Cheat Sheet SL2Portfolio on 180318

March 18, 2018 5:48 PM

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

http://drorbn.net/AcademicPensieve/Projects/SL2Portfolio/
modified March 18, 2018, 12:34

$\mathcal{U}_{\gamma\epsilon;\hbar}$ 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 $\langle y^lb^i,a^jx^k\rangle=\delta_{ij}\delta_{kl}\hbar^{-(j+k)}j![k]_q!$ so $R=\sum\frac{\hbar^{j+k}y^kb^j\otimes a^jx^k}{j![k]_q!}$. Then $\mathcal{U}=H^{*cop}\otimes H$ with $(\phi f)(\psi g)=\langle\psi_1S^{-1}f_3\rangle\langle\psi_3,f_1\rangle(\phi\psi_2)(f_2g)$ 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 $t:=\epsilon a-\gamma b$, $T:=e^{\hbar t}=A^{-1/2}B^{1/2}$ 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 $\epsilon=0$, $\mathcal{U}_{\hbar;\gamma0}=\langle t,y,a,x\rangle/([t,\cdot]=0,[a,x]=\gamma x,[a,y]=-\gamma y,[x,y]=(1-T)/\hbar)$ 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: $\{\gamma,\epsilon,a,b,x,y\}\to1$, $\{\hbar\}\to-2$, $\{t\}\to2$, $\{\omega\}\to3$.

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

```
$p = 2; $k = 1; $E := {$k, $p};
$trim := {ℏ^p_. /; p > $p → 0, ϵ^k_. /; k > $k → 0};
SetAttributes[{SS, SST}, HoldAll];
TRule = {T_i_ → e^(ℏ t_i), T → e^(ℏ t)}; qℏ = e^(γ ϵ ℏ);
SS[ℰ_, op_] := Collect[
   Normal@Series[If[$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@CU@y = -y_CU; S@a_CU = -a_CU; S@x_CU = -x_CU;)
S_i_[CU, Centrals] = {t_i → -t_i};
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[qℏ - 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};
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}]
AD$f = γ (Cosh[ℏ (a ϵ + γϵ/2 - t/2)] - Cosh[ℏ Sqrt[((t-γϵ)/2)^2 + ϵ ω]]) /
   (ℏ e^(ℏ ((a+γ) ϵ - t/2)) Sinh[γϵℏ/2] (a^2 ϵ + a γ ϵ - a t - ω));
AD$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}]
SD$g = Sqrt[(2 γ (Cosh[ℏ/2 Sqrt[t^2 + γ^2 ϵ^2 + 4 ϵ ϖ]] - Cosh[(t-ϵγ-2ϵa)/(2/ℏ)])) /
   (Sinh[γϵℏ/2] (t (2 a + γ) - 2 a (a + γ) ϵ + 2 ϖ) ℏ)];
SD$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[{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_(l,i,j,k)[η], {l, 0, $k}, {i, 0, 1},
         {j, 0, 1}, {k, 0, 1}];
     F = fs.(bs = fs /. f_(l_,i_,j_,k_)[η] :> ϵ^l U@{y^i, a^j, x^k});
     es = Flatten[Table[Coefficient[e, b] == 0,
         {e, {F - 1_U /. η → 0, F ** G - y_U ** F - ∂_η F}},
         {b, bs}]];
     F = F /. DSolve[es, fs, η][[1]];
     𝔼_U[{y, a, x},
      ξ x + η y + (U /. {CU → -t η ξ, QU → η ξ (1 - T) / ℏ}),
      F + 0_$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@
    Echo[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]];
SxF[0] = 1;
SxF[k_] := SxF[k] = Module[{fs, bs, F, rhs, at0, atξ},
    fs = Flatten@Table[f_(i,j)[ξ], {i, 0, 2 k}, {j, 0, 2 k - i}];
    F = SxF[k - 1] + ϵ^k fs.(bs = fs /. f_(i_,j_)[ξ] :> a^i x^j);
    rhs =
     Normal@
      Last@
       Cord[𝔼_QU[{a_1, x_1, a_2, x_2}_1, -ξ x_1,
           (F /. {a → a_1, x → x_1})
            Series[-x_2 e^(ℏ ϵ a_2), {ϵ, 0, k}]] /. ξ → ℏ ξ] /.
     {ξ → ℏ^-1 ξ, a_1 → a, x_1 → x};
    at0 = (# == 0) & /@
      Flatten@CoefficientList[F - 1 /. ξ → 0, {a, x}];
    atξ = (# == 0) & /@
      Flatten@CoefficientList[(∂_ξ F) - x F - rhs, {a, x}];
    F /. DSolve[And @@ (at0 ⋃ atξ), fs, ξ][[1]]
   ];
```

```
SyF[0] = 1;
SyF[k_] := SyF[k] = Module[{fs, bs, F, rhs, at0, atη},
   fs = Flatten@Table[f_{i,j}[η], {i, 0, 2 k}, {j, 0, 2 k - i}];
   F = SyF[k - 1] + ϵ^k fs.(bs = fs /. f_{i_,j_}[η] :→ y^i a^j);
   rhs =
    Normal@
     Last@
      Cord[Œ_QU[{y1, a1, a2, y2}1, -T^-1 η y1,
          (F /. {a → a1, y → y1})
           Series[-y2 T^-1 e^(ℏ ϵ a2), {ϵ, 0, k}]] /. η → ℏ η] /.
     {η → ℏ^-1 η, a1 → a, y1 → y};
   at0 = (# == 0) & /@
     Flatten@CoefficientList[F - 1 /. η → 0, {a, y}];
   atη = (# == 0) & /@
     Flatten@CoefficientList[(∂_η F) - T^-1 y F - rhs, {a, y}];
   F /. DSolve[And @@ (at0 ⋃ atη), fs, η][[1]]
  ];
```

Next task: $\mathrm{Exp}_U : U \to Œ$ ...

Next next task: Define $S\Lambda_{U,k}[\eta, \alpha, \xi, \delta]$, whose value is an $Œ_U[\{y_1, a_1, x_1\}_1,\ Q,\ P+0_k]$ such that

```
U@SΛ_{U,k}[η, α, ξ, δ] ==
 S1@U@Œ_U[{y1, a1, x1}1, η y1 + α a1 + ξ x1 + δ x1 y1, 0_k]
```

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!

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; )
m_{j_→j_} = Identity;
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];
```

```
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}]];
  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]]] ]
```

emancipati?

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

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

```
cѲ = Œ_CU[{y1, a1, x1}1, {y2, a2, x2}2,
   ℏ Sum[l_{10 i+j} t_i a_j + γ_{10 i+j} y_i x_j, {i, 2}, {j, 2}], 1_1];
Short[Simplify /@ (cexample = cѲ // m_{1→2}), 12]
Short[Simplify /@ (qexample = (qѲ = cѲ /. CU → QU) // m_{1→2}),
 12]
```

$$\mathbb{E}_{CU}\Big[\{y_2, a_2, x_2\}_2,$$
$$\hbar\, a_2\, (l_{11}+l_{12}+l_{21}+l_{22})\, t_2 + \frac{1}{1+\hbar\, t_2\, \gamma_{21}} e^{-\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, x_2\, y_2$$
$$\Big(\gamma_{21} + e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \gamma_{12}\, (1+\hbar\, t_2\, \gamma_{21}) + e^{\gamma\hbar\,(l_{12}+l_{22})\, t_2}\, \gamma_{22} +$$
$$\gamma_{11}\, \Big(e^{\gamma\hbar\,(l_{11}+l_{21})\, t_2} - e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, t_2\, \gamma_{22}\Big)\Big),$$
$$\frac{1}{1+\hbar\, t_2\, \gamma_{21}} + \frac{1}{2\,(1+\hbar\, t_2\, \gamma_{21})^5} e^{-2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}$$
$$\hbar\, \Big(4\, a_2\, (1+\hbar\, t_2\, \gamma_{21})^2$$
$$\Big(e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, \Big(e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, t_2 + x_2\, y_2\Big)\, \gamma_{21}^2 +$$
$$e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, x_2\, y_2\, \gamma_{11}\, \gamma_{22} + \gamma_{21}$$
$$\Big(e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2} + \hbar\, x_2\, y_2\, \Big(e^{\gamma\hbar\,(2 l_{11}+l_{12}+2 l_{21}+l_{22})\, t_2}$$
$$\gamma_{11} + e^{\gamma\hbar\,(l_{11}+2 l_{12}+l_{21}+2 l_{22})\, t_2}\, \gamma_{22}\Big)\Big)\Big) -$$
$$\gamma\hbar\, \Big(-2\, e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, t_2\, \gamma_{21}^2\, (1+\hbar\, t_2\, \gamma_{21})^2 +$$
$$4 \ll 5 \gg (\ll 1 \gg) + \hbar \ll 4 \gg \Big(3\, \hbar\, t_2\, \gamma_{21}^2 + 2\, e^{\gamma\hbar\,(l_{12}+l_{\ll 2 \gg})\, t_2}$$
$$\gamma_{22} + \gamma_{21}\, \Big(4 + e^{\gamma \ll 3 \gg}\, \hbar\, t_2\, \gamma_{22}\Big) + e^{\gamma\hbar\,(l_{11}+l_{21})\, t_2}\, \gamma_{11}$$
$$\Big(2 + \hbar\, t_2\, \Big(\gamma_{21} - e^{\gamma\hbar\,(l_{12}+l_{22})\, t_2}\, \gamma_{22}\Big)\Big)\Big)\Big)\Big)\, \epsilon + O[\epsilon]^2\Big]$$

$$\mathbb{E}_{QU}\Big[\{y_2, a_2, x_2\}_2,$$
$$\hbar\, a_2\, (l_{11}+l_{12}+l_{21}+l_{22})\, t_2 + \frac{1}{1+(-1+T_2)\, \gamma_{21}} e^{-\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}$$
$$\hbar\, x_2\, y_2\, \Big(\gamma_{21} + e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \gamma_{12}\, (1+(-1+T_2)\, \gamma_{21}) +$$
$$e^{\gamma\hbar\,(l_{12}+l_{22})\, t_2}\, \gamma_{22} +$$
$$\gamma_{11}\, \Big(e^{\gamma\hbar\,(l_{11}+l_{21})\, t_2} - e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, (-1+T_2)\, \gamma_{22}\Big)\Big),$$
$$\frac{1}{1+(-1+T_2)\, \gamma_{21}} + \frac{1}{4\,(1+(-1+T_2)\, \gamma_{21})^5} e^{-2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}$$
$$\hbar\, \Big(8\, a_2\, T_2\, (1+(-1+T_2)\, \gamma_{21})^2\, \Big(e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}$$
$$\Big(-e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2} + e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, T_2 + \hbar\, x_2\, y_2\Big)$$
$$\gamma_{21}^2 + e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, x_2\, y_2\, \gamma_{11}\, \gamma_{22} +$$
$$\gamma_{21}\, \Big(e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2} + \hbar\, x_2\, y_2\, \Big(e^{\gamma\hbar\,(2 l_{11}+l_{12}+2 l_{21}+l_{22})\, t_2}$$
$$\gamma_{11} + e^{\gamma\hbar\,(l_{11}+2 l_{12}+l_{21}+2 l_{22})\, t_2}\, \gamma_{22}\Big)\Big)\Big) +$$
$$\gamma\, \Big(2\, e^{2\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \Big(1 - 4\, T_2 + 3\, T_2^2\Big)\, \gamma_{21}^2$$
$$(1+(-1+T_2)\, \gamma_{21})^2 + 4\, e^{\gamma\hbar\,(l_{11}+l_{12}+l_{21}+l_{22})\, t_2}\, \hbar\, x_2\, y_2$$
$$\gamma_{21}\, (1+(-1+T_2)\, \gamma_{21})\, (\ll 1 \gg) - \ll 1 \gg\Big)\Big)\, \epsilon + O[\epsilon]^2\Big]$$

(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.

$$S(x) = e^{\epsilon a_1} x = \left(1 + \epsilon a + \frac{\epsilon^2}{2} a^2\right) x$$