```
LZip_ξs_List@𝔼[L_, Q_, P_] :=
  PP_LZip@Module[{ξ, z, zs, Zs, c, ys, ηs, lt, zrule,
     Zrule, ξrule, Q1, EEQ, EQ},
    zs = Table[ξ*, {ξ, ξs}];
    Zs = zs /. {b → B, t → T, α → 𝒜};
    c = L /. Alternatives @@ (ξs ⋃ zs) → 0;
    ys = Table[∂_ξ (L /. Alternatives @@ zs → 0), {ξ, ξs}];
    ηs = Table[∂_z (L /. Alternatives @@ ξs → 0), {z, zs}];
    lt = Inverse@Table[Kδ_{z,ξ*} - ∂_{z,ξ} L, {ξ, ξs}, {z, zs}];
    zrule = Thread[zs → lt.(zs + ys)];
    Zrule = zrule ~Join~
      (zrule /.
        r_Rule :↦ ((U = r⟦1⟧ /. {b → B, t → T, α → 𝒜}) →
           (U /. U2l /. r //. l2U)));
    ξrule = Thread[ξs → ξs + ηs.lt];
    Q1 = Q /. (Zrule ⋃ ξrule);
    EEQ[ps___] :=
     EEQ[ps] =
      (PP_"EEQ"@(CF[e^-Q1 D_Thread[{zs,{ps}}][e^Q1]] /.
          {Alternatives @@ zs → 0,
           Alternatives @@ Zs → 1}));
    CF /@ ((*CF/@*)𝔼[
       c + ηs.lt.ys,
       Q1 /. {Alternatives @@ zs → 0,
         Alternatives @@ Zs → 1},
       Det[lt]
        (Zip_ξs[(EQ @@ zs) (P /. (Zrule ⋃ ξrule))] /.
           Derivative[ps___][EQ][___] :↦ EEQ[ps] /.
           _EQ → 1)
      ])
    ];
```