

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[ $\mathcal{E}$ _] := (Exponent[Normal@ $\mathcal{E}$  /.
  { $a \rightarrow a / \epsilon$ ,  $a_{i-} \rightarrow a_i / \epsilon$ ,  $(u : x | y) \Rightarrow \epsilon^{-1/2} u$ ,  $(u : x | y)_{i-} \Rightarrow \epsilon^{-1/2} u_i$ },  $\epsilon$ , Min]  $\geq 0$ );
```

Initialization / Utilities

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

TD

```
$p = 2; $k = 1; $U = QU; $E := {$k, $p};
$trim := { $\hbar^{p-}$  /;  $p > $p \rightarrow 0$ ,  $\epsilon^{k-}$  /;  $k > $k \rightarrow 0$ };
SetAttributes[{SS, SST}, HoldAll];
 $q_{\hbar} = e^{\gamma \epsilon \hbar}$ ;
(* Upper to lower and lower to Upper: *)
U21 = { $B_{i-}^{p-} \rightarrow e^{-p \hbar \gamma b_i}$ ,  $B_{-}^{p-} \rightarrow e^{-p \hbar \gamma b}$ ,  $T_{i-}^{p-} \rightarrow e^{p \hbar t_i}$ ,  $T_{-}^{p-} \rightarrow e^{p \hbar t}$ ,  $\mathcal{A}_{i-}^{p-} \rightarrow e^{p \gamma \alpha_i}$ ,  $\mathcal{A}_{-}^{p-} \rightarrow e^{p \gamma \alpha}$ };
L2U = { $e^{c- \cdot b_i + d-} \Rightarrow B_{i-}^{c / (\hbar \gamma)} e^d$ ,  $e^{c- \cdot b + d-} \Rightarrow B^{-c / (\hbar \gamma)} e^d$ ,
   $e^{c- \cdot t_i + d-} \Rightarrow T_{i-}^{c / \hbar} e^d$ ,  $e^{c- \cdot t + d-} \Rightarrow T^{c / \hbar} e^d$ ,
   $e^{c- \cdot \alpha_i + d-} \Rightarrow \mathcal{A}_{i-}^{c / \gamma} e^d$ ,  $e^{c- \cdot \alpha + d-} \Rightarrow \mathcal{A}^{c / \gamma} e^d$ ,
   $e^{\mathcal{E}-} \Rightarrow e^{\text{Expand@}\mathcal{E}}$ };
SS[ $\mathcal{E}$ _, op_] := Collect[
  Normal@Series[If[$p > 0,  $\mathcal{E}$ ,  $\mathcal{E}$  /. U21], { $\hbar$ , 0, $p}],
   $\hbar$ , op];
SS[ $\mathcal{E}$ _] := SS[ $\mathcal{E}$ , Together];
SST[ $\mathcal{E}$ _, op___] := SS[ $\mathcal{E}$  /. U21, op];
Simp[ $\mathcal{E}$ _, op_] := Collect[ $\mathcal{E}$ , _CU | _QU, op];
Simp[ $\mathcal{E}$ _] := Simp[ $\mathcal{E}$ , SS[#, Expand] &];
SimpT[ $\mathcal{E}$ _] := Collect[ $\mathcal{E}$ , _CU | _QU, SST[#, Expand] &];
K $\delta$  /: K $\delta_{i-, j-}$  := If[i == j, 1, 0];
 $c\_Integer_{k\_Integer} := c + 0[\epsilon]^{k+1}$ ;
```

CF

```
In[*]:= CF[ $\mathcal{E}$ _] := ExpandDenominator@
  ExpandNumerator@Together[Expand[ $\mathcal{E}$ ] //.  $e^{x-} e^{y-} \Rightarrow e^{x+y}$  /.  $e^{x-} \Rightarrow e^{\text{CF}[x]}$ ];
```

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 . ξd . => ∂{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, gi_ -> {i, k}}, {g, gs}]; (* sorting -> *)
  cp = Alternatives @@ cs; (* cents *)
  SetAttributes[M, HoldRest]; M[0, _] = 0; M[a_, x_] := a x;
  CE[_] := Collect[_U, Expand] /. $trim;
  Ui[_] := _ /. {t : cp -> ti, u_U -> (#i &) /@u};
  Ui[NCM[]] = pow[_] = U@{ } = 1_U = U[];
  B[U@(x_)i_, U@(y_)i_] := Ui@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[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[_NonCommutativeMultiply] := U /@ _;
  OU[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];
  OU[specs___, E[L_, Q_, P_]] := OU[specs, SS@Normal[P e^{L+Q}]];
  pow[_] := pow[_] ** _;
  SU[_] := CE@Total[
    CoefficientRules[_] /. (p_ -> c_) -> c NCM@@MapThread[pow, {Last /@ {ss}, p}];
    sigma_rs___[c_. * u_U] := (c /. (t : cp)j_ -> tj /. {rs}) U[List@@(u /. v_j_ -> vj /. {rs})];
    mj_ -> k_ [c_. * u_U] := CE[(c /. (t : cp)j_ -> tk) DeleteCases[u, _j|k]] **
      U@@Cases[u, w_j -> wk] ** U@@Cases[u, _k];
    U /: c_. * u_U * v_U := CE[c u ** v];
    Si[c_. * u_U] := CE[(c /. Si[U, Centrals]) DeleteCases[u, _i]] **
      Ui[NCM@@Reverse@Cases[u, x_i -> S@U@x]];
    Delta_i -> j_, k_ [c_. * u_U] := CE[(c /. Delta_i -> j_, k [U, Centrals]) DeleteCases[u, _i]] **
      (NCM@@Cases[u, x_i -> sigma_1 -> j_, 2 -> k @Delta@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[ ]:= sigma_rs[_E_Plus] := sigma_rs /@ _E;
m_j_to_j_ = Identity; m_j_to_k_[0] = 0;
m_j_to_k_[_E_Plus] := Simp[m_j_to_k_ /@ _E];
m_is_...i_j_to_k_[_E_] := m_j_to_k_@m_is_i_to_j@_E;
S_i[_E_Plus] := Simp[S_i /@ _E];
Delta_is[_E_Plus] := Simp[Delta_is /@ _E];
```

Implementing $CU = \mathcal{U}(sl_2^{\epsilon})$

Verify σ and Δ ! Also Generalize Δ to $\Delta_{i_1 j_2, \dots}$

CU

```
In[ ]:= DeclareAlgebra[CU, Generators -> {y, a, x}, Centrals -> {t}];
B[a_CU, y_CU] = -gamma y_CU; B[x_CU, a_CU] = -gamma x_CU;
B[x_CU, y_CU] = 2 epsilon 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};
Delta@y_CU = CU@y_1 + CU@y_2; Delta@a_CU = CU@a_1 + CU@a_2; Delta@x_CU = CU@x_1 + CU@x_2;
Delta_i_to_j_k[CU, Centrals] = {t_i -> t_j + t_k};
```

Implementing $QU = \mathcal{U}_q(sl_2^{\epsilon})$

QU

```
In[ ]:= DeclareAlgebra[QU, Generators -> {y, a, x}, Centrals -> {t, T}];
B[a_QU, y_QU] = -gamma y_QU; B[x_QU, a_QU] = -gamma QU@x;
B[x_QU, y_QU] := SS[q_h - 1] QU@{y, x} + O_QU[{a}, SS[(1 - T e^{-2 epsilon a h}) / h]];
(S@y_QU := O_QU[{a, y}, SS[-T^{-1} e^{h epsilon a} y]]; S@a_QU = -a_QU; S@x_QU := O_QU[{a, x}, SS[-e^{h epsilon a} x]]);
S_i[QU, Centrals] = {t_i -> -t_i, T_i -> T_i^{-1}};
Delta@y_QU := O_QU[{y_1, a_1}_1, {y_2}_2, SS[y_1 + T_1 e^{-h epsilon a_1} y_2]];
Delta@a_QU = QU@a_1 + QU@a_2; Delta@x_QU := O_QU[{a_1, x_1}_1, {x_2}_2, SS[x_1 + e^{-h epsilon a_1} x_2]];
Delta_i_to_j_k[QU, Centrals] = {t_i -> t_j + t_k, T_i -> T_j T_k};
```

Implementing θ

theta

```
In[*]:= DeclareMorphism[C $\theta$ , CU  $\rightarrow$  CU, {y  $\rightarrow$  -xCU, a  $\rightarrow$  -aCU, x  $\rightarrow$  -yCU}, {t  $\rightarrow$  -t, T  $\rightarrow$  T-1}}];
DeclareMorphism[Q $\theta$ , QU  $\rightarrow$  QU, {y  $\mapsto$  0QU[{a, x}], SS[-T-1/2 e $\hbar \epsilon a$  x]},
a  $\rightarrow$  -aQU, x  $\mapsto$  0QU[{a, y}], SS[-T-1/2 e $\hbar \epsilon a$  y]}], {t  $\rightarrow$  -t, T  $\rightarrow$  T-1}}]
```

The Asymmetric Dequantizator

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

A_{Deq}

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

A_{Deq}

```
In[*]:= AD$ $\omega$  =  $\gamma$  CU[y, x] +  $\epsilon$  CU[a, a] - (t -  $\gamma \epsilon$ ) CU[a];
```

A_{Deq}

```
In[*]:= DeclareMorphism[AD, QU  $\rightarrow$  CU,
{a  $\rightarrow$  aCU, x  $\rightarrow$  CU@x, y  $\mapsto$  SCU[SS[AD$f], a  $\rightarrow$  aCU,  $\omega \rightarrow$  AD$ $\omega$ ] ** yCU}]
```

The Symmetric Dequantizator

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

S_{Deq}

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

S_{Deq}

```
In[*]:= SD$f = Simplify[e $\hbar (t/2 - \epsilon a)$  (SD$g /. {a  $\rightarrow$  -a, t  $\rightarrow$  -t})];
```

S_{Deq}

```
In[*]:= SD$ $\omega$  =  $\gamma$  CU[y, x] +  $\epsilon$  CU[a, a] - (t -  $\gamma \epsilon$ ) CU[a] - t  $\gamma$  1CU / 2;
```

SDeq

```
In[ ]:= DeclareMorphism[SID, QU -> CU, {a -> a_CU,
  x := S_CU[SS[SID$f], a -> a_CU, w -> SID$w] ** x_CU,
  y := S_CU[SS[SID$g], a -> a_CU, w -> SID$w] ** y_CU }]
```

The representation ρ

rho

```
 $\rho@y_{CU} = \rho@y_{QU} = \begin{pmatrix} 0 & 0 \\ \epsilon & 0 \end{pmatrix}; \rho@a_{CU} = \rho@a_{QU} = \begin{pmatrix} \gamma & 0 \\ 0 & 0 \end{pmatrix};$ 
 $\rho@x_{CU} = \begin{pmatrix} 0 & \gamma \\ 0 & 0 \end{pmatrix}; \rho@x_{QU} = \begin{pmatrix} 0 & (1 - e^{-\gamma \epsilon \hbar}) / (\epsilon \hbar) \\ 0 & 0 \end{pmatrix};$ 
 $\rho[e^{\xi}] := \text{MatrixExp}[\rho[\xi]];
\rho[\xi] := (\xi /. U21 /. t -> \gamma \epsilon /. (U : CU | QU) [u___] := Fold[Dot, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \rho / @ U / @ \{u\}])$ 
```

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 $\partial_\eta F = \partial_\eta (e^{-\eta y} e^{\eta G}) = -yF + FG$ with initial conditions $F(\eta = 0) = 1$. So we set it up and solve:

tSW

```
In[ ]:= SW_xy[U_, kk_] :=
  SW_xy[U, kk] = Block[{ $U = U, $k = kk, $p = kk }, Module[{ G, F, fs, f, bs, e, b, es },
    G = Simp[Table[\xi^k / k!, {k, 0, $k + 1}].NestList[Simp[B[x_U, #]] &, y_U, $k + 1]];
    fs = Flatten@Table[f_{1,i,j,k}[\eta], {1, 0, $k}, {i, 0, 1}, {j, 0, 1}, {k, 0, 1}];
    F = fs.(bs = fs /. f_{L_,i_,j_,k_}[\eta] := e^{L U} @ {y^i, a^j, x^k});
    es = Flatten[
      Table[Coefficient[e, b] == 0, {e, {F - 1_U /. \eta -> 0, F ** G - y_U ** F - \partial_\eta F}}, {b, bs}]];
    F = F /. DSolve[es, fs, \eta][[1]];
    E[0,
      \xi x + \eta y + (U /. {CU -> -t \eta \xi, QU -> \eta \xi (1 - T) / \hbar}),
      F + 0_{$k} /. {e -> 1, U -> Times}
    ] /. (v : \eta | \xi | t | T | y | a | x) -> v_1
  ]];
tSW_xy_{i_,j_ -> k_} := SW_xy[$U, $k] /. {\xi_1 -> \xi_i, \eta_1 -> \eta_j, (v : t | T | y | a | x)_1 -> v_k};
tSW_xa_{i_,j_ -> k_} := E[\alpha_j a_k, e^{-\gamma \alpha_j} \xi_i x_k, 1];
tSW_ay_{i_,j_ -> k_} := E[\alpha_i a_k, e^{-\gamma \alpha_i} \eta_j y_k, 1];
```

R in QU.

The Faddeev-Quesne formula:

Faddeev

$$\mathbf{e}_{q_-,k_-}[X_-] := \mathbf{e}^{\sum_{j=1}^{k_-+1} \frac{(1-q)^j X^j}{j(1-q^j)}}; \quad \mathbf{e}_{q_-}[X_-] := \mathbf{e}_{q_-,k_-}[X_-]$$

R

$$\begin{aligned} \mathbf{QU}[R_{i,j}] &:= \mathbf{OQU}[\{y_1, a_1\}_i, \{a_2, x_2\}_j, \mathbf{SS}[e^{\hbar b_1 a_2} \mathbf{e}_{q_{\hbar}}[\hbar y_1 x_2] /. b_1 \rightarrow \gamma^{-1} (\epsilon a_1 - t_i)]]; \\ \mathbf{QU}[R_{i,j}^{-1}] &:= \mathbf{S}_j @ \mathbf{QU}[R_{i,j}]; \end{aligned}$$

Exponentials as needed.

Exp

Task. Define $\text{Exp}_{U_i,k}[\xi, P]$ which computes $e^{\xi \mathbf{O}(P)}$ 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 $U @ \text{Exp}_{U_i,k}[\xi, P] == \mathbf{S}_U[e^{\xi X}, X \rightarrow \mathbf{O}(P)]$.

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

$$\mathbf{O}(e^{\xi P_0} (P_0 F(\xi) + \partial_{\xi} F)) = \mathbf{O}(\partial_{\xi} e^{\xi P_0} F(\xi)) = \partial_{\xi} \mathbf{O}(e^{\xi P_0} F(\xi)) = \partial_{\xi} e^{\xi \mathbf{O}(P_0)} = e^{\xi \mathbf{O}(P_0)} \mathbf{O}(P_0) = \mathbf{O}(e^{\xi P_0} F(\xi)) \mathbf{O}(P_0).$$

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

```
(* Bug: The first line is valid only if O(e^{P_0}) == e^{O(P_0)}. *)
(* Bug: xi must be a symbol. *)
Exp_{U_i,0}[\xi_-, P_-] := Module[{LQ = Normal@P /. \epsilon -> 0},
  E[\xi LQ /. (x | y)_i -> 0, \xi LQ /. (t | a)_i -> 0, 1]];
Exp_{U_i,k_-}[\xi_-, P_-] := Block[{$U = U, $k = k},
  Module[{P0, \varphi, \varphi_s, F, j, rhs, at0, at\xi},
    P0 = Normal@P /. \epsilon -> 0;
    \varphi_s =
      Flatten@Table[\varphi_{j1,j2,j3}[\xi], {j2, 0, k}, {j1, 0, 2k+1-j2}, {j3, 0, 2k+1-j2-j1}];
    F = Normal@Last@Exp_{U_i,k-1}[\xi, P] + \epsilon^k \varphi_s. (\varphi_s /. \varphi_{js}[\xi] -> Times@@{y_i, a_i, x_i}^{js});
    rhs = Normal@
      Last@m_{i,j -> i}[E[\xi P0 /. (x | y)_i -> 0, \xi P0 /. (t | a)_i -> 0, F + \theta_k] m_{i -> j} @ E[\theta, 0, P + \theta_k]];
    at0 = (# == 0) & /@ Flatten@CoefficientList[F - 1 /. \xi -> 0, {y_i, a_i, x_i}];
    at\xi = (# == 0) & /@ Flatten@CoefficientList[(\partial_{\xi} F) + P0 F - rhs, {y_i, a_i, x_i}];
    E[\xi P0 /. (x | y)_i -> 0, \xi P0 /. (t | a)_i -> 0, F + \theta_k] /.
      DSolve[And@@(at0 | at\xi), \varphi_s, \xi][[1]]]
```

Zip and Bind

E

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

Zip

```
In[*]:= Zip[_][P_] := P; Zip[_][P_] := (Expand[P // Zip[_]] /. 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.

Zip

```
QZip[_List,simp_]@E[L_, Q_, P_] := 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[_][e^{Q1} (P /. zrule)]];
  QZip[_List] := QZip[_List,CF];
```

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 t and α and the ζ ’s are τ and a .

Zip

```
LZip[_List,simp_]@E[L_, Q_, P_] := 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 /. U21 /. zrule) /. Alternatives @@ zs -> 0;
  simp /@ E[L2, Q2, Det[lt] e^{-L2-Q2} Zip[_][e^{L1+Q1} (P /. U21 /. zrule)]] // L2U];
  LZip[_List] := LZip[_List,CF];
```

Bind

```
Bind[_][L_, R_] := L R;
Bind[_][L_IE_, R_IE_] := 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}}]
  ] // LZippFlatten@Table[{β_{n+i}, τ_{n+i}, a_{n+i}}, {i, {is}}] // QZipFlatten@Table[{ξ_{n+i}, y_{n+i}}, {i, {is}}];
  B_L_List[L_, R_] := Bind_L[L, R]; B_IS_List[L_, R_] := Bind_IS[L, R];
```


Tensorial Representations

t1

```
In[ ]:= t $\eta$  = t $\mathbf{1}$  =  $\mathbb{E}[\mathbf{0}, \mathbf{0}, \mathbf{1} + \mathbf{0}_{\$k}]$ ;
```

tm

```
In[ ]:= tm $_{i,j \rightarrow k}$  := Module [{t $k$ },
   $\mathbb{E}[(\tau_i + \tau_j) \mathbf{t}_k + \alpha_i \mathbf{a}_k + \alpha_j \mathbf{a}_k, \eta_i \mathbf{y}_k + \xi_j \mathbf{x}_k, \mathbf{1}]$ 
  (t $SW_{xy, i, j \rightarrow tk}$  /. {t $_{tk} \rightarrow \mathbf{t}_k, T_{tk} \rightarrow T_k, \mathbf{y}_{tk} \rightarrow e^{-\gamma \alpha_i} \mathbf{y}_k, \mathbf{a}_{tk} \rightarrow \mathbf{a}_k, \mathbf{x}_{tk} \rightarrow e^{-\gamma \alpha_j} \mathbf{x}_k$ });
  m $_{j \rightarrow k}[\mathcal{E}_- \mathbb{E}] := \mathcal{E} \sim \mathbf{B}_{j, k} \sim \mathbf{tm}_{j, k \rightarrow k}$ ;
```

```
In[ ]:= tm $_{1, 2 \rightarrow 3}$ 
```

```
Out[ ]:=  $\mathbb{E}\left[\mathbf{a}_3 \alpha_1 + \mathbf{a}_3 \alpha_2 + \mathbf{t}_3 (\tau_1 + \tau_2), \mathbf{y}_3 \eta_1 + e^{-\gamma \alpha_1} \mathbf{y}_3 \eta_2 + e^{-\gamma \alpha_2} \mathbf{x}_3 \xi_1 + \frac{(1 - T_3) \eta_2 \xi_1}{\hbar} + \mathbf{x}_3 \xi_2,$ 
 $1 + \frac{1}{4 \hbar} \eta_2 \xi_1 (8 \hbar \mathbf{a}_3 T_3 + 4 e^{-\gamma \alpha_1 - \gamma \alpha_2} \gamma \hbar^2 \mathbf{x}_3 \mathbf{y}_3 + 2 e^{-\gamma \alpha_1} \gamma \hbar \mathbf{y}_3 \eta_2 - 6 e^{-\gamma \alpha_1} \gamma \hbar T_3 \mathbf{y}_3 \eta_2 +$ 
 $2 e^{-\gamma \alpha_2} \gamma \hbar \mathbf{x}_3 \xi_1 - 6 e^{-\gamma \alpha_2} \gamma \hbar T_3 \mathbf{x}_3 \xi_1 + \gamma \eta_2 \xi_1 - 4 \gamma T_3 \eta_2 \xi_1 + 3 \gamma T_3^2 \eta_2 \xi_1) \in + O[\epsilon]^2\right]$ 
```

tS

```
In[ ]:= S[U_, kk_] := S[U, kk] = Module [{OE},
  OE = m $_{3, 2, 1 \rightarrow 1}[\mathbf{Exp}_{QU_1, \$k}[\eta, S_1[QU[y_1]] /. QU \rightarrow Times]$ 
  Exp $_{QU_2, \$k}[\alpha, S_2[QU[a_2]] /. QU \rightarrow Times]$  Exp $_{QU_3, \$k}[\xi, S_3[QU[x_3]] /. QU \rightarrow Times]$ ];
   $\mathbb{E}[-\mathbf{t}_1 \tau_1 + \mathbf{OE}[[1]], \mathbf{OE}[[2]], \mathbf{OE}[[3]]] /. \{\eta \rightarrow \eta_1, \alpha \rightarrow \alpha_1, \mathcal{A} \rightarrow \mathcal{A}_1, \xi \rightarrow \xi_1\}$ ];
  tS $_i := S[\$U, \$k] /. \{(v : \tau | \eta | \alpha | \mathcal{A} | \xi)_1 \rightarrow v_i, (v : t | T | y | a | x)_1 \rightarrow v_i\}$ ;
```

```
In[ ]:= tS $_1$ 
```

```
Out[ ]:=  $\mathbb{E}\left[-\mathbf{a}_1 \alpha_1 - \mathbf{t}_1 \tau_1, \frac{1}{\hbar T_1} (-e^{\gamma \alpha_1} \hbar \mathbf{y}_1 \eta_1 - e^{\gamma \alpha_1} \hbar T_1 \mathbf{x}_1 \xi_1 + e^{\gamma \alpha_1} \eta_1 \xi_1 - e^{\gamma \alpha_1} T_1 \eta_1 \xi_1),$ 
 $1 + \frac{1}{4 \hbar T_1^2} (4 e^{\gamma \alpha_1} \gamma \hbar^2 T_1 \mathbf{y}_1 \eta_1 - 4 e^{\gamma \alpha_1} \hbar^2 \mathbf{a}_1 T_1 \mathbf{y}_1 \eta_1 - 2 e^{2\gamma \alpha_1} \gamma \hbar^2 \mathbf{y}_1^2 \eta_1^2 - 4 e^{\gamma \alpha_1} \hbar^2 \mathbf{a}_1 T_1^2 \mathbf{x}_1 \xi_1 -$ 
 $4 e^{\gamma \alpha_1} \gamma \hbar T_1 \eta_1 \xi_1 + 8 e^{\gamma \alpha_1} \hbar \mathbf{a}_1 T_1 \eta_1 \xi_1 + 4 e^{\gamma \alpha_1} \gamma \hbar T_1^2 \eta_1 \xi_1 - 4 e^{2\gamma \alpha_1} \gamma \hbar^2 T_1 \mathbf{x}_1 \mathbf{y}_1 \eta_1 \xi_1 +$ 
 $6 e^{2\gamma \alpha_1} \gamma \hbar \mathbf{y}_1 \eta_1^2 \xi_1 - 2 e^{2\gamma \alpha_1} \gamma \hbar T_1 \mathbf{y}_1 \eta_1^2 \xi_1 - 2 e^{2\gamma \alpha_1} \gamma \hbar^2 T_1^2 \mathbf{x}_1^2 \xi_1^2 + 6 e^{2\gamma \alpha_1} \gamma \hbar T_1 \mathbf{x}_1 \eta_1 \xi_1^2 -$ 
 $2 e^{2\gamma \alpha_1} \gamma \hbar T_1^2 \mathbf{x}_1 \eta_1 \xi_1^2 - 3 e^{2\gamma \alpha_1} \gamma \eta_1^2 \xi_1^2 + 4 e^{2\gamma \alpha_1} \gamma T_1 \eta_1^2 \xi_1^2 - e^{2\gamma \alpha_1} \gamma T_1^2 \eta_1^2 \xi_1^2) \in + O[\epsilon]^2\right]$ 
```

tDelta

```
In[ ]:=  $\Delta$ [U_, kk_] :=  $\Delta$ [U, kk] = Module [{OE},
  OE = Block [{$k = kk, $p = kk + 1},
  m $_{1, 3, 5 \rightarrow 1} @ \mathbf{m}_{2, 4, 6 \rightarrow 2} @ Times [ (* Warning: wrong unless $p \geq $k + 1! *)$ 
  ReplacePart [1  $\rightarrow$  0] @ Exp $_{QU_1, \$k}[\eta, \Delta_{1 \rightarrow 1, 2}[QU[y_1]] /. QU \rightarrow Times]$ ,
  ReplacePart [2  $\rightarrow$  0] @ Exp $_{QU_3, \$k}[\alpha, \Delta_{3 \rightarrow 3, 4}[QU[a_3]] /. QU \rightarrow Times]$ ,
  ReplacePart [1  $\rightarrow$  0] @ Exp $_{QU_5, \$k}[\xi, \Delta_{5 \rightarrow 5, 6}[QU[x_5]] /. QU \rightarrow Times]$ 
  ] /. { $\eta \rightarrow \eta_1, \alpha \rightarrow \alpha_1, \xi \rightarrow \xi_1$ };
   $\mathbb{E}[\tau_1 (\mathbf{t}_1 + \mathbf{t}_2) + \alpha_1 (\mathbf{a}_1 + \mathbf{a}_2), \mathbf{OE}[[2]], \mathbf{OE}[[3]]]$ ];
  t $\Delta$  $_{i \rightarrow j, k} :=$ 
   $\Delta$ [$U, $k] /. { $(v : \tau | \eta | \alpha | \xi)_1 \rightarrow v_i, (v : t | T | y | a | x)_1 \rightarrow v_j, (v : t | T | y | a | x)_2 \rightarrow v_k$ };
```

In[*]:= $\mathbf{t}\Delta_{1 \rightarrow 1,2}$

$$\text{Out[*]} = \mathbb{E} \left[(\mathbf{a}_1 + \mathbf{a}_2) \alpha_1 + (\mathbf{t}_1 + \mathbf{t}_2) \tau_1, \mathbf{y}_1 \eta_1 + \mathbf{T}_1 \mathbf{y}_2 \eta_1 + \mathbf{x}_1 \xi_1 + \mathbf{x}_2 \xi_1, \right. \\ \left. 1 + \frac{1}{2} \left(-2 \hbar \mathbf{a}_1 \mathbf{T}_1 \mathbf{y}_2 \eta_1 + \gamma \hbar \mathbf{T}_1 \mathbf{y}_1 \mathbf{y}_2 \eta_1^2 - 2 \hbar \mathbf{a}_1 \mathbf{x}_2 \xi_1 + \gamma \hbar \mathbf{x}_1 \mathbf{x}_2 \xi_1^2 \right) \epsilon + \mathcal{O}[\epsilon]^2 \right]$$

tR

```
In[*]:=  $\mathbb{C}_{\text{QU},k}[\mathbf{R}_{i,j}] := \mathbb{C}_{\text{QU}}[\{\mathbf{y}_i, \mathbf{a}_i, \mathbf{x}_i\}_i, \{\mathbf{y}_j, \mathbf{a}_j, \mathbf{x}_j\}_j, -\hbar \gamma^{-1} \mathbf{t}_i \mathbf{a}_j + \hbar \mathbf{y}_i \mathbf{x}_j, \\ \text{Series}[e^{\hbar \gamma^{-1} \mathbf{t}_i \mathbf{a}_j - \hbar \mathbf{y}_i \mathbf{x}_j} (e^{\hbar \mathbf{b}_i \mathbf{a}_j} e_{\mathbf{q}_n,k}[\hbar \mathbf{y}_i \mathbf{x}_j] /. \mathbf{b}_i \rightarrow \gamma^{-1} (\epsilon \mathbf{a}_i - \mathbf{t}_i)), \{\epsilon, \theta, k\}]]]; \\ \mathbf{R}[\text{QU}, k] := \mathbf{R}[\text{QU}, k] = \text{Module}[\{\text{OE}\}, \\ \text{OE} = \text{Simplify} / @ \mathbb{C}_{\text{QU},k}[\mathbf{R}_{1,2}]; \\ \mathbb{E}[-\frac{\hbar \mathbf{a}_2 \mathbf{t}_1}{\gamma}, \hbar \mathbf{x}_2 \mathbf{y}_1, \text{Last}@\text{OE}]]; \\ \mathbf{tR}_{i,j} := \mathbf{R}[\$U, \$k] /. \{(v : t | T | y | a | x)_1 \rightarrow v_i, (v : t | T | y | a | x)_2 \rightarrow v_j\}; \\ \overline{\mathbf{tR}}_{i,j} := \overline{\mathbf{tR}}_{i,j} = \mathbf{tR}_{i,j} \sim \mathbf{B}_j \sim \mathbf{tS}_j;$ 
```

In[*]:= $\{\mathbf{tR}_{1,2}, \overline{\mathbf{tR}}_{1,2}\}$

$$\text{Out[*]} = \left\{ \mathbb{E} \left[-\frac{\hbar \mathbf{a}_2 \mathbf{t}_1}{\gamma}, \hbar \mathbf{x}_2 \mathbf{y}_1, 1 + \left(\frac{\hbar \mathbf{a}_1 \mathbf{a}_2}{\gamma} - \frac{1}{4} \gamma \hbar^3 \mathbf{x}_2^2 \mathbf{y}_1^2 \right) \epsilon + \mathcal{O}[\epsilon]^2 \right], \mathbb{E} \left[\frac{\hbar \mathbf{a}_2 \mathbf{t}_1}{\gamma}, -\frac{\hbar \mathbf{x}_2 \mathbf{y}_1}{\mathbf{T}_1}, \right. \right. \\ \left. \left. 1 - \frac{1}{4 (\gamma \mathbf{T}_1^2)} \left(\hbar (4 \mathbf{a}_1 \mathbf{T}_1 (\mathbf{a}_2 \mathbf{T}_1 + \gamma \hbar \mathbf{x}_2 \mathbf{y}_1) + \gamma \hbar \mathbf{x}_2 \mathbf{y}_1 (4 \mathbf{a}_2 \mathbf{T}_1 + 3 \gamma \hbar \mathbf{x}_2 \mathbf{y}_1)) \right) \epsilon + \mathcal{O}[\epsilon]^2 \right] \right\}$$

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

tC

```
In[*]:=  $\mathbf{tC}_i := \mathbb{E}[\theta, \theta, \mathbf{T}_i^{1/2} e^{-\epsilon \mathbf{a}_i \hbar} + \theta_{\$k}]; \\ \overline{\mathbf{tC}}_i := \mathbb{E}[\theta, \theta, \mathbf{T}_i^{-1/2} e^{\epsilon \mathbf{a}_i \hbar} + \theta_{\$k}];$ 
```

In[*]:= $\text{Block}[\{\$k = 3\}, \{\mathbf{tC}_1, \overline{\mathbf{tC}}_2\}]$

$$\text{Out[*]} = \left\{ \mathbb{E}[\theta, \theta, \sqrt{\mathbf{T}_1} - \hbar \mathbf{a}_1 \sqrt{\mathbf{T}_1} \epsilon + \frac{1}{2} \hbar^2 \mathbf{a}_1^2 \sqrt{\mathbf{T}_1} \epsilon^2 - \frac{1}{6} (\hbar^3 \mathbf{a}_1^3 \sqrt{\mathbf{T}_1}) \epsilon^3 + \mathcal{O}[\epsilon]^4], \right. \\ \left. \mathbb{E}[\theta, \theta, \frac{1}{\sqrt{\mathbf{T}_2}} + \frac{\hbar \mathbf{a}_2 \epsilon}{\sqrt{\mathbf{T}_2}} + \frac{\hbar^2 \mathbf{a}_2^2 \epsilon^2}{2 \sqrt{\mathbf{T}_2}} + \frac{\hbar^3 \mathbf{a}_2^3 \epsilon^3}{6 \sqrt{\mathbf{T}_2}} + \mathcal{O}[\epsilon]^4] \right\}$$

tKink

```
In[*]:=  $\mathbf{Kink}[\text{QU}, k] := \mathbf{Kink}[\text{QU}, k] = \text{Block}[\{\$k = k\}, (\mathbf{tR}_{1,3} \overline{\mathbf{tC}}_2) \sim \mathbf{B}_{1,2} \sim \mathbf{tm}_{1,2 \rightarrow 1} \sim \mathbf{B}_{1,3} \sim \mathbf{tm}_{1,3 \rightarrow 1}]; \\ \mathbf{tKink}_i := \mathbf{Kink}[\$U, \$k] /. \{(v : t | T | y | a | x)_1 \rightarrow v_i\}; \\ \overline{\mathbf{Kink}}[\text{QU}, k] := \overline{\mathbf{Kink}}[\text{QU}, k] = \text{Block}[\{\$k = k\}, (\overline{\mathbf{tR}}_{1,3} \mathbf{tC}_2) \sim \mathbf{B}_{1,2} \sim \mathbf{tm}_{1,2 \rightarrow 1} \sim \mathbf{B}_{1,3} \sim \mathbf{tm}_{1,3 \rightarrow 1}]; \\ \overline{\mathbf{tKink}}_i := \overline{\mathbf{Kink}}[\$U, \$k] /. \{(v : t | T | y | a | x)_1 \rightarrow v_i\}$ 
```

Alternative Algorithms

AltLogos

```
In[ ]:= λalt,k[CU] := If[k == 0, 1, Module[{eq, d, b, c, so},
    eq = ρ@eξxcu.ρ@eηycu == ρ@edycu.ρ@ec(t1cu-2εacu).ρ@ebxcu;
    {so} = Solve[Thread[Flatten/@eq], {d, b, c}] /. C@1 → 0;
    Series[e-ηy-ξx+ηξt+c t+d y-2εca+bx /. so, {ε, 0, k}]]];
```

Asides

Aside

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

Aside

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