

```

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] := PP_B@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}}];];
    ] // LZipJoin@Table[{ $\beta_{nei}$ ,  $\tau_{nei}$ ,  $a_{nei}$ }, {i, {is}}] //
      QZipJoin@Table[{ $\xi_{nei}$ ,  $\eta_{nei}$ }, {i, {is}}];];
B_{is___} [L_, R_] := B_{is} [L, R];

```

E morphisms with domain and range.

```

B_{is_List} [E $d_1 \rightarrow r_1$  [L1_, Q1_, P1_], E $d_2 \rightarrow r_2$  [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]];
E $d_1 \rightarrow r_1$  [L1_, Q1_, P1_] // E $d_2 \rightarrow r_2$  [L2_, Q2_, P2_] :=
  B_{r1  $\cap$  d2} [E $d_1 \rightarrow r_1$  [L1, Q1, P1], E $d_2 \rightarrow r_2$  [L2, Q2, P2]];
E $d_1 \rightarrow r_1$  [L1_, Q1_, P1_]  $\equiv$  E $d_2 \rightarrow r_2$  [L2_, Q2_, P2_]  $\wedge$  :=
  (d1 == d2)  $\wedge$  (r1 == r2)  $\wedge$  (E[L1, Q1, P1]  $\equiv$  E[L2, Q2, P2]);
E $d_1 \rightarrow r_1$  [L1_, Q1_, P1_] E $d_2 \rightarrow r_2$  [L2_, Q2_, P2_]  $\wedge$  :=
  E (d1  $\cup$  d2)  $\rightarrow$  (r1  $\cup$  r2) @@ (E[L1, Q1, P1]  $\times$  E[L2, Q2, P2]);
E $d_r$  [L_, Q_, P_]  $\$k$  := E $d_r$  @@ E[L, Q, P]  $\$k$ ;
E_ [E___] [i_] := {E} [i];

```

E[\wedge]

```

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

```

Exponentials as needed.

Task. Define $\text{Exp}_{m,j,k}[P]$ to compute $e^{O(P)}$ 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 $e^{A O(P)} = O(e^{A P_0} F(\lambda))$, then

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

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

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

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

```

Exp $m, i, \theta$  [P_] := Module[{LQ = Normal@P /.  $\epsilon \rightarrow$  0},
  E[LQ /. (x | y)_i  $\rightarrow$  0, LQ /. (b | a | t)_i  $\rightarrow$  0, 1]];];
Exp $m, i, k$  [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@Exp $m, i, k-1$  [P];];
  While[
    rhs =
      m $i, j \rightarrow i$  [
        E[ $\rightarrow$  {i}] [ $\lambda$  P0 /. (x | y)_i  $\rightarrow$  0,  $\lambda$  P0 /. (b | a | t)_i  $\rightarrow$  0,
          F] $k$  S $\sigma$  i  $\rightarrow$  j @ E[ $\rightarrow$  {i}] [0, 0, P] $k$ ] // Last // Normal;];
    eqn = CF[( $\partial_\lambda$  F) + P0 F - rhs];];
    eqn != 0, (*do*)];
    pows = First /@ CoefficientRules[eqn, {y $_i$ , b $_i$ , a $_i$ , x $_i$ };];
    F += Sum[eR  $\varphi_{js}$  [ $\lambda$ ] Times@@ {y $_i$ , b $_i$ , a $_i$ , x $_i$ }js,
      {js, pows}];];
    rhs =
      m $i, j \rightarrow i$  [
        E[ $\rightarrow$  {i}] [ $\lambda$  P0 /. (x | y)_i  $\rightarrow$  0,  $\lambda$  P0 /. (b | a | t)_i  $\rightarrow$  0,
          F] $k$  S $\sigma$  i  $\rightarrow$  j @ E[ $\rightarrow$  {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, {y $_i$ , b $_i$ , a $_i$ , x $_i$ };]);];
    F = F /. DSolve[And@@ (at0  $\cup$  at $\lambda$ ),  $\varphi_s$ ,  $\lambda$ ] [[1]]];];
  E[ $\rightarrow$  {i}] [P0 /. (x | y)_i  $\rightarrow$  0, P0 /. (b | a | t)_i  $\rightarrow$  0,
    F + O[ $\epsilon$ ]k+1 /.  $\lambda \rightarrow$  1] ]];];

```

“Define” Code