# Cheat Sheet $sl_2$-Invariant (the $sl_2$ portfolio and invariant)



**Objects.** All are of the form $\mathbb{e}^{L+Q}P$, where

- $L$ is a quadratic of the form $\sum l_{z\zeta}z\zeta$, where $z$ runs over $\{t_i,\alpha_i\}_{i\in S}$ and $\zeta$ over $\{\tau_i,a_i\}_{i\in S}$, with integer coefficients $l_{z\zeta}$.
- $Q$ is a quadratic of the form $\sum q_{z\zeta}z\zeta$, where $z$ runs over $\{x_i,\eta_i\}_{i\in S}$ and $\zeta$ over $\{\xi_i,y_i\}_{i\in S}$, with coefficients $q_{z\zeta}$ in the ring $R_S$ of rational functions in $\{T_i,\mathcal{A}_i\}_{i\in S}$.
- $P=\sum\epsilon^kP_k$ is docile ($\deg P_k\leq 4k$) in $\{y_i,a_i,x_i,\eta_i,\xi_i\}_{i\in S}$ with coefficients in $R_S$, and where $\deg(y_i,a_i,x_i,\eta_i,\xi_i)=(1,2,1,1,1)$.

**In QuarksAndDegrees.m:** Gradings to remove $\hbar$ and $\gamma$.

**Q.** What becomes of the classical-level automorphism $(y,b,\epsilon)\to(-y,-b,-\epsilon)$?

## Internal Utilities

Canonical Form:

```
CCF[ℰ_] :=
  PP_CCF@ExpandDenominator@
    ExpandNumerator@PP_Together@Together[PP_Exp[
      Expand[ℰ] //. 𝕖^x_ 𝕖^y_ :> 𝕖^(x+y) /. 𝕖^x_ :> 𝕖^CCF[x]]];
CF[ℰ_List] := CF /@ ℰ;
CF[sd_SeriesData] := MapAt[CF, sd, 3];
CF[ℰ_] := PP_CF@Module[
    {vs = Cases[ℰ, (y | b | t | a | x | η | β | τ | α | ξ)_, ∞] ⋃
       {y, b, t, a, x, η, β, τ, α, ξ}},
    Total[CoefficientRules[Expand[ℰ], vs] /.
       (ps_ → c_) :> CCF[c] (Times @@ vs^ps)]
  ];
CF[ℰ_𝔼] := CF /@ ℰ;
CF[𝔼_sp___[ℰs___]] := CF /@ 𝔼_sp[ℰs];
```

The Kronecker $\delta$:

```
Kδ /: Kδ_{i_,j_} := If[i === j, 1, 0];
```

Equality, multiplication, and degree-adjustment of perturbed Gaussians; $\mathbb{E}[L,Q,P]$ stands for $\mathbb{e}^{L+Q}P$:

```
𝔼 /: 𝔼[L1_, Q1_, P1_] ≡ 𝔼[L2_, Q2_, P2_] :=
  CF[L1 == L2] ∧ CF[Q1 == Q2] ∧ CF[Normal[P1 - P2] == 0];
𝔼 /: 𝔼[L1_, Q1_, P1_] 𝔼[L2_, Q2_, P2_] :=
  𝔼[L1 + L2, Q1 + Q2, P1 * P2];
𝔼[L_, Q_, P_]_$k_ := 𝔼[L, Q, Series[Normal@P, {ϵ, 0, $k}]];
𝔼3@𝔼_sp___[ω_, L_, Q_, P_] := Module[
  {NP = Normal[P]},
  𝔼_sp[L, ω^-1 Q, (ω^-1 NP /. ϵ → ω^-4 ϵ) + O[ϵ]^($k+1)] // CF
 ];
𝔼4@𝔼_sp___[L_, Q_, P_] := Module[
  {NP = Normal[P], ω},
  ω = (NP /. ϵ → 0)^-1;
  𝔼_sp[ω, L, ω Q, (ω NP /. ϵ → ω^4 ϵ) + O[ϵ]^($k+1)] // CF
 ];
```

## Zip and Bind

Variables and their duals:

```
{t*, b*, y*, a*, x*, z*} = {τ, β, η, α, ξ, ζ};
{τ*, β*, η*, α*, ξ*, ζ*} = {t, b, y, a, x, z};
(u_i_)* := (u*)_i;
```

Finite Zips:

```
collect[sd_SeriesData, ζ_] :=
  MapAt[collect[#, ζ] &, sd, 3];
collect[ℰ_, ζ_] := PP_Collect@Collect[ℰ, ζ];
Zip_{}[P_] := P; Zip_{ζ_,ζs___}[P_] := PP_Zip[
  (collect[P // Zip_{ζs}, ζ] /. f_. ζ^d_. :> ∂_{ζ*,d} f) /. ζ* → 0]
```

QZip implements the “$Q$-level zips” on $\mathbb{E}(L,Q,P)=P\mathbb{e}^{L+Q}$. Such zips regard the $L$ variables as scalars.

```
QZip_ζs_List@𝔼[L_, Q_, P_] :=
  PP_QZip@Module[{ζ, z, zs, c, ys, ηs, qt, zrule, ζrule},
    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[Kδ_{z,ζ*} - ∂_{z,ζ}Q, {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] Zip_ζs[P /. (zrule ⋃ ζrule)]] ];
```

Upper to lower and lower to Upper:

```
U2l = {B_i_^p_. → 𝕖^(-p ħ γ b_i), B^p_. → 𝕖^(-p ħ γ b), T_i_^p_. → 𝕖^(p ħ t_i),
   T^p_. → 𝕖^(p ħ t), 𝒜_i_^p_. → 𝕖^(p γ α_i), 𝒜^p_. → 𝕖^(p γ α)};
l2U = {𝕖^(c_. b_i_ + d_.) :> B_i^(-c/(ħ γ)) 𝕖^d, 𝕖^(c_. b + d_.) :> B^(-c/(ħ γ)) 𝕖^d,
   𝕖^(c_. t_i_ + d_.) :> T_i^(c/ħ) 𝕖^d, 𝕖^(c_. t + d_.) :> T^(c/ħ) 𝕖^d,
   𝕖^(c_. α_i_ + d_.) :> 𝒜_i^(c/γ) 𝕖^d, 𝕖^(c_. α + d_.) :> 𝒜^(c/γ) 𝕖^d,
   𝕖^ℰ_ :> 𝕖^(Expand@ℰ)};
```

LZip implements the “$L$-level zips” on $\mathbb{E}(L,Q,P)=P\mathbb{e}^{L+Q}$. Such zips regard all of $P\mathbb{e}^Q$ as a single”$P$”. Here the $z$’s are $b$ and $\alpha$ and the $\zeta$’s are $\beta$ and $a$.

**The Zipping Theorem.** If $P$ has a finite $\zeta$-degree,

$$\left\langle P(z_i,\zeta^j)\mathbb{e}^{c+\eta^iz_i+y_j\zeta^j+q^i_jz_i\zeta^j}\right\rangle_{(\zeta^j)}=|\tilde{q}|\mathbb{e}^{c+\eta^i\tilde{q}^k_iy_k}\left\langle P\left(\tilde{q}^k_i(z_k+y_k),\zeta^j+\eta^i\tilde{q}^j_i\right)\right\rangle_{(\zeta^j)}.$$

where $\tilde{q}$ is the inverse matrix of $1-q$: $(\delta^i_j-q^i_j)\tilde{q}^j_k=\delta^i_k$.

```
LZip_ζs_List@𝔼[L_, Q_, P_] :=
  PP_LZip@Module[{ζ, z, zs, c, ys, ηs, lt, zrule, Zrule,
     ζrule, Q1, EEQ, EQ},
    zs = Table[ζ*, {ζ, ζs}];
    c = L /. Alternatives @@ (ζs ⋃ zs) → 0;
    ys = Table[∂_ζ (L /. Alternatives @@ zs → 0), {ζ, ζs}];
    ηs = Table[∂_z (L /. Alternatives @@ ζs → 0), {z, zs}];
    lt = Inverse@Table[Kδ_{z,ζ*} - ∂_{z,ζ}L, {ζ, ζs}, {z, zs}];
    zrule = Thread[zs → lt.(zs + ys)];
    Zrule =
     zrule /.
      r_Rule :> ((U = r[[1]] /. {b → B, t → T, α → 𝒜}) →
         (U /. U2l /. r //. l2U)); (* not used *)
    ζrule = Thread[ζs → ζs + ηs.lt];
    Q1 = Q /. U2l /. (zrule ⋃ ζrule);
    EEQ[ps___] :=
     EEQ[ps] =
      PP_"EEQ"@
       (CF[𝕖^-Q1 D[𝕖^Q1, Sequence @@ Thread[{zs, {ps}}]]] /.
         Alternatives @@ zs → 0 //. l2U);
    CF /@ ((*CF/@*)𝔼[
       c + ηs.lt.ys, Q1 /. Alternatives @@ zs → 0,
       Det[lt]
        (Zip_ζs[(EQ @@ zs) (P /. U2l /. (zrule ⋃ ζrule))] /.
          Derivative[ps___][EQ][___] :> EEQ[ps] /.
          _EQ → 1)
       ] //. l2U)
  ];
```

```
B{}[L_, R_] := L R;
B{is__}[L_𝔼, R_𝔼] := PP_B@Module[{n},
   Times[
     L /. Table[(ν : b | B | t | T | a | x | y)_i → ν_{n@i},
       {i, {is}}],
     R /. Table[(ν : β | τ | α | 𝒜 | ξ | η)_i → ν_{n@i}, {i, {is}}]
   ] // LZip_{Join@@Table[{β_{n@i}, τ_{n@i}, a_{n@i}}, {i, {is}}]} //
  QZip_{Join@@Table[{ξ_{n@i}, y_{n@i}}, {i, {is}}]} ];
B_{is___}[L_, R_] := B_{{is}}[L, R];
```

## 𝔼 morphisms with domain and range.

```
B_{is_List}[𝔼_{d1_→r1_}[L1_, Q1_, P1_], 𝔼_{d2_→r2_}[L2_, Q2_, P2_]] :=
  𝔼_{(d1⋃Complement[d2,is])→(r2⋃Complement[r1,is])} @@
   B_{is}[𝔼[L1, Q1, P1], 𝔼[L2, Q2, P2]];
𝔼_{d1_→r1_}[L1_, Q1_, P1_] // 𝔼_{d2_→r2_}[L2_, Q2_, P2_] :=
  B_{r1⋂d2}[𝔼_{d1→r1}[L1, Q1, P1], 𝔼_{d2→r2}[L2, Q2, P2]];
𝔼_{d1_→r1_}[L1_, Q1_, P1_] ≡ 𝔼_{d2_→r2_}[L2_, Q2_, P2_] ^:=
  (d1 == d2) ∧ (r1 == r2) ∧ (𝔼[L1, Q1, P1] ≡ 𝔼[L2, Q2, P2]);
𝔼_{d1_→r1_}[L1_, Q1_, P1_] 𝔼_{d2_→r2_}[L2_, Q2_, P2_] ^:=
  𝔼_{(d1⋃d2)→(r1⋃r2)} @@ (𝔼[L1, Q1, P1] 𝔼[L2, Q2, P2]);
𝔼_{d_→r_}[L_, Q_, P_]_{$k_} := 𝔼_{d→r} @@ 𝔼[L, Q, P]_{$k};
𝔼_[ℰ___][i_] := {ℰ}〚i〛;
```

## "Define" code

Define[lhs = rhs, ...] defines the lhs to be rhs, except that rhs is computed only once for each value of $k. Fancy Mathematica not for the faint of heart. Most readers should ignore.

```
SetAttributes[Define, HoldAll];
Define[def_, defs__] := (Define[def]; Define[defs];);
Define[op_{is__} = ℰ_] :=
 Module[{SD, ii, jj, kk, isp, nis, nisp, sis},
  Block[{i, j, k},
   ReleaseHold[Hold[
      SD[op_{nisp,$k_Integer}, PP_Boot@Block[{i, j, k}, op_{isp,$k} = ℰ;
         op_{nis,$k}]];
      SD[op_{isp}, op_{{is},$k}]; SD[op_{sis__}, op_{{sis}}];
     ] /. {SD → SetDelayed,
      isp → {is} /. {i → i_, j → j_, k → k_},
      nis → {is} /. {i → ii, j → jj, k → kk},
      nisp → {is} /. {i → ii_, j → jj_, k → kk_}
     }] ]]
```

## Booting Up

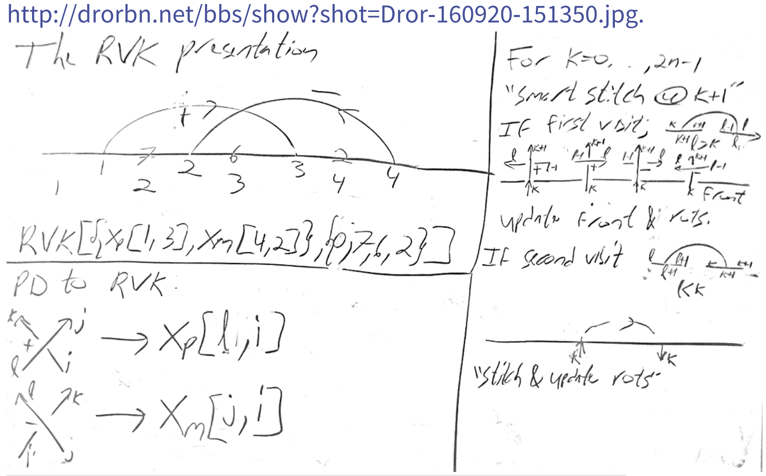
```
$k = 2; (*ℏ=γ=1;*)
Define[am_{i,j→k} = 𝔼_{{i,j}→{k}}[(α_i + α_j) a_k, (e^{-γ α_j} ξ_i + ξ_j) x_k, 1]_{$k},
 bm_{i,j→k} = 𝔼_{{i,j}→{k}}[(β_i + β_j) b_k, (η_i + η_j) y_k, e^{(e^{-ϵ β_i}-1) η_j y_k}]_{$k}]
Define[
 R_{i,j} =
  CF@
   𝔼_{{}→{i,j}}[ℏ a_j b_i, ℏ x_j y_i, e^(Sum_{k=2}^{$k+1} (1 - e^{γ ϵ ℏ})^k (ℏ y_i x_j)^k / (k (1 - e^{k γ ϵ ℏ})))]_{$k},
 R̄_{i,j} = CF@𝔼_{{}→{i,j}}[-ℏ a_j b_i, -ℏ x_j y_i / B_i,
    1 + If[$k == 0, 0, (R̄_{{i,j},$k-1})_{$k}[3] -
       (((R̄_{{i,j},0})_{$k} R_{1,2} (R̄_{{3,4},$k-1})_{$k}) // (bm_{i,1→i} am_{j,2→j}) //
          (bm_{i,3→i} am_{j,4→j}))[3]]],
 P_{i,j} = 𝔼_{{i,j}→{}}[β_i α_j / ℏ, η_i ξ_j / ℏ,
   1 + If[$k == 0, 0, (P_{{i,j},$k-1})_{$k}[3] -
      (R_{1,2} // ((P_{{1,j},0})_{$k} (P_{{i,2},$k-1})_{$k}))[3]]]]
```

```
Define[aS_j = R̄_{i,j} ~ B_i ~ P_{i,j},
 aS̄_i = 𝔼_{{i}→{i}}[-a_i α_i, -x_i 𝒜_i ξ_i,
   1 + If[$k == 0, 0, (aS̄_{{i},$k-1})_{$k}[3] -
      ((aS̄_{{i},0})_{$k} ~ B_i ~ aS_i ~ B_i ~ (aS̄_{{i},$k-1})_{$k})[3]]]]
Define[bS_i = R_{i,1} ~ B_1 ~ aS_1 ~ B_1 ~ P_{i,1},
 bS̄_i = R_{i,1} ~ B_1 ~ aS̄_1 ~ B_1 ~ P_{i,1},
 aΔ_{i→j,k} = (R_{1,j} R_{2,k}) // bm_{1,2→3} // P_{3,i},
 bΔ_{i→j,k} = (R_{j,1} R_{k,2}) // am_{1,2→3} // P_{i,3}]
Define[
 dm_{i,j→k} =
  (𝔼_{{i,j}→{i,j}}[β_i b_i + α_j a_j, η_i y_i + ξ_j x_j, 1]
     (aΔ_{i→1,2} // aΔ_{2→2,3} // aS̄_3) (bΔ_{j→-1,-2} // bΔ_{-2→-2,-3})) //
   (P_{-1,3} P_{-3,1} am_{2,j→k} bm_{i,-2→k}),
 dS_i = 𝔼_{{i}→{1,2}}[β_i b_1 + α_i a_2, η_i y_1 + ξ_i x_2, 1] // (bS̄_1 aS_2) //
   dm_{2,1→i},
 dΔ_{i→j,k} = (bΔ_{i→3,1} aΔ_{i→2,4}) // (dm_{3,4→k} dm_{1,2→j}) ]
Define[C_i = 𝔼_{{}→{i}}[0, 0, B_i^{1/2} e^{-ℏ ϵ a_i/2}]_{$k},
 C̄_i = 𝔼_{{}→{i}}[0, 0, B_i^{-1/2} e^{ℏ ϵ a_i/2}]_{$k},
 Kink_i = (R_{1,3} C̄_2) // dm_{1,2→1} // dm_{1,3→i},
 K̄ink_i = (R̄_{1,3} C_2) // dm_{1,2→1} // dm_{1,3→i}]
```

Note. $t == \epsilon a - \gamma b$ and $b == -t/\gamma + \epsilon a/\gamma$.

```
Define[
 b2t_i = 𝔼_{{i}→{i}}[α_i a_i - β_i t_i / γ, ξ_i x_i + η_i y_i, e^{ϵ β_i a_i/γ}]_{$k},
 t2b_i = 𝔼_{{i}→{i}}[α_i a_i - τ_i γ b_i, ξ_i x_i + η_i y_i, e^{ϵ τ_i a_i}]_{$k}]
Define[kR_{i,j} = R_{i,j} // (b2t_i b2t_j) /. t_{i|j} → t,
 kR̄_{i,j} = R̄_{i,j} // (b2t_i b2t_j) /. {t_{i|j} → t, T_{i|j} → T},
 km_{i,j→k} = (t2b_i t2b_j) // dm_{i,j→k} //
   b2t_k /. {t_k → t, T_k → T, τ_{i|j} → 0},
 kC_i = C_i // b2t_i /. T_i → T,
 kC̄_i = C̄_i // b2t_i /. T_i → T,
 kKink_i = Kink_i // b2t_i /. {t_i → t, T_i → T},
 kK̄ink_i = K̄ink_i // b2t_i /. {t_i → t, T_i → T}]
```

Some details of the code below are at http://drorbn.net/bbs/show?shot=Dror-160920-151350.jpg.




```
RVK::usage =
  "RVK[xs, rots] represents a Rotational Virtual
   Knot with a list of n Xp/Xm crossings xs and
   a length 2n list of rotation numbers rots.
   Crossing sites are indexed 1 through 2n, and
   rots〚k〛 is the rotation between site k-1 and
   site k. RVK is also a casting operator
   converting to the RVK presentation from other
   knot presentations.";
```

```
RVK[pd_PD] := PP_RVK@Module[{n, xs, x, rots, front = {0}, k},
  n = Length@pd; rots = Table[0, {2 n}];
  xs = Cases[pd, x_X :> { Xp[x[[4]], x[[1]]]   PositiveQ@x
                          Xm[x[[2]], x[[1]]]   True        ];
  For[k = 0, k < 2 n, ++k, If[k == 0 ∨ FreeQ[front, -k],
    front = Flatten[front /. k → (xs /. {
          Xp[k + 1, l_] | Xm[l_, k + 1] :> {l, k + 1, 1 - l},
          Xp[l_, k + 1] | Xm[k + 1, l_] :> (++rots[[l]];
            {1 - l, k + 1, l})
        })],
    Cases[front, k | -k] /. {k, -k} :> --rots[[k + 1]];
   ]];
  RVK[xs, rots] ];
RVK[K_] := RVK[PD[K]];
rot[i_, 0] := E_{{}→{i}}[0, 0, 1];
rot[i_, n_] := Module[{j},
  rot[i, n] = If[n > 0, rot[i, n - 1] kC_j, rot[i, n + 1] \overline{kC}_j] //
    km_{i,j→i}];
$k = 1; Timing@Z@Knot[10, 100]
```

$$\Big\{35.3125,\ \mathbb{E}_{\{\}\to\{0\}}\Big[0,\ 0,\ \frac{T^4}{1-4T+9T^2-12T^3+13T^4-12T^5+9T^6-4T^7+T^8}+\Big(T^4\hbar\,\big(4a\,(-2+14T-51T^2+120T^3-203T^4+258T^5-246T^6+152T^7-152T^9+246T^{10}-258T^{11}+203T^{12}-120T^{13}+51T^{14}-14T^{15}+2T^{16})+\gamma\,(-6+2T^{16}-8xy\hbar-440T^9(-1+xy\hbar)-4T^{15}(3+2xy\hbar)+8T^8(-97+21xy\hbar)+8T^7(131+21xy\hbar)-20T^6(57+22xy\hbar)+T^{14}(37+48xy\hbar)+T(44+48xy\hbar)-8T^{11}(2+61xy\hbar)+8T^5(127+68xy\hbar)-2T^{13}(35+78xy\hbar)+4T^{10}(-39+136xy\hbar)-T^2(167+156xy\hbar)+T^{12}(79+324xy\hbar)+T^3(410+324xy\hbar)-T^4(733+488xy\hbar))\big)\,\epsilon\Big)\Big/\big(1-4T+9T^2-12T^3+13T^4-12T^5+9T^6-4T^7+T^8\big)^3+O[\epsilon]^2\Big]\Big\}$$

```
PrintProfile[]
```

```
ProfileRoot is root. Profiled time: 77.548
  (     1)   0.204/ 36.984 above Z
  (   157)   0.420/ 32.503 above B
  (    37)   0.173/  7.984 above Boot
  (   147)   0.031/  0.061 above CF
  (     2)   0.016/  0.016 above RVK
CF: called 12199 times, time in 24.968/59.806
  (  1047)   0.973/  3.836 under EEQ
  (    47)   0.016/  0.063 under Boot
  (  1347)   6.357/ 18.406 under LZip
  (   147)   0.031/  0.061 under ProfileRoot
  (  9611)  17.591/ 37.440 under QZip
  ( 35642)  11.191/ 34.838 above CCF
Together: called 36776 times, time in 17.439/23.914
  ( 36776)  17.439/ 23.914 under CCF
  ( 36776)   5.755/  6.475 above Exp
CCF: called 36776 times, time in 11.644/35.558
  ( 35642)  11.191/ 34.838 under CF
  (  1134)   0.453/  0.720 under Exp
  ( 36776)  17.439/ 23.914 above Together
Zip: called 2675 times, time in 7.819/37.464
  (   294)   0.890/  6.074 under LZip
  (   294)   0.709/  3.749 under QZip
  (  2087)   6.220/ 27.641 under Zip
  (  2675)   2.004/  2.004 above Collect
  (  2087)   6.220/ 27.641 above Zip
Exp: called 36776 times, time in 5.755/6.475
  ( 36776)   5.755/  6.475 under Together
  (  1134)   0.453/  0.720 above CCF
LZip: called 294 times, time in 4.745/33.467
  (   294)   4.745/ 33.467 under B
  (  1047)   0.406/  4.242 above EEQ
  (  1347)   6.357/ 18.406 above CF
  (   294)   0.890/  6.074 above Zip
Collect: called 2675 times, time in 2.004/2.004
  (  2675)   2.004/  2.004 under Zip
QZip: called 294 times, time in 1.484/42.673
  (   294)   1.484/ 42.673 under B
  (  9611)  17.591/ 37.440 above CF
  (   294)   0.709/  3.749 above Zip
B: called 294 times, time in 0.641/76.781
  (    72)   0.063/ 36.686 under Z
  (    65)   0.158/  7.592 under Boot
  (   157)   0.420/ 32.503 under ProfileRoot
  (   294)   4.745/ 33.467 above LZip
  (   294)   1.484/ 42.673 above QZip
Boot: called 59 times, time in 0.423/12.515
  (     3)       0/  0.094 under Z
  (    19)   0.250/  4.437 under Boot
  (    37)   0.173/  7.984 under ProfileRoot
  (    65)   0.158/  7.592 above B
  (    19)   0.250/  4.437 above Boot
  (    47)   0.016/  0.063 above CF
EEQ: called 1047 times, time in 0.406/4.242
  (  1047)   0.406/  4.242 under LZip
  (  1047)   0.973/  3.836 above CF
Z: called 1 times, time in 0.204/36.984
  (     1)   0.204/ 36.984 under ProfileRoot
  (    72)   0.063/ 36.686 above B
  (     3)       0/  0.094 above Boot
RVK: called 2 times, time in 0.016/0.016
  (     2)   0.016/  0.016 under ProfileRoot
```