Pensieve header: A unified verification program for the sl_2-portfolio project, Uxi version. Continues pensieve://Projects/SL2Portfolio/nb/Verification.pdf.

Also continues pensieve://Projects/PPSA/nb/Verification.pdf and pensieve://2017-06/ and pensieve://2017-08/.

DocileQ

DocileQ

In[•]:=
```
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);
```

Initialization / Utilities

It is verification-risky to work with low $E!

TD
```
$p = 2; $k = 1; $U = QU; $E := {$k, $p};
$trim := {ħ^p_. /; p > $p → 0, ϵ^k_. /; k > $k → 0};
SetAttributes[{SS, SST}, HoldAll];
q_ħ = e^(γ ϵ ħ);
(* Upper to lower and lower to Upper: *)
U2l = {B_i_^p_. → e^(-p ħ γ b_i), B^p_. → e^(-p ħ γ b), T_i_^p_. → e^(p ħ t_i), T^p_. → e^(p ħ t),
   𝒜_i_^p_. → e^(p γ α_i), 𝒜^p_. → e^(p γ α)};
l2U = {e^(c_. b_i_+d_.) :> B_i^(c/(ħ γ)) e^d, e^(c_. b+d_.) :> B^(c/(ħ γ)) e^d,
   e^(c_. t_i_+d_.) :> T_i^(c/ħ) e^d, e^(c_. t+d_.) :> T^(c/ħ) e^d,
   e^(c_. α_i_+d_.) :> 𝒜_i^(c/γ) e^d, e^(c_. α+d_.) :> 𝒜^(c/γ) e^d,
   e^ℰ_ :> e^(Expand@ℰ)};
SS[ℰ_, op_] := Collect[
   Normal@Series[If[$p > 0, ℰ, ℰ /. U2l], {ħ, 0, $p}],
   ħ, op];
SS[ℰ_] := SS[ℰ, Together];
SST[ℰ_, op___] := SS[ℰ /. U2l, op];
Simp[ℰ_, op_] := Collect[ℰ, _CU | _QU, op];
Simp[ℰ_] := Simp[ℰ, SS[#, Expand] &];
SimpT[ℰ_] := Collect[ℰ, _CU | _QU, SST[#, Expand] &];
Kδ /: Kδ_i_,j_ := If[i === j, 1, 0];
c_Integer_k_Integer := c + O[ϵ]^(k+1);
```

CF

In[•]:=
```
CF[ℰ_] := ExpandDenominator@
   ExpandNumerator@Together[Expand[ℰ] //. e^x_ e^y_ :> e^(x+y) /. e^x_ :> e^CF[x]];
```

SeriesData

In[•]:=
```
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];
```

Self-Pair (SP):

SP

In[•]:=
```
SP{}[P_] := P; SP{ξ_→x_,ps___}[P_] := Expand[P // SP{ps}] /. f_. ξ^d_. :↦ ∂{x,d} f
```

DeclareAlgebra

QLImplementation

In[•]:=
```
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];
```

QLImplementation

In[•]:=

```
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];
  𝒪_U[specs___, 𝔼[L_, Q_, P_]] := 𝒪_U[specs, SS@Normal[P ℮^(L+Q)]];
  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[])]; ]
```

DeclareMorphism

QLImplementation

```
In[•]:= 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; )
```

Meta-Operations

QLImplementation

```
In[•]:= 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];
```

Implementing CU = $\mathcal{U}(\mathrm{sl}_2^{\gamma\epsilon})$

Verify σ and Δ! Also Generalize Δ to $\Delta_{i,j_1,j_2,\dots}$.

CU

```
In[•]:= 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};
```

Implementing QU = $\mathcal{U}_q(\mathrm{sl}_2^{\gamma\epsilon})$

QU

```
In[•]:= 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};
```

Implementing θ

theta

```
DeclareMorphism[Cθ, CU → CU,
 {y → -x_CU, a → -a_CU, x → -y_CU}, {t_i_ → -t_i, T_i_ → T_i^-1, 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_i_ → -t_i, T_i_ → T_i^-1, t → -t, T → T^-1}]
```

The Asymmetric Dequantizator

Following pensieve://People/VanDerVeen/Dequant1.pdf.

ADeq

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

ADeq

In[ ]:=
```
AD$ω = γ CU[y, x] + ϵ CU[a, a] - (t - γ ϵ) CU[a];
```

ADeq

In[ ]:=
```
DeclareMorphism[AD, QU → CU,
 {a → a_CU, x → CU@x, y :> S_CU[SS[AD$f], a → a_CU, ω → AD$ω] ** y_CU}]
```

The Symmetric Dequantizator

Following pensieve://People/VanDerVeen/Dequant1.pdf.

SDeq

In[ ]:=
$$SD\$g = \sqrt{\left(\left(2\gamma \left(\text{Cosh}\left[\frac{\hbar}{2}\sqrt{t^2 + \gamma^2\epsilon^2 + 4\epsilon\varpi}\,\right] - \text{Cosh}\left[\frac{t - \epsilon\gamma - 2\epsilon a}{2/\hbar}\right] \right) \right) \Big/ \left(\text{Sinh}\left[\frac{\gamma\epsilon\hbar}{2}\right] \left(t\left(2a+\gamma\right) - 2a\left(a+\gamma\right)\epsilon + 2\varpi\right)\hbar \right) \right)};$$

SDeq

In[ ]:=
```
SD$f = Simplify[e^ħ (t/2-ϵ a) (SD$g /. {a → -a, t → -t})];
```

SDeq

In[ ]:=
```
SD$ϖ = γ CU[y, x] + ϵ CU[a, a] - (t - γ ϵ) CU[a] - t γ 1_CU / 2;
```

SDeq

```
In[•]:= DeclareMorphism[SD, QU → CU, {a → aCU,
  x :⇢ SCU[SS[SD$f], a → aCU, ϖ → SD$ϖ] ** xCU,
  y :⇢ SCU[SS[SD$g], a → aCU, ϖ → SD$ϖ] ** yCU }]
```

The representation ρ

rho

```
ρ@yCU = ρ@yQU = ((0 0) (ϵ 0)); ρ@aCU = ρ@aQU = ((γ 0) (0 0));
ρ@xCU = ((0 γ) (0 0)); ρ@xQU = ((0 (1 - e^(-γ ϵ ħ)) / (ϵ ħ)) (0 0));
ρ[e^ℰ_] := MatrixExp[ρ[ℰ]];
ρ[ℰ_] := (ℰ /. U2l /. t → γ ϵ /. (U : CU | QU)[u___] :⇢ Fold[Dot, ((1 0) (0 1)), ρ /@ U /@ {u}])
```

tSW

Logoi from Pensieve://Talks/Toulouse-1705/DogmaDemo.nb and from Pensieve://Talks/Sydney-1708/ExtraDetails@@.nb.

Goal. In either U, compute $F = e^{-\eta y}\, e^{\xi x}\, e^{\eta y}\, e^{-\xi x}$. First compute $G = e^{\xi x}\, y e^{-\xi x}$, a finite sum. Now F satisfies the ODE $\partial_\eta F = \partial_\eta\left(e^{-\eta y}\, e^{\eta G}\right) = -yF + FG$ with initial conditions $F(\eta = 0) = 1$. So we set it up and solve:

tSW

```
In[•]:= SWxy[U_, kk_] :=
  SWxy[U, kk] = Block[{$U = U, $k = kk, $p = kk}, Module[{G, F, fs, f, bs, e, b, es},
    G = Simp[Table[ξ^k / k!, {k, 0, $k + 1}].NestList[Simp[B[xU, #]] &, yU, $k + 1]];
    fs = Flatten@Table[f1,i,j,k[η], {l, 0, $k}, {i, 0, l}, {j, 0, l}, {k, 0, l}];
    F = fs. (bs = fs /. fL_,i_,j_,k_[η] :⇢ ϵ^l U@{y^i, a^j, x^k});
    es = Flatten[
      Table[Coefficient[e, b] == 0, {e, {F - 1U /. η → 0, F ** G - yU ** F - ∂η F}}, {b, bs}]];
    F = F /. DSolve[es, fs, η][[1]];
    E[0,
      ξ x + η y + (U /. {CU → -t η ξ, QU → η ξ (1 - T) / ħ}),
      F + 0$k /. {e- → 1, U → Times}
     ] /. (v : η | ξ | t | T | y | a | x) → v1
   ]];
tSWxy_,i_,j_→k_ := SWxy[$U, $k] /. {ξ1 → ξi, η1 → ηj, (v : t | T | y | a | x)1 → vk};
tSWxa,i_,j_→k_ := E[αj ak, e^(-γ αj) ξi xk, 1];
tSWay,i_,j_→k_ := E[αi ak, e^(-γ αi) ηj yk, 1];
```

R in QU.

The Faddeev-Quesne formula:

Faddeev

In[ ]:=
```
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]
```

R

In[ ]:=
```
QU[R_i_,j_] := O_QU[{y_1, a_1}_i, {a_2, x_2}_j, SS[e^(ħ b_1 a_2) e_qħ[ħ y_1 x_2] /. b_1 → γ^-1 (ϵ a_1 - t_i)]];
QU[R_i_^-1,j_] := S_j@QU[R_i,j];
```

Exponentials as needed.

Exp

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 $e^{\xi\,\mathbb{O}(P)} = \mathbb{O}\big(e^{\xi P_0} F(\xi)\big)$, then $F(\xi=0)=1$ and we have:
$\mathbb{O}\big(e^{\xi P_0}(P_0 F(\xi) + \partial_\xi F)\big) = \mathbb{O}\big(\partial_\xi e^{\xi P_0} F(\xi)\big) = \partial_\xi \mathbb{O}\big(e^{\xi P_0} F(\xi)\big) = \partial_\xi e^{\xi\,\mathbb{O}(P)} = e^{\xi\,\mathbb{O}(P)}\,\mathbb{O}(P) = \mathbb{O}\big(e^{\xi P_0} F(\xi)\big)\,\mathbb{O}(P)$.
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.

Exp

In[ ]:=
```
(* 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_] := Module[{LQ = Normal@P /. ϵ → 0},
   𝔼[ξ LQ /. (x | y)_i → 0, ξ LQ /. (t | a)_i → 0, 1]];
Exp_U_i_,k_[ξ_, P_] := Block[{$U = U, $k = k},
  Module[{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 @@ {y_i, a_i, x_i}^{js});
   rhs = Normal@
     Last@m_i,j→i[𝔼[ξ P0 /. (x | y)_i → 0, ξ P0 /. (t | a)_i → 0, F + 0_k] m_i→j@𝔼[0, 0, P + 0_k]];
   at0 = (# == 0) & /@ Flatten@CoefficientList[F - 1 /. ξ → 0, {y_i, a_i, x_i}];
   atξ = (# == 0) & /@ Flatten@CoefficientList[(∂_ξ F) + P0 F - rhs, {y_i, a_i, x_i}];
   𝔼[ξ P0 /. (x | y)_i → 0, ξ P0 /. (t | a)_i → 0, F + 0_k] /.
    DSolve[And @@ (at0 ⋃ atξ), φs, ξ][[1]]]]
```

Zip and Bind

E

In[ ]:=
```
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];
```

Zip

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

Zip

```
In[•]:= Zip_{}[P_] := P;  Zip_{ξ_,ξs___}[P_] := (Expand[P // Zip_{ξs}] /. f_. ξ^d_. :↦ ∂_{ξ*,d} f) /. ξ* → 0
```

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

Zip

```
QZip_{ξs_List,simp_}@𝔼[L_, Q_, P_] := Module[{ξ, z, zs, c, ys, ηs, qt, zrule, Q1, Q2},
   zs = Table[ξ*, {ξ, ξs}];
   c = Q /. Alternatives @@ (ξs ⋃ zs) → 0;
   ys = Table[∂_ξ (Q /. Alternatives @@ zs → 0), {ξ, ξs}];
   ηs = Table[∂_z (Q /. Alternatives @@ ξs → 0), {z, zs}];
   qt = Inverse@Table[Kδ_{z,ξ*} - ∂_{z,ξ} Q, {ξ, ξs}, {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_ξs[e^Q1 (P /. zrule)]] ];
QZip_{ξs_List} := QZip_{ξs,CF};
```

LZip implements the "L-level zips" on $\mathbb{E}(L, Q, P) = \mathrm{P}e^{L+Q}$. Such zips regard all of $\mathrm{P}e^{Q}$ as a single"P". Here the z's are t and α and the ζ's are τ and a.

Zip

```
LZip_{ξs_List,simp_}@𝔼[L_, Q_, P_] := Module[{ξ, z, zs, c, ys, ηs, lt, zrule, L1, L2, Q1, Q2},
   zs = Table[ξ*, {ξ, ξs}];
   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)];
   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_ξs[e^L1+Q1 (P /. U2l /. zrule)]] //. l2U ];
LZip_{ξs_List} := LZip_{ξs,CF};
```

Bind

```
Bind_{}[L_, R_] := L R;
Bind_{is__}[L_𝔼, R_𝔼] := 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[{x_{n@i},η_{n@i}},{i,{is}}]} ];
B_{l_List}[L_, R_] := Bind_l[L, R];  B_{is___}[L_, R_] := Bind_{is}[L, R];
```

Tensorial Representations

t1

In[•]:=
```
tη = t1 = 𝔼[0, 0, 1 + 0$k];
```

tm

In[•]:=
```
tm_{i_,j_→k_} := Module[{tk},
    𝔼[(τ_i + τ_j) t_k + α_i a_k + α_j a_k, η_i y_k + ξ_j x_k, 1]
     (tSW_{xy,i,j→tk} /. {t_tk → t_k, T_tk → T_k, y_tk → ℯ^{-γ α_i} y_k, a_tk → a_k, x_tk → ℯ^{-γ α_j} x_k})];
m_{j_→k_}[ℰ_, 𝔼] := ℰ ~B_{j,k}~ tm_{j,k→k};
```

In[•]:=
```
tm_{1,2→3}
```

Out[•]=
$$\mathbb{E}\Big[a_3\,\alpha_1 + a_3\,\alpha_2 + t_3\,(\tau_1+\tau_2),\ y_3\,\eta_1 + \mathrm{e}^{-\gamma\alpha_1}\,y_3\,\eta_2 + \mathrm{e}^{-\gamma\alpha_2}\,x_3\,\xi_1 + \frac{\left(1-T_3\right)\,\eta_2\,\xi_1}{\hbar} + x_3\,\xi_2,$$
$$1 + \frac{1}{4\,\hbar}\,\eta_2\,\xi_1\,\left(8\,\hbar\,a_3\,T_3 + 4\,\mathrm{e}^{-\gamma\alpha_1-\gamma\alpha_2}\,\gamma\,\hbar^2\,x_3\,y_3 + 2\,\mathrm{e}^{-\gamma\alpha_1}\,\gamma\,\hbar\,y_3\,\eta_2 - 6\,\mathrm{e}^{-\gamma\alpha_1}\,\gamma\,\hbar\,T_3\,y_3\,\eta_2 + \right.$$
$$\left. 2\,\mathrm{e}^{-\gamma\alpha_2}\,\gamma\,\hbar\,x_3\,\xi_1 - 6\,\mathrm{e}^{-\gamma\alpha_2}\,\gamma\,\hbar\,T_3\,x_3\,\xi_1 + \gamma\,\eta_2\,\xi_1 - 4\,\gamma\,T_3\,\eta_2\,\xi_1 + 3\,\gamma\,T_3^2\,\eta_2\,\xi_1\right)\,\epsilon + 0[\epsilon]^2\Big]$$

tS

In[•]:=
```
S[U_, kk_] := S[U, kk] = Module[{OE},
    OE = m_{3,2,1→1}[Exp_{QU_1,$k}[η, S_1[QU[y_1]] /. QU → Times]
        Exp_{QU_2,$k}[α, S_2[QU[a_2]] /. QU → Times] Exp_{QU_3,$k}[ξ, S_3[QU[x_3]] /. QU → Times]];
    𝔼[-t_1 τ_1 + OE[[1]], OE[[2]], OE[[3]]] /. {η → η_1, α → α_1, 𝒜 → 𝒜_1, ξ → ξ_1}];
tS_i_ := S[$U, $k] /. {(v : τ | η | α | 𝒜 | ξ)_1 → v_i, (v : t | T | y | a | x)_1 → v_i};
```

In[•]:=
```
tS_1
```

Out[•]=
$$\mathbb{E}\Big[-a_1\,\alpha_1 - t_1\,\tau_1,\ \frac{1}{\hbar\,T_1}\left(-\mathrm{e}^{\gamma\alpha_1}\,\hbar\,y_1\,\eta_1 - \mathrm{e}^{\gamma\alpha_1}\,\hbar\,T_1\,x_1\,\xi_1 + \mathrm{e}^{\gamma\alpha_1}\,\eta_1\,\xi_1 - \mathrm{e}^{\gamma\alpha_1}\,T_1\,\eta_1\,\xi_1\right),$$
$$1 + \frac{1}{4\,\hbar\,T_1^2}\,\left(4\,\mathrm{e}^{\gamma\alpha_1}\,\gamma\,\hbar^2\,T_1\,y_1\,\eta_1 - 4\,\mathrm{e}^{\gamma\alpha_1}\,\hbar^2\,a_1\,T_1\,y_1\,\eta_1 - 2\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar^2\,y_1^2\,\eta_1^2 - 4\,\mathrm{e}^{\gamma\alpha_1}\,\hbar^2\,a_1\,T_1^2\,x_1\,\xi_1 - \right.$$
$$4\,\mathrm{e}^{\gamma\alpha_1}\,\gamma\,\hbar\,T_1\,\eta_1\,\xi_1 + 8\,\mathrm{e}^{\gamma\alpha_1}\,\hbar\,a_1\,T_1\,\eta_1\,\xi_1 + 4\,\mathrm{e}^{\gamma\alpha_1}\,\gamma\,\hbar\,T_1^2\,\eta_1\,\xi_1 - 4\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar^2\,T_1\,x_1\,y_1\,\eta_1\,\xi_1 +$$
$$6\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar\,y_1\,\eta_1^2\,\xi_1 - 2\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar\,T_1\,y_1\,\eta_1^2\,\xi_1 - 2\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar^2\,T_1^2\,x_1^2\,\xi_1^2 + 6\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar\,T_1\,x_1\,\eta_1\,\xi_1^2 -$$
$$\left.2\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\hbar\,T_1^2\,x_1\,\eta_1\,\xi_1^2 - 3\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,\eta_1^2\,\xi_1^2 + 4\,\mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,T_1\,\eta_1^2\,\xi_1^2 - \mathrm{e}^{2\,\gamma\alpha_1}\,\gamma\,T_1^2\,\eta_1^2\,\xi_1^2\right)\,\epsilon + 0[\epsilon]^2\Big]$$

tDelta

In[•]:=
```
Δ[U_, kk_] := Δ[U, kk] = Module[{OE},
    OE = Block[{$k = kk, $p = kk + 1},
      m_{1,3,5→1}@m_{2,4,6→2}@Times[ (* Warning: wrong unless $p≥$k+1! *)
          ReplacePart[1 → 0]@Exp_{QU_1,$k}[η, Δ_{1→1,2}[QU[y_1]] /. QU → Times],
          ReplacePart[2 → 0]@Exp_{QU_3,$k}[α, Δ_{3→3,4}[QU[a_3]] /. QU → Times],
          ReplacePart[1 → 0]@Exp_{QU_5,$k}[ξ, Δ_{5→5,6}[QU[x_5]] /. QU → Times]
        ] /. {η → η_1, α → α_1, ξ → ξ_1}];
    𝔼[τ_1 (t_1 + t_2) + α_1 (a_1 + a_2), OE[[2]], OE[[3]]]];
tΔ_{i_→j_,k_} :=
  Δ[$U, $k] /. {(v : τ | η | α | ξ)_1 → v_i, (v : t | T | y | a | x)_1 → v_j, (v : t | T | y | a | x)_2 → v_k};
```

```
In[•]:= tΔ1→1,2
```

```
Out[•]= 𝔼[ (a1 + a2) α1 + (t1 + t2) τ1, y1 η1 + T1 y2 η1 + x1 ξ1 + x2 ξ1,
    1 + 1/2 (-2 ħ a1 T1 y2 η1 + γ ħ T1 y1 y2 η1^2 - 2 ħ a1 x2 ξ1 + γ ħ x1 x2 ξ1^2) ϵ + O[ϵ]^2 ]
```

tR

```
In[•]:= ℭQU,k_[Ri_,j_] := ℭQU[{yi, ai, xi}i, {yj, aj, xj}j, -ħ γ^-1 ti aj + ħ yi xj,
            Series[e^(ħ γ^-1 ti aj - ħ yi xj) (e^(ħ bi aj) eqħ,k[ħ yi xj] /. bi → γ^-1 (ϵ ai - ti)), {ϵ, 0, k}]];
        R[QU, kk_] := R[QU, kk] = Module[{OE},
            OE = Simplify /@ ℭQU,kk@R1,2;
            𝔼[-ħ a2 t1/γ, ħ x2 y1, Last@OE] ];
        tRi_,j_ := R[$U, $k] /. {(v : t | T | y | a | x)1 → vi, (v : t | T | y | a | x)2 → vj};
        tR̄i_,j_ := tR̄i,j = tRi,j ~ Bj ~ tSj;
```

```
In[•]:= {tR1,2, tR̄1,2}
```

```
Out[•]= {𝔼[-ħ a2 t1/γ, ħ x2 y1, 1 + (ħ a1 a2/γ - 1/4 γ ħ^3 x2^2 y1^2) ϵ + O[ϵ]^2], 𝔼[ħ a2 t1/γ, -ħ x2 y1/T1,
    1 - 1/(4 (γ T1^2)) (ħ (4 a1 T1 (a2 T1 + γ ħ x2 y1) + γ ħ x2 y1 (4 a2 T1 + 3 γ ħ x2 y1))) ϵ + O[ϵ]^2]}
```

tC is the counterclockwise spinner; $\overline{\text{tC}}$ is its inverse.

tC

```
In[•]:= tCi_ := 𝔼[0, 0, T_i^(1/2) e^(-ϵ ai ħ) + 0$k];
        tC̄i_ := 𝔼[0, 0, T_i^(-1/2) e^(ϵ ai ħ) + 0$k];
```

```
In[•]:= Block[{$k = 3}, {tC1, tC̄2}]
```

```
Out[•]= {𝔼[0, 0, √T1 - ħ a1 √T1 ϵ + 1/2 ħ^2 a1^2 √T1 ϵ^2 - 1/6 (ħ^3 a1^3 √T1) ϵ^3 + O[ϵ]^4],
    𝔼[0, 0, 1/√T2 + ħ a2 ϵ/√T2 + ħ^2 a2^2 ϵ^2/(2 √T2) + ħ^3 a2^3 ϵ^3/(6 √T2) + O[ϵ]^4]}
```

tKink

```
In[•]:= Kink[QU, kk_] := Kink[QU, kk] = Block[{$k = kk}, (tR1,3 tC̄2) ~ B1,2 ~ tm1,2→1 ~ B1,3 ~ tm1,3→1];
        tKinki_ := Kink[$U, $k] /. {(v : t | T | y | a | x)1 → vi};
        K̄ink[QU, kk_] := K̄ink[QU, kk] = Block[{$k = kk}, (tR̄1,3 tC2) ~ B1,2 ~ tm1,2→1 ~ B1,3 ~ tm1,3→1];
        tK̄inki_ := K̄ink[$U, $k] /. {(v : t | T | y | a | x)1 → vi}
```

Alternative Algorithms

AltLogos

In[ ]:=
```
λalt,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}]]];
```

Asides

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

Aside

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