightarrow$  0},
  E [LQ /. (x | y)_i  $\rightarrow$  0, LQ /. (b | a | t)_i  $\rightarrow$  0, 1]];
Expm,i,k [P_] := Block[{$k = k},
  Module[{P0,  $\lambda$ ,  $\varphi$ ,  $\varphi_s$ , F, j, rhs, eqn, pows, at0, at $\lambda$ },
    P0 = Normal@P /.  $\epsilon \rightarrow$  0;
    F = Normal@Last@Expm,i,k-1 [P];
    While[
      rhs =
        mi,j $\rightarrow$ i [
          Ei $\rightarrow$ {i} [  $\lambda$  P0 /. (x | y)_i  $\rightarrow$  0,  $\lambda$  P0 /. (b | a | t)_i  $\rightarrow$  0,
            F ]k S $\sigma$ i $\rightarrow$ j @ Ei $\rightarrow$ {i} [0, 0, P]k] // Last // Normal;
      eqn = CF [( $\partial_{\lambda}$  F) + P0 F - rhs];
      eqn != 0, (*do*)
      pows = First /@ CoefficientRules[eqn, {yi, bi, ai, xi}];
      F += Sum[eR  $\varphi_{js}$  [ $\lambda$ ] Times@@ {yi, bi, ai, xi}js,
        {js, pows}];
      rhs =
        mi,j $\rightarrow$ i [
          Ei $\rightarrow$ {i} [  $\lambda$  P0 /. (x | y)_i  $\rightarrow$  0,  $\lambda$  P0 /. (b | a | t)_i  $\rightarrow$  0,
            F ]k S $\sigma$ i $\rightarrow$ j @ Ei $\rightarrow$ {i} [0, 0, P]k] // Last // Normal;
      eqn = CF [( $\partial_{\lambda}$  F) + P0 F - rhs];
       $\varphi_s$  = Table[ $\varphi_{js}$  [ $\lambda$ ], {js, pows}];
      at0 = Table[ $\varphi_{js}$  [0] == 0, {js, pows}];
      at $\lambda$  = (# == 0) & /@
        (pows /. CoefficientRules[eqn, {yi, bi, ai, xi}]);
      F = F /. DSolve[And@@ (at0  $\cup$  at $\lambda$ ),  $\varphi_s$ ,  $\lambda$ ] [[1]]
    ];
    Ei $\rightarrow$ {i} [P0 /. (x | y)_i  $\rightarrow$  0, P0 /. (b | a | t)_i  $\rightarrow$  0,
      F + 0[ek+1 /.  $\lambda \rightarrow$  1]]];

```

“Define” Code