Cheat Sheet SL2Portfolio on 180331

March 31, 2018 2:33 PM

Cheat Sheet sl_2-Portfolio (an implementation of the sl_2 portfolio)

http://drorbn.net/AcademicPensieve/Projects/SL2Portfolio/
modified March 31, 2018, 14:32

$\mathcal{U}_{\gamma\epsilon;\hbar}$ conventions. "consolidate"

$q = e^{\hbar\gamma\epsilon}$, $H = \langle a, x\rangle / ([a,x] = \gamma x)$ with

$$A = e^{-\hbar\epsilon a}, \quad xA = qAx, \quad S_H(a, A, x) = (-a, A^{-1}, -A^{-1}x),$$

$$\Delta_H(a, A, x) = (a_1 + a_2, A_1A_2, x_1 + A_1x_2)$$

and dual $H^* = \langle b, y\rangle / ([b,y] = -\epsilon y)$ with

$$B = e^{-\hbar\gamma b}, \quad By = qyB, \quad S_{H^*}(b, B, y) = (-b, B^{-1}, -yB^{-1}),$$

$$\Delta_{H^*}(b, B, y) = (b_1 + b_2, B_1B_2, y_1B_2 + y_2).$$

Pairing by $(a,x)^* = \hbar(b,y)$ ($\Rightarrow \langle B, A\rangle = q$) making $\langle y^l b^i, a^j x^k\rangle = \delta_{ij}\delta_{kl}\hbar^{-(j+k)} j![k]_q!$ so $R = \sum \frac{\hbar^{j+k} y^k b^j \otimes a^j x^k}{j![k]_q!}$. Then $\mathcal{U} = H^{*cop} \otimes H$ with $(\phi f)(\psi g) = \langle \psi_1 S^{-1} f_3\rangle\langle\psi_3, f_1\rangle(\phi\psi_2)(f_2 g)$ and

$$S(y, b, a, x) = (-B^{-1}y, -b, -a, -A^{-1}x),$$

$$\Delta(y, b, a, x) = (y_1 + y_2B_1, b_1 + b_2, a_1 + a_2, x_1 + A_1x_2).$$

With the central $t := \epsilon a - \gamma b$, $T := e^{\hbar t} = A^{-1}B$ get

$$[a, y] = -\gamma y, \quad [b, x] = \epsilon x, \quad xy - qyx = (1 - TA^2)/\hbar.$$

Cartan: $\theta(y, b, a, x) = (-B^{-1}T^{1/2}x, -b, -a, -A^{-1}T^{-1/2}y)$. (Suggesting that it may be better to redefine $y \to y' = \theta x = A^{-1}T^{-1/2}y$.)

At $\epsilon = 0$, $\mathcal{U}_{\hbar;\gamma 0} = \langle t, y, a, x\rangle / ([t, \cdot] = 0, [a, x] = \gamma x, [a, y] = -\gamma y, [x, y] = (1 - T)/\hbar)$ with $\Delta(t, y, a, x) = (t_1 + t_2, y_1 + T_1y_2, a_1 + a_2, x_1 + x_2)$ and $\theta(y, b, a, x) = (-T^{-1/2}x, -b, -a, -T^{-1/2}y)$.

Working Hypothesis. $(\hbar, t, y, a, x)$ makes a PBW basis.

Casimir. $\omega = \gamma yx + \epsilon a^2 - (t - \gamma\epsilon)a$, satisfies.... Roland in MixOrder.pdf: Centrals are valuable; perhaps we should write everything in CU/QU as $(x \vee y)\cdot$(functions of a)$\cdot$(centrals).

Scaling with deg: $\{\gamma, \epsilon, a, b, x, y\} \to 1$, $\{\hbar\} \to -2$, $\{t\} \to 2$, $\{\omega\} \to 3$.

Verification (as in Projects/PPSA/Verification.nb).

```
DQ[ℰ_] :=
  (Exponent[Normal@ℰ /.
      {a → a/ϵ, a_i_ → a_i/ϵ, (u : x | y) :→ ϵ^(-1/2) u,
       (u : x | y)_i_ :→ ϵ^(-1/2) u_i}, ϵ, Min] ≥ 0);
$p = 2; $k = 1; $E := {$k, $p};
$trim := {ℏ^p_. /; p > $p → 0, ϵ^k_. /; k > $k → 0};
SetAttributes[{SS, SST}, HoldAll];
TRule = {T_i_ → e^(ℏ t_i), T → e^(ℏ t)}; q_ℏ = e^(γ ϵ ℏ);
SS[ℰ_, op_] := Collect[
   Normal@Series[If[$p > 0, ℰ, ℰ /. TRule], {ℏ, 0, $p}],
   ℏ, op];
SS[ℰ_] := SS[ℰ, Together];
SST[ℰ_, op___] := SS[ℰ /. TRule, op];
Simp[ℰ_, op_] := Collect[ℰ, _CU | _QU, op];
Simp[ℰ_] := Simp[ℰ, SS[#, Expand] &];
SimpT[ℰ_] := Collect[ℰ, _CU | _QU, SST[#, Expand] &];
DP_{α_→D_x_, β_→D_y_}[P_][λ_] :=
 Total[CoefficientRules[Normal@P, {α, β}] /.
   ({m_, n_} → c_) :→ c ∂_{{x,m},{y,n}} λ]
CF[ℰ_] := ExpandDenominator@
   ExpandNumerator@
    Together[Expand[ℰ] //. e^x_ e^y_ :→ e^(x+y) /. e^x_ :→ e^CF[x]];
```

```
Unprotect[SeriesData];
SeriesData /: CF[sd_SeriesData] := MapAt[CF, sd, 3];
SeriesData /: Expand[sd_SeriesData] :=
  MapAt[Expand, sd, 3];
SeriesData /: Simplify[sd_SeriesData] :=
  MapAt[Simplify, sd, 3];
SeriesData /: Together[sd_SeriesData] :=
  MapAt[Together, sd, 3];
SeriesData /: Collect[sd_SeriesData, specs__] :=
  MapAt[Collect[#, specs] &, sd, 3];
Protect[SeriesData];
SP_{}[P_] := P;
SP_{ξ_→x_, ps___}[P_] := Expand[P // SP_{ps}] /. f_. ξ^d_. :→ ∂_{{x,d}} f
DeclareAlgebra[CU, Generators → {y, a, x}, Centrals → {t}];
B[a_CU, y_CU] = -γ y_CU; B[x_CU, a_CU] = -γ x_CU;
B[x_CU, y_CU] = 2 ϵ a_CU - t 1_CU;
(S@y_CU = -y_CU; S@a_CU = -a_CU; S@x_CU = -x_CU;)
S_i_[CU, Centrals] = {t_i → -t_i};
Δ@y_CU = CU@y_1 + CU@y_2; Δ@a_CU = CU@a_1 + CU@a_2;
Δ@x_CU = CU@x_1 + CU@x_2;
Δ_{i_→j_,k_}[CU, Centrals] = {t_i → t_j + t_k};
DeclareAlgebra[QU, Generators → {y, a, x},
  Centrals → {t, T}];
B[a_QU, y_QU] = -γ y_QU; B[x_QU, a_QU] = -γ QU@x;
B[x_QU, y_QU] := SS[q_ℏ - 1] QU@{y, x} +
   𝕆_QU[{a}, SS[(1 - T e^(-2 ϵ a ℏ))/ℏ]];
(S@y_QU := 𝕆_QU[{a, y}, SS[-T^-1 e^(ℏ ϵ a) y]]; S@a_QU = -a_QU;
 S@x_QU := 𝕆_QU[{a, x}, SS[-e^(ℏ ϵ a) x]];)
S_i_[QU, Centrals] = {t_i → -t_i, T_i → T_i^-1};
Δ@y_QU := 𝕆_QU[{y_1, a_1}_1, {y_2}_2, SS[y_1 + T_1 e^(-ℏ ϵ a_1) y_2]];
Δ@a_QU = QU@a_1 + QU@a_2;
Δ@x_QU := 𝕆_QU[{a_1, x_1}_1, {x_2}_2, SS[x_1 + e^(-ℏ ϵ a_1) x_2]];
Δ_{i_→j_,k_}[QU, Centrals] = {t_i → t_j + t_k, T_i → T_j T_k};
DeclareMorphism[Cθ, CU → CU, {y → -x_CU, a → -a_CU, x → -y_CU},
  {t → -t, T → T^-1}];
DeclareMorphism[Qθ, QU → QU,
  {y :→ 𝕆_QU[{a, x}, SS[-T^(-1/2) e^(ℏ ϵ a) x]], a → -a_QU,
   x :→ 𝕆_QU[{a, y}, SS[-T^(-1/2) e^(ℏ ϵ a) y]]}, {t → -t, T → T^-1}]
```

$$\text{A}\mathbb{D}\$f = \gamma\,\frac{\mathrm{Cosh}\left[\hbar\left(a\epsilon + \frac{\gamma\epsilon}{2} - \frac{t}{2}\right)\right] - \mathrm{Cosh}\left[\hbar\sqrt{\left(\frac{t-\gamma\epsilon}{2}\right)^2 + \epsilon\omega}\right]}{\hbar\, e^{\hbar((a+\gamma)\epsilon - t/2)}\,\mathrm{Sinh}\left[\frac{\gamma\epsilon\hbar}{2}\right]\left(a^2\epsilon + a\gamma\epsilon - at - \omega\right)};$$

```
A𝔻$ω = γ CU[y, x] + ϵ CU[a, a] - (t - γ ϵ) CU[a];
DeclareMorphism[A𝔻, QU → CU,
 {a → a_CU, x → CU@x,
  y :→ 𝕊_CU[SS[A𝔻$f], a → a_CU, ω → A𝔻$ω] ** y_CU}]
```

$$\text{S}\mathbb{D}\$g = \sqrt{\frac{2\gamma\left(\mathrm{Cosh}\left[\frac{\hbar}{2}\sqrt{t^2 + \gamma^2\epsilon^2 + 4\epsilon\varpi}\right] - \mathrm{Cosh}\left[\frac{t - \epsilon\gamma - 2\epsilon a}{2/\hbar}\right]\right)}{\mathrm{Sinh}\left[\frac{\gamma\epsilon\hbar}{2}\right]\left(t(2a + \gamma) - 2a(a+\gamma)\epsilon + 2\varpi\right)\hbar}};$$

```
S𝔻$f = Simplify[e^(ℏ (t/2 - ϵ a)) (S𝔻$g /. {a → -a, t → -t})];
S𝔻$ϖ = γ CU[y, x] + ϵ CU[a, a] - (t - γ ϵ) CU[a] - t γ 1_CU / 2;
DeclareMorphism[S𝔻, QU → CU, {a → a_CU,
  x :→ 𝕊_CU[SS[S𝔻$f], a → a_CU, ϖ → S𝔻$ϖ] ** x_CU,
  y :→ 𝕊_CU[SS[S𝔻$g], a → a_CU, ϖ → S𝔻$ϖ] ** y_CU }]
```

```
ρ@y_CU = ρ@y_QU = (0 0
                   ϵ 0); ρ@a_CU = ρ@a_QU = (γ 0
                                            0 0);
ρ@x_CU = (0 γ
          0 0); ρ@x_QU = (0 (1 - e^(-γ ϵ ℏ)) / (ϵ ℏ)
                          0          0              );
ρ[e^ℰ_] := MatrixExp[ρ[ℰ]];
ρ[ℰ_] :=
  (ℰ /. TRule /. t → γ ϵ /.
    (U : CU | QU) [u___] ⧴ Fold[Dot, (1 0
                                       0 1), ρ /@ U /@ {u}])
𝔼_U_[s1__, Q1_, P1_] 𝔼_U_[s2__, Q2_, P2_] ^:=
  𝔼_U[s1, s2, Q1 + Q2, P1 P2];
CU@𝔼_CU[specs___, Q_, P_] := 𝕆_CU[specs, SS[e^Q P]];
QU@𝔼_QU[specs___, Q_, P_] := 𝕆_QU[specs, SS[e^Q P]];
c_Integer_k_Integer := c + O[ϵ]^(k+1);
Λ_U_,k_[{α_, β_}, {x_, x_}] := 𝔼_U[{x}, (α + β) x, 1_k];
Λ_U_,k_[{ξ_, α_}, {x, a}] := 𝔼_U[{a, x}, α a + e^(-γ α) ξ x, 1_k];
Λ_U_,k_[{α_, η_}, {a, y}] := 𝔼_U[{y, a}, α a + e^(-γ α) η y, 1_k];
```

Fear Not. If $G = e^{\xi x} y e^{-\xi x}$ then $F = e^{-\eta y} e^{\xi x} e^{\eta y} e^{-\xi x} = e^{-\eta y} e^{\eta G}$ satisfies $\partial_\eta F = -yF + FG$ and $F_{\eta=0} = 1$:

```
Λ_U_,kk_[{ξ1_, η1_}, {x, y}] :=
  Λ_U,kk[{ξ1, η1}, {x, y}] =
   Block[{$k = kk, $p = kk},
    Module[{ξ, η, G, F, fs, f, bs, e, b, es},
     G = Simp[Table[ξ^k / k!, {k, 0, $k + 1}].
         NestList[Simp[B[x_U, #]] &, y_U, $k + 1]];
     fs = Flatten@Table[f_1,i,j,k[η], {l, 0, $k}, {i, 0, l},
        {j, 0, l}, {k, 0, l}];
     F = fs.(bs = fs /. f_l_,i_,j_,k_[η] ⧴ ϵ^l U@{y^i, a^j, x^k});
     es = Flatten[Table[Coefficient[e, b] == 0,
        {e, {F - 1_U /. η → 0, F ** G - y_U ** F - ∂_η F}},
        {b, bs}]];
     F = F /. DSolve[es, fs, η][[1]];
     𝔼_U[{y, a, x},
       ξ x + η y + (U /. {CU → -t η ξ, QU → η ξ (1 - T) / ℏ}),
       F + 0_$k /. {e- → 1, U → Times}
      ] /. {ξ → ξ1, η → η1}]];
Simp[𝔼_U_[specs___, Q_, P_]] := 𝔼_U[specs, CF[Q], CF[P]];
Λ_U_,k_[{υ1_, ω1_, δ_}, {u_, w_}] :=
  Simp@Module[{υ, ω, yax, q, p, Q, d},
    {yax, q, p} = List @@ Λ_U,k[{υ, ω}, {u, w}];
    𝔼_U[yax, Q = (υ u + ω w + δ u w + d υ ω) / (1 - d δ),
      Expand[(1 - d δ)^-1 e^-Q DP_υ→D_u,ω→D_w[p][e^q]] + 0_k] /.
     {d → ∂_υ,ω q} /. {υ → υ1, ω → ω1}];
Rord_u_i_,w_j_→k_[𝔼_U_[L___, {L___, u_i_, w_j_, r___}_s_,
    R___, Q_, P_]] :=
  Simp@Module[{υ, ω, δ, Λ1, yax, q, p, kk = P[[5]],
     δ1 = ∂_u_i,w_j Q},
    {yax, q, p} =
     Echo[
      List @@ If[δ1 === 0, Λ_U,kk[{υ, ω}, {u, w}],
         Λ_U,kk[{υ, ω, δ}, {u, w}]] /.
       {y → y_k, a → a_k, x → x_k, t → t_s, T → T_s}];
    𝔼_U[L, {L, Sequence @@ yax, r}_s, R, q + (Q /. u_i | w_j → 0),
      e^-q DP_u_i→D_υ,w_j→D_ω[P][p e^q]] /.
     {υ → ∂_u_i Q /. w_j → 0, ω → ∂_w_j Q /. u_i → 0, δ → δ1}];
Rord_u_i_,w_j_→k_[𝔼_U_[L___, {L___, u_i_, w_j_, r___}_s_,
    R___, Q_, P_]] :=
  Simp@Module[{υ, ω, δ, Λ1, yax, q, p, n, kk = P[[5]],
     δ1 = ∂_u_i,w_j Q},
    {yax, q, p} =
     List @@ If[δ1 === 0, Λ_U,kk[{υ, ω}, {u, w}],
        Λ_U,kk[{υ, ω, δ}, {u, w}]] /.
      {y → y_n, a → a_n, x → x_n, t → t_s, T → T_s};
    (*Echo@{{{u_i,υ},{w_j,ω}},P,p e^q};*)
    𝔼_U[L, {L, Sequence @@ yax, r}_s, R, q + (Q /. u_i | w_j → 0),
      e^-q SP_{u_i→υ,w_j→ω}[P p e^q]] /.
     {n → k, υ → ∂_u_i Q /. w_j → 0, ω → ∂_w_j Q /. u_i → 0, δ → δ1}];
Cord[𝔼_U_[L___, {L___, u_i_, w_j_, r___}_s_, R___, Q_, P_]] /;
   OrderedQ[{w, u} /. {y → 1, a → 2, x → 3}] :=
  ((*Echo@{u_i,w_j};*)
   Cord[Rord_u_i,w_j→Unique[][𝔼_U[L, {L, u_i, w_j, r}_s, R, Q, P]]]);
Cord[𝔼_U_[specs__, Q_, P_]] :=
  𝔼_U[Sequence @@ Sort@{specs}, Q, P] /.
   Flatten[{specs} /. {yax__}_s_ ⧴ ({yax} /. u_i_ ⧴ (u_i → u_s))]
m_j_→k_[𝔼_U_[specs__, Q_, P_]] :=
  Cord[
   𝔼_U[Sequence @@ Append[DeleteCases[{specs}, {__}_j|k],
      Flatten[{Cases[{specs}, {us__}_j ⧴ {us}],
        Cases[{specs}, {us__}_k ⧴ {us}]}]_k], Q, P] /.
    {t_j → t_k, T_j → T_k}]
e_q_,k_[x_] := e^(Sum[(1 - q)^j x^j / (j (1 - q^j)), {j, 1, k+1}]); e_q_[x_] := e_q,$k[x]
QU[R_i_,j_] := 𝕆_QU[{y_1, a_1}_i, {a_2, x_2}_j,
   SS[e^(ℏ b_1 a_2) e_q_ℏ[ℏ y_1 x_2] /. b_1 → γ^-1 (ϵ a_1 - t_i)]];
QU[R^-1_i_,j_] := S_j@QU[R_i,j];
𝔼_QU,k_[R_i_,j_] := 𝔼_QU[{y_i, a_i, x_i}_i, {y_j, a_j, x_j}_j,
   -ℏ γ^-1 t_i a_j + ℏ y_i x_j,
   Series[e^(ℏ γ^-1 t_i a_j - ℏ y_i x_j)
     (e^(ℏ b_i a_j) e_q_ℏ,k[ℏ y_i x_j] /. b_i → γ^-1 (ϵ a_i - t_i)), {ϵ, 0, k}]]
𝔼_U_,k_[a_ * b_] := 𝔼_U,k[a] 𝔼_U,k[b];
𝔼_U_,k_[m_is__[a_]] := m_is[𝔼_U,k[a]];
```

Task. Define $\mathrm{Exp}_{U_i,k}[\xi, P]$ which computes $e^{\xi\,\mathbb{O}(P)}$ to ϵ^k in the algebra U_i, where ξ is a scalar, X is x_i or y_i, and P is an ϵ-dependent near-docile element, giving the answer in $\mathbb{E}$-form. Should satisfy $U@\mathrm{Exp}_{U_i,k}[\xi, P] == \mathbb{S}_U[e^{\xi x}, x \to \mathbb{O}(P)]$.

Methodology. If $P_0 := P_{\epsilon=0}$ and $e^{\xi\,\mathbb{O}(P)} = \mathbb{O}(e^{\xi P_0} F(\xi))$, then $F(\xi = 0) = 1$ and we have:

$$\mathbb{O}\left(e^{\xi P_0}(P_0 F(\xi) + \partial_\xi F\right) = \mathbb{O}\left(\partial_\xi e^{\xi P_0} F(\xi)\right) =$$

$$\partial_\xi \mathbb{O}\left(e^{\xi P_0} F(\xi)\right) = \partial_\xi e^{\xi\,\mathbb{O}(P)} = e^{\xi\,\mathbb{O}(P)}\,\mathbb{O}(P) = \mathbb{O}\left(e^{\xi P_0} F(\xi)\right)\mathbb{O}(P)$$

This is an 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^P0) == e^O(P0). *)
(* Bug: ξ must be a symbol. *)
Exp_{U_i_,0}[ξ_, P_] := 𝔼_U[{y_i, a_i, x_i}_i, Normal@P /. ϵ → 0,
   1 + 0_0];
Exp_{U_i_,k_}[ξ_, P_] :=
 Module[{yax = {y_i, a_i, x_i}, P0, φ, φs, F, j, rhs, at0, atξ},
  P0 = Normal@P /. ϵ → 0;
  φs = Flatten@Table[φ_{j1,j2,j3}[ξ], {j2, 0, k},
     {j1, 0, 2 k + 1 - j2}, {j3, 0, 2 k + 1 - j2 - j1}];
  F = Normal@Last@Exp_{U_i,k-1}[ξ, P] +
    ϵ^k φs.(φs /. φ_{js__}[ξ] :> Times @@ yax^{js});
  rhs =
   Normal@
    Last@m_{i,j→i}[𝔼_U[yax_i, ξ P0, F + 0_k]
      m_{i→j}@𝔼_U[{y_i, a_i, x_i}_i, 0, P + 0_k]];
  at0 = (# == 0) & /@
    Flatten@CoefficientList[F - 1 /. ξ → 0, yax];
  atξ = (# == 0) & /@
    Flatten@CoefficientList[(∂_ξ F) + P0 F - rhs, yax];
  𝔼_U[yax_i, ξ P0, F + 0_k] /.
   DSolve[And @@ (at0 ⋃ atξ), φs, ξ][[1]] ]
```

To do. • Consider renormalizing x and y. • Can everything be done at $\hbar = 1$ defining a filtration by other means? That ought to be possible as the end results depend on t/T and not on $\hbar$. • Bound the degrees of the logoi! • $r = \theta r$?

Alternative Algorithms.

```
λ_{alt,k_}[CU] := If[k == 0, 1, Module[{eq, d, b, c, so},
    eq = ρ@e^{ξ x_CU}.ρ@e^{η y_CU} == ρ@e^{d y_CU}.ρ@e^{c (t 1_CU - 2 ϵ a_CU)}.ρ@e^{b x_CU};
    {so} = Solve[Thread[Flatten /@ eq], {d, b, c}] /.
      C@1 → 0;
    Series[e^{-η y - ξ x + η ξ t + c t + d y - 2 ϵ c a + b x} /. so, {ϵ, 0, k}]]];
```

Program (as in Projects/PPSA/Verification.nb).

```
Unprotect[NonCommutativeMultiply];
Attributes[NonCommutativeMultiply] = {};
(NCM = NonCommutativeMultiply)[x_] := x;
NCM[x_, y_, z__] := (x ** y) ** z;
0 ** _ = _ ** 0 = 0;
(x_Plus) ** y_ := (# ** y) & /@ x;
x_ ** (y_Plus) := (x ** #) & /@ y;
B[x_, x_] = 0; B[x_, y_] := x ** y - y ** x;
B[x_, y_, e_] := B[x, y, e] = B[x, y];
DeclareMorphism[m_, U_ → V_, ongs_List, oncs_List: {}] := (
  Replace[ongs, {(g_ → img_) :> (m[U[g]] = img),
     (g_ :> img_) :> (m[U[g]] := img /. $trim)}, {1}];
  m[1_U] = 1_V;
  m[U[g_i_]] := V_i[m[U@g]];
  m[U[vs__]] := NCM @@ (m /@ U /@ {vs});
  m[ℰ_] := Simp[ℰ /. oncs /. u_U :> m[u]] /. $trim; )
σ_{rs___}[ℰ_Plus] := σ_rs /@ ℰ;
m_{j_→j_} = Identity; m_{j_→k_}[0] = 0;
m_{j_→k_}[ℰ_Plus] := Simp[m_{j→k} /@ ℰ];
m_{is___,i_,j_→k_}[ℰ_] := m_{j→k}@m_{is,i→j}@ℰ;
S_{i_}[ℰ_Plus] := Simp[S_i /@ ℰ];
Δ_{is___}[ℰ_Plus] := Simp[Δ_is /@ ℰ];
```

```
DeclareAlgebra[U_Symbol, opts__Rule] :=
 Module[{gp, sr, g, cp, M, CE, pow, k = 0,
   gs = Generators /. {opts},
   cs = Centrals /. {opts} /. Centrals → {} },
  (#_U = U@#) & /@ gs;
  gp = Alternatives @@ gs; gp = gp | gp_; (* gens *)
  sr = Flatten@Table[{g → ++k, g_i_ → {i, k}}, {g, gs}];
  (* sorting → *)
  cp = Alternatives @@ cs; (* cents *)
  SetAttributes[M, HoldRest]; M[0, _] = 0;
  M[a_, x_] := a x;
  CE[ℰ_] := Collect[ℰ, _U, Expand] /. $trim;
  U_i_[ℰ_] := ℰ /. {t : cp :> t_i, u_U :> (#_i &) /@ u};
  U_i_[NCM[]] = pow[ℰ_, 0] = U@{} = 1_U = U[];
  B[U@(x_)_i_, U@(y_)_i_] := U_i@B[U@x, U@y];
  B[U@(x_)_i_, U@(y_)_j_] /; i =!= j := 0;
  B[U@y_, U@x_] := CE[-B[U@x, U@y]];
  x_ ** (c_. 1_U) := CE[c x]; (c_. 1_U) ** x_ := CE[c x];
  (a_. U[xx___, x_]) ** (b_. U[y_, yy___]) :=
   If[OrderedQ[{x, y} /. sr],
    CE@M[a b /. $trim, U[xx, x, y, yy]],
    U@xx **
     CE@M[a b /. $trim, U@y ** U@x + B[U@x, U@y, $E]] **
     U@yy ];
  U@{c_. * (l : gp)^n_, r___} /; FreeQ[c, gp] :=
   CE[c U@Table[l, {n}] ** U@{r}];
  U@{c_. * l : gp, r___} := CE[c U[l] ** U@{r}];
  U@{c_, r___} /; FreeQ[c, gp] := CE[c U@{r}];
  U@{l_Plus, r___} := CE[U@{#, r} & /@ l];
  U@{l_, r___} := U@{Expand[l], r};
  U[ℰ_NonCommutativeMultiply] := U /@ ℰ;
  O_U[specs___, poly_] := Module[{sp, null, vs, us},
    sp = Replace[{specs}, l_List :> l_null, {1}];
    vs = Join @@ (First /@ sp);
    us = Join @@ (sp /. l_s_ :> (l /. x_i_ :> x_s));
    CE[Total[
       CoefficientRules[poly, vs] /. (p_ → c_) :> c U@(us^p)
      ]] /. x_null :> x];
  pow[ℰ_, n_] := pow[ℰ, n - 1] ** ℰ;
  S_U[ℰ_, ss___Rule] := CE@Total[
     CoefficientRules[ℰ, First /@ {ss}] /.
      (p_ → c_) :>
       c NCM @@ MapThread[pow, {Last /@ {ss}, p}]];
  σ_{rs___}[c_. * u_U] :=
   (c /. (t : cp)_j_ :> t_{j /. {rs}}) U[List @@ (u /. v_j_ :> v_{j /. {rs}})];
  m_{j_→k_}[c_. * u_U] :=
   CE[((c /. (t : cp)_j → t_k) DeleteCases[u, _j|k]) **
     U @@ Cases[u, w_j :> w_k] ** U @@ Cases[u, _k]];
  U /: c_. * u_U * v_U := CE[c u ** v];
  S_i_[c_. * u_U] :=
   CE[((c /. S_i[U, Centrals]) DeleteCases[u, _i]) **
     U_i[NCM @@ Reverse@Cases[u, x_i :> S@U@x]]];
  Δ_{i_→j_,k_}[c_. * u_U] :=
   CE[((c /. Δ_{i→j,k}[U, Centrals]) DeleteCases[u, _i]) **
     (NCM @@ Cases[u, x_i :> σ_{1→j,2→k}@Δ@U@x] /.
      NCM[] → U[])]; ]
```

emancipate?

Asides. `Series[(1 - T e^{-2 ϵ a ħ}) / ħ, {a, 0, 3}]`

$$\frac{1-T}{\hbar} + 2 T \epsilon a - 2 \left(T \epsilon^2 \hbar\right) a^2 + \frac{4}{3} T \epsilon^3 \hbar^2 a^3 + O[a]^4$$

```
c𝕆 = 𝔼CU[{y1, a1, x1}1, {y2, a2, x2}2,
   ℏ Sum[l10 i+j ti aj + γ10 i+j yi xj, {i, 2}, {j, 2}], 11];
Short[Simplify /@ (cexample = c𝕆 // m1→2), 12]
Short[Simplify /@ (qexample = (q𝕆 = c𝕆 /. CU → QU) // m1→2),
 12]
```

```
𝔼CU[{y2, a2, x2}2,
 ℏ a2 (l11 + l12 + l21 + l22) t2 + 1/(1 + ℏ t2 γ21) e^(-γ ℏ (l11+l12+l21+l22) t2) ℏ x2 y2
   (γ21 + e^(γ ℏ (l11+l12+l21+l22) t2) γ12 (1 + ℏ t2 γ21) + e^(γ ℏ (l12+l22) t2) γ22 +
    γ11 (e^(γ ℏ (l11+l21) t2) - e^(γ ℏ (l11+l12+l21+l22) t2) ℏ t2 γ22)),
 1/(1 + ℏ t2 γ21) + 1/(2 (1 + ℏ t2 γ21)^5) e^(-2 γ ℏ (l11+l12+l21+l22) t2)
   ℏ (4 a2 (1 + ℏ t2 γ21)^2
      (e^(γ ℏ (l11+l12+l21+l22) t2) ℏ (e^(γ ℏ (l11+l12+l21+l22) t2) t2 + x2 y2) γ21^2 +
       e^(2 γ ℏ (l11+l12+l21+l22) t2) ℏ x2 y2 γ11 γ22 + γ21
        (e^(2 γ ℏ (l11+l12+l21+l22) t2) + ℏ x2 y2 (e^(γ ℏ (2 l11+l12+2 l21+l22) t2)
             γ11 + e^(γ ℏ (l11+2 l12+l21+2 l22) t2) γ22))) -
     γ ℏ (-2 e^(2 γ ℏ (l11+l12+l21+l22) t2) t2 γ21^2 (1 + ℏ t2 γ21)^2 +
        4 <<5>> (<<1>>) + ℏ <<4>> (3 ℏ t2 γ21^2 + 2 e^(γ ℏ (l12+l<<2>>) t2)
             γ22 + γ21 (4 + e^(γ <<3>>) ℏ t2 γ22) + e^(γ ℏ (l11+l21) t2) γ11
             (2 + ℏ t2 (γ21 - e^(γ ℏ (l12+l22) t2) γ22))))) ∈ + O[∈]^2]
𝔼QU[{y2, a2, x2}2,
 ℏ a2 (l11 + l12 + l21 + l22) t2 + 1/(1 + (-1 + T2) γ21) e^(-γ ℏ (l11+l12+l21+l22) t2)
   ℏ x2 y2 (γ21 + e^(γ ℏ (l11+l12+l21+l22) t2) γ12 (1 + (-1 + T2) γ21) +
     e^(γ ℏ (l12+l22) t2) γ22 +
     γ11 (e^(γ ℏ (l11+l21) t2) - e^(γ ℏ (l11+l12+l21+l22) t2) (-1 + T2) γ22)),
 1/(1 + (-1 + T2) γ21) + 1/(4 (1 + (-1 + T2) γ21)^5) e^(-2 γ ℏ (l11+l12+l21+l22) t2)
   ℏ (8 a2 T2 (1 + (-1 + T2) γ21)^2 (e^(γ ℏ (l11+l12+l21+l22) t2)
          (-e^(γ ℏ (l11+l12+l21+l22) t2) + e^(γ ℏ (l11+l12+l21+l22) t2) T2 + ℏ x2 y2)
          γ21^2 + e^(2 γ ℏ (l11+l12+l21+l22) t2) ℏ x2 y2 γ11 γ22 +
         γ21 (e^(2 γ ℏ (l11+l12+l21+l22) t2) + ℏ x2 y2 (e^(γ ℏ (2 l11+l12+2 l21+l22) t2)
               γ11 + e^(γ ℏ (l11+2 l12+l21+2 l22) t2) γ22))) +
     γ (2 e^(2 γ ℏ (l11+l12+l21+l22) t2) (1 - 4 T2 + 3 T2^2) γ21^2
          (1 + (-1 + T2) γ21)^2 + 4 e^(γ ℏ (l11+l12+l21+l22) t2) ℏ x2 y2
          γ21 (1 + (-1 + T2) γ21) (<<1>>) - <<1>>)) ∈ + O[∈]^2]
```

```
Timing@Block[{$p = 3, $k = 1}, {
   (sexp = m3,2,1→1[ExpQU1,$k[η, S1[QU[y1]]] /. QU → Times]
        ExpQU2,$k[α, S2[QU[a2]]] /. QU → Times]
        ExpQU3,$k[ξ, S3[QU[x3]]] /. QU → Times]]) /. u_1 :→ u,
   HL@SimpT[QU@(sexp /. {η → ℏ η, α → ℏ α, ξ → ℏ ξ}) -
      S1@𝕆QU[{y1, a1, x1}1, SS[e^(ℏ (η y1+α a1+ξ x1))]]]
  }]
```

```
{15.2969, {𝔼QU[{y1, a1, x1},
   (e^(α γ) η ξ - e^(α γ) T η ξ - a T α ℏ - e^(α γ) y η ℏ - e^(α γ) T x ξ ℏ)/(T ℏ),
   1 + 1/(4 T^2 ℏ) (-3 e^(2 α γ) γ η^2 ξ^2 + 4 e^(2 α γ) T γ η^2 ξ^2 -
       e^(2 α γ) T^2 γ η^2 ξ^2 + 8 a e^(α γ) T η ξ ℏ - 4 e^(α γ) T γ η ξ ℏ +
       4 e^(α γ) T^2 γ η ξ ℏ + 6 e^(2 α γ) y γ η^2 ξ ℏ - 2 e^(2 α γ) T y γ η^2 ξ ℏ +
       6 e^(2 α γ) T x γ η ξ^2 ℏ - 2 e^(2 α γ) T^2 x γ η ξ^2 ℏ - 4 a e^(α γ) T y η ℏ^2 +
       4 e^(α γ) T y γ η ℏ^2 - 2 e^(2 α γ) y^2 γ η^2 ℏ^2 - 4 a e^(α γ) T^2 x ξ ℏ^2 -
       4 e^(2 α γ) T x y γ η ξ ℏ^2 - 2 e^(2 α γ) T^2 x^2 γ ξ^2 ℏ^2) ∈ + O[∈]^2], 0}}
```

```
Timing@Block[{$p = 4, $k = 2}, {
   sexp = m1,3,5→1@m2,4,6→2@Times[
       Prepend[{y2}2]@ExpQU1,$k[η, Δ1→1,2[QU[y1]] /. QU → Times],
       Prepend[{a4}4]@
        ExpQU3,$k[α, Δ3→3,4[QU[a3]] /. QU → Times],
       Prepend[{x6}6]@
        ExpQU5,$k[ξ, Δ5→5,6[QU[x5]] /. QU → Times]
      ] /. {η → ℏ η, α → ℏ α, ξ → ℏ ξ},
   HL@
    SimpT[
     QU@sexp - Δ1→1,2@𝕆QU[{y1, a1, x1}1, SS[e^(ℏ (η y1+α a1+ξ x1))]]]
  }]
```

```
{162., {𝔼QU[{y2, a2, x2}2, {y1, a1, x1}1,
   α ℏ a1 + α ℏ a2 + ξ ℏ x1 + ξ ℏ x2 + η ℏ y1 + η ℏ T1 y2, 1 +
    1/2 (-2 ξ ℏ^2 a1 x2 + γ ξ^2 ℏ^3 x1 x2 - 2 η ℏ^2 a1 T1 y2 + γ η^2 ℏ^3 T1 y1 y2) ∈ +
    1/24 (12 ξ ℏ^3 a1^2 x2 + 6 γ^2 ξ^2 ℏ^4 x1 x2 - 12 γ ξ^2 ℏ^4 a1 x1 x2 +
       4 γ^2 ξ^3 ℏ^5 x1^2 x2 + 12 ξ^2 ℏ^4 a1^2 x2^2 + 4 γ^2 ξ^3 ℏ^5 x1 x2^2 -
       12 γ ξ^3 ℏ^5 a1 x1 x2^2 + 3 γ^2 ξ^4 ℏ^6 x1^2 x2^2 + 12 η ℏ^3 a1^2 T1 y2 +
       24 η ξ ℏ^4 a1^2 T1 x2 y2 - 12 γ η ξ^2 ℏ^5 a1 T1 x1 x2 y2 +
       6 γ^2 η^2 ℏ^4 T1 y1 y2 - 12 γ η^2 ℏ^4 a1 T1 y1 y2 -
       12 γ η^2 ξ ℏ^5 a1 T1 x2 y1 y2 + 6 γ^2 η^2 ξ^2 ℏ^6 T1 x1 x2 y1 y2 +
       4 γ^2 η^3 ℏ^5 T1 y1^2 y2 + 12 η^2 ℏ^4 a1^2 T1^2 y2^2 + 4 γ^2 η^3 ℏ^5 T1^2 y1 y2^2 -
       12 γ η^3 ℏ^5 a1 T1^2 y1 y2^2 + 3 γ^2 η^4 ℏ^6 T1^2 y1^2 y2^2) ∈^2 + O[∈]^3], 0}}
```

(Proposed) Agenda. Using Århus-like techniques, construct a map $Z\colon \mathcal{T}_{vous} \to \mathcal{A}_{vous}$, where $\mathcal{T}_{vous}$ is the space of VOUS-tangles: Virtual tangles with only Over or Under strands, some labeled as Surgery strands, with a non-singular linking matrix between the surgery strands, modulo acyclic Reidemeister 2 moves and Kirby slide relations, and where $\mathcal{A}_{vous}$ is some space of arrow diagrams modulo appropriate relations. The construction will either fix the definitions of $\mathcal{T}_{vous}$ and $\mathcal{A}_{vous}$ or will allow some flexibility that will be fixed so that the following will hold true:

1. $\mathcal{T}_{vous}$ should have a clearer topological interpretation, perhaps in terms of Heegaard diagrams.
2. $\mathcal{A}_{vous}$ should pair with some kind of Lie bialgebras.
3. $\mathcal{A}_{vous}$ should be the associated graded of $\mathcal{T}_{vous}$ and Z should be an expansion.
4. Ordinary tangles $\mathcal{T}_{ord}$ and ordinary virtual tangles $\mathcal{T}_{v\text{-}ord}$ should map into $\mathcal{T}_{vous}$, and when viewed on $\mathcal{T}_{(v\text{-})ord}$, the invariant Z should explain the Drinfel'd double construction.

It may be better to first construct a Z and only later worry about the numbered properties. Yet property 4 has stand-alone topological content which may be very interesting: $\mathcal{T}_{vous}$ is a space with an $R3$-free presentation and which contains $\mathcal{T}_{(v\text{-})ord}$, at least nearly faithfully. What does it mean? To what extent does it make $R3$ superfluous in knot theory?
As for constructing Z, the first step should be a $Z\colon \mathcal{T}_{vou} \to \mathcal{A}_{vou}$ (no surgery), which would have a prescribed behaviour on strand-doubling.