CS-SL2Invariant on 180614

June 14, 2018 11:28 AM

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

http://drorbn.net/AcademicPensieve/Projects/SL2Invariant/
modified 14/6/18, 11:24

Internal Utilities

Canonical Form:

```
CF[sd_SeriesData] := MapAt[CF, sd, 3];
CF[ℰ_] :=
  PP_CF@ExpandDenominator@
    ExpandNumerator@
     Together[Expand[ℰ] //. e^x_ e^y_ :→ e^(x+y) /. e^x_ :→ e^CF[x]];
```

The Kronecker δ:

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

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

```
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}]];
```

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: (* Perhaps switch Expand to Collect[__, ζ]? *)

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

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

```
QZip_{ζs_List,simp_}@𝔼[L_, Q_, P_] :=
  PP_QZip@Module[{ζ, z, zs, c, ys, ηs, qt, zrule, Q1, Q2},
    zs = Table[ζ*, {ζ, ζs}];
    c = Q /. Alternatives @@ (ζs ⋃ zs) → 0;
    ys = Table[∂_ζ (Q /. Alternatives @@ zs → 0), {ζ, ζs}];
    ηs = Table[∂_z (Q /. Alternatives @@ ζs → 0), {z, zs}];
    qt = Inverse@Table[Kδ_{z,ζ*} - ∂_{z,ζ}Q, {ζ, ζs}, {z, zs}];
    zrule = Thread[zs → qt.(zs + ys)];
    Q2 = (Q1 = c + ηs.zs /. zrule) /. Alternatives @@ zs → 0;
    simp /@ 𝔼[L, Q2, Det[qt] e^(-Q2) Zip_ζs[e^Q1 (P /. zrule)]] ];
QZip_{ζs_List} := QZip_{ζs,CF};
```

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

```
LZip_{ζs_List,simp_}@𝔼[L_, Q_, P_] :=
  PP_LZip@Module[{ζ, z, zs, c, ys, ηs, lt, zrule, L1, L2,
     Q1, Q2},
    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)];
    L2 = (L1 = c + ηs.zs /. zrule) /. Alternatives @@ zs → 0;
    Q2 = (Q1 = Q /. U2l /. zrule) /. Alternatives @@ zs → 0;
    simp /@
      𝔼[L2, Q2, Det[lt] e^(-L2-Q2)
        Zip_ζs[e^(L1+Q1) (P /. U2l /. zrule)]] //. l2U];
LZip_{ζs_List} := LZip_{ζs,CF};
Bind_{}[L_, R_] := L R;
Bind_{is__}[L_𝔼, R_𝔼] := PP_Bind@Module[{n},
    Times[
       L /. Table[(v : b | B | t | T | a | x | y)_i → v_{n@i},
         {i, {is}}],
       R /. Table[(v : β | τ | α | 𝒜 | ξ | η)_i → v_{n@i}, {i, {is}}]
      ] // LZip_{Flatten@Table[{β_{n@i}, τ_{n@i}, a_{n@i}}, {i, {is}}]} //
     QZip_{Flatten@Table[{ξ_{n@i}, y_{n@i}}, {i, {is}}]} ];
B_{l_List}[L_, R_] := Bind_l[L, R];
B_{is___}[L_, R_] := Bind_{is}[L, R];
```

"Define" code

Define[lhs = rhs] defines the lhs to be rhs, except that rhs is computed once and forever yet gets recomputed whenever $k changes. Fancy Mathematica not for the faint of heart. Most readers

```
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, ~~l, m, n~~}, cut
   ReleaseHold[Hold[
      SD[op_{nisp,$k_Integer},
       PP_{Boot@$k}@Block[{i, j, k, ~~l, m, n~~}, op_{isp,$k} = ℰ;
         op_{nis,$k}]];
      SD[op_isp, op_{{is},$k}];
      SD[op_{sis__}, op_{{sis}}];
     ] /. {
      isp → {is} /. {i → i_, j → j_, k → k_},
      nis → {is} /. {i → ii, j → jj, k → kk},
      nisp → {is} /. {i → ii_, j → jj_, k → kk_},
      SD → SetDelayed
     }]
 ]]
```

cut

Booting Up

```
$k = 2;
```

Split: R&P, a&b, d, KT, t

```
Define[
```

Timing@ } does it make sense to "Monitor" this?

```
PrintProfile[]
```

Split: R&P, a&b, d, KT, t

Timing@

} make sure this goes to "Monitor"

```
Define[
  am_{i,j→k} = 𝔼[(α_i + α_j) a_k, (e^{-γ α_j} ξ_i + ξ_j) x_k, 1]_{$k},
  bm_{i,j→k} = 𝔼[(β_i + β_j) b_k, (η_i + η_j) y_k, e^{(e^{-ε β_i} - 1) η_j y_k}]_{$k},
  R_{i,j} = 𝔼[ħ a_j b_i, ħ x_j y_i, e^( Σ_{k=2}^{$k+1} (1 - e^{γ ε ħ})^k (ħ y_i x_j)^k / (k (1 - e^{k γ ε ħ})) )]_{$k},
  P_{i,j} = If[$k == 0, 𝔼[β_i α_j / ħ, η_i ξ_j / ħ, 1]_0,
    MapAt[
      (# - ε^{$k} Coefficient[
          (R_{n,m} ~ B_{n,m} ~ ((P_{{n,j},0})_{$k} (P_{{i,m},$k-1})_{$k}))[[3]],
          ε, $k]) &, (P_{{i,j},$k-1})_{$k}, 3]],
  aS_i = 𝔼[-α_i a_j, -ξ_i x_i,
      e^{ξ_i x_i}
        Sum[Expand[(-ħ γ ε)^k / (2^k k!) Nest[Expand[x_i^2 ∂_{x_i,2} #] &,
          e^{-ξ_i e^{ħ ε a_i} x_i}, k]], {k, 0, $k}]]_{$k} ~ B_{i,j} ~ am_{i,j→i},
  \bar{aS}_i = If[$k == 0, 𝔼[-a_i α_i, -x_i 𝒜_i ξ_i, 1]_0,
    MapAt[
      (# - ε^{$k} Coefficient[
          ((\bar{aS}_{{i},0})_{$k} ~ B_i ~ aS_i ~ B_i ~ (\bar{aS}_{{i},$k-1})_{$k})[[3]],
          ε, $k]) &, (\bar{aS}_{{i},$k-1})_{$k}, 3]],
  bS_i = R_{i,n} ~ B_n ~ aS_n ~ B_n ~ P_{i,n},
  \bar{bS}_i = R_{i,n} ~ B_n ~ \bar{aS}_n ~ B_n ~ P_{i,n},
  aΔ_{i→j,k} = (R_{n,j} R_{m,k}) ~ B_{n,m} ~ bm_{n,m→1} ~ B_1 ~ P_{1,i},
  bΔ_{i→j,k} = (R_{j,n} R_{k,m}) ~ B_{n,m} ~ am_{n,m→1} ~ B_1 ~ P_{i,1},
  dm_{i,j→k} =
    (𝔼[β_i b_i + α_j a_j, η_i y_i + ξ_j x_j, 1] (aΔ_{i→1,2} ~ B_2 ~ aΔ_{2→2,3})
        (bΔ_{j→-1,-2} ~ B_{-2} ~ bΔ_{-2→-2,-3})) ~ B_3 ~ \bar{aS}_3 ~ B_{-1,3} ~ (P_{-1,3}) ~
    B_{-3,1} ~ (P_{-3,1}) ~ B_{2,j,i,-2} ~ (am_{2,j→k} bm_{i,-2→k}),
  dS_i = 𝔼[β_i b_n + α_i a_m, η_i y_n + ξ_i x_m, 1] ~ B_{n,m} ~ (\bar{bS}_n aS_m) ~
    B_{n,m} ~ dm_{m,n→i},
  dΔ_{i→j,k} = (bΔ_{i→3,1} aΔ_{i→2,4}) ~ B_{1,2,3,4} ~ (dm_{3,4→k} dm_{1,2→j}),
  \bar{R}_{i,j} = Expand /@ R_{i,j} ~ B_j ~ dS_j,
  CC_i = 𝔼[0, 0, B_i^{1/2} e^{-ħ ε a_i/2}]_{$k},
  \bar{CC}_i = 𝔼[0, 0, B_i^{-1/2} e^{ħ ε a_i/2}]_{$k},
  Kink_i = (R_{1,3} \bar{CC}_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→i},
  \bar{Kink}_i = (\bar{R}_{1,3} CC_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→i},
  (* t==εa-γb and b==-t/γ+εa/γ: *)
  b2t_i = 𝔼[α_i a_i - β_i t_i / γ, ξ_i x_i + η_i y_i, e^{ε β_i a_i/γ}]_{$k},
  t2b_i = 𝔼[α_i a_j - τ_i γ b_j, ξ_i x_j + η_i y_j, e^{ε τ_i a_j}]_{$k}
];
Monitor[Timing@Block[{$k = 1},
   Z = R_{1,5} R_{6,2} R_{3,7} \bar{CC}_4 \bar{Kink}_8 \bar{Kink}_9 \bar{Kink}_{10};
   Do[Z = Z ~ B_{1,r} ~ dm_{1,r→1}, {r, 2, 10}];
   Simplify /@ Z], r]
```

ab, R&P, aS, d, KT, t

$$\{112.813,\ \mathbb{E}\Big[0, 0, \frac{B_1}{1-B_1+B_1^2} - \frac{1}{(1-B_1+B_1^2)^3}\hbar B_1 \left(-a_1\left(-1+B_1-B_1^3+B_1^4\right) + \gamma\left(B_1 - 2B_1^2 - 2B_1^4 + 2\hbar x_1 y_1 + B_1^3(3+2\hbar x_1 y_1)\right)\right)\epsilon + O[\epsilon]^2\Big]\}$$

```
PrintProfile[]
Zip: called 1705 times, time in 228.214/759.584
  ( 206)     7.726/  51.122 under LZip
  ( 206)    25.823/ 177.092 under QZip
  ( 1293)  194.665/ 531.370 under Zip
  ( 1293)  194.665/ 531.370 above Zip
CF: called 31225 times, time in 72.464/78.916
  ( 29863)   6.452/   6.452 under CF
  ( 618)    51.508/  57.960 under LZip
  ( 126)         0/       0 under ProfileRoot
  ( 618)    14.504/  14.504 under QZip
  ( 29863)   6.452/   6.452 above CF
LZip: called 206 times, time in 17.418/126.5
  ( 206)    17.418/ 126.500 under Bind
  ( 618)    51.508/  57.960 above CF
  ( 206)     7.726/  51.122 above Zip
QZip: called 206 times, time in 4.031/195.627
  ( 206)     4.031/ 195.627 under Bind
  ( 618)    14.504/  14.504 above CF
  ( 206)    25.823/ 177.092 above Zip
Bind: called 206 times, time in 1.262/323.389
  ( 136)     0.949/ 274.936 under ProfileRoot
  ( 26)      0.062/   3.375 under Boot[1]
  ( 26)      0.047/  10.546 under Boot[2]
  ( 18)      0.204/  34.532 under Boot[3]
  ( 206)    17.418/ 126.500 above LZip
  ( 206)     4.031/ 195.627 above QZip
Boot[3]: called 11 times, time in 0.281/56.373
  ( 5)       0.064/  34.813 under ProfileRoot
  ( 6)       0.217/  21.560 under Boot[3]
  ( 18)      0.204/  34.532 above Bind
  ( 6)       0.217/  21.560 above Boot[3]
Boot[2]: called 18 times, time in 0.094/10.686
  ( 16)      0.079/  10.640 under ProfileRoot
  ( 2)       0.015/   0.046 under Boot[2]
  ( 26)      0.047/  10.546 above Bind
  ( 2)       0.015/   0.046 above Boot[2]
Boot[1]: called 20 times, time in 0.031/4.437
  ( 12)          0/   3.406 under ProfileRoot
  ( 8)       0.031/   1.031 under Boot[1]
  ( 26)      0.062/   3.375 above Bind
  ( 2)           0/       0 above Boot[0]
  ( 8)       0.031/   1.031 above Boot[1]
Boot[0]: called 2 times, time in 0./0.
  ( 2)           0/       0 under Boot[1]
ProfileRoot: called 0 times, time in 0./0.
  ( 136)     0.949/ 274.936 above Bind
  ( 126)         0/       0 above CF
  ( 12)          0/   3.406 above Boot[1]
  ( 16)      0.079/  10.640 above Boot[2]
  ( 5)       0.064/  34.813 above Boot[3]
```

Fix recursive counting

Fix