Pensieve header: Roland's verification of the double in tensorial language.

# The tensorial double at $\epsilon = 0$

In[∘]:=
```
CF[ℰ_] := ExpandDenominator@
   ExpandNumerator@Together[Expand[ℰ] //. e^x_ e^y_ :→ e^(x+y) /. e^x_ :→ e^CF[x]];
Kδ /: Kδ_(i_,j_) := If[i === j, 1, 0];
```

## Zip and Bind

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];
```

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

In[∘]:=
```
Zip_{}[P_] := P; Zip_{ζ_,ζs___}[P_] := (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.

In[∘]:=
```
QZip_(ζs_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_ζs[e^Q1 (P /. zrule)]] ];
QZip_(ζs_List) := QZip_(ζs,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 $b$ and $\alpha$ and the $\zeta$'s are $\beta$ and $a$.

In[ ]:=
```
LZip_{ζs_List,simp_}@𝔼[L_, Q_, P_] :=
  Module[{ζ, z, zs, c, ys, ηs, ηsrule, lt, zrule, L1, L2, Q1, Q2},
   (*Print["LZipping"];*)
   (*ζs//Echo;*)
   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}];
   ηsrule = Table[z* -> ∂_z (L /. Alternatives @@ ζs → 0), {z, zs}]; (*NEW*)
   lt = Inverse@Table[Kδ_{z,ζ*} - ∂_{z,ζ} L, {ζ, ζs}, {z, zs}];
   zrule = Thread[zs → lt.(zs + ys)];
   L1 = c + ηs.zs /. zrule;
   L2 = L1 /. Alternatives @@ zs → 0;
   Q1 = Q /. Join[zrule, ηsrule];
   Q2 = Q1 /. Join[{Alternatives @@ zs → 0}, ηsrule];
   (*{Det[lt] e^{-L2-Q2}, Zip_ζs[e^{L1+Q1} (P/.zrule)]}//Echo;*)
   simp /@ 𝔼[L2, Q2, Det[lt] e^{-L2-Q2} Zip_ζs[e^{L1+Q1} (P /. zrule)]] ];
LZip_{ζs_List} := LZip_{ζs,CF};
```

In[ ]:=
```
Bind_{}[L_, R_] := L R;
Bind_{is__}[L_𝔼, R_𝔼] := Module[{n},
   Flatten@Table[{τ_{n@i}, a_{n@i}}, {i, {is}}];
   Flatten@Table[{ξ_{n@i}, y_{n@i}}, {i, {is}}];
   Times[
     L /. Table[(v : b | a | x | y)_i → v_{n@i}, {i, {is}}],
     R /. Table[(v : β | α | ξ | η)_i → v_{n@i}, {i, {is}}]
    ] // LZip_{Flatten@Table[{β_{n@i},a_{n@i}},{i,{is}}]} // QZip_{Flatten@Table[{ξ_{n@i},y_{n@i}},{i,{is}}]} ];
B_{l_List} := Bind_l; B_{is___} := Bind_{is};
```

## The Alexander story in Gaussian tensor language

```
(*Hopf algebra on the a,x side*)
tam_{i_,j_->k_} := 𝔼[(α_i + α_j) a_k, (e^{-α_j} ξ_i + ξ_j) x_k, 1]
taΔ_{i_->j_,k_} := 𝔼[α_i (a_j + a_k), ξ_i (x_j + x_k), 1]
taS_{i_} := 𝔼[-α_i a_i, -e^{α_i} ξ_i x_i, 1]
taSi_{i_} := 𝔼[-α_i a_i, -e^{α_i} ξ_i x_i, 1]
(*Hopf algebra on the y,b side*)
tbm_{i_,j_->k_} := 𝔼[(β_i + β_j) b_k, (η_i + η_j) y_k, 1]
tbΔ_{i_->j_,k_} := 𝔼[β_i (b_j + b_k), η_i (e^{-b_k} y_j + y_k), 1]
tbS_{i_} := 𝔼[-β_i b_i, -e^{b_i} η_i y_i, 1]
tbSi_{i_} := 𝔼[-β_i b_i, -e^{b_i} η_i y_i, 1]
```

Hopf algebra axioms on both sides separately

(co)-associativity on both sides

In[*]:=
```
{(taΔ1→1,2 ~ B2 ~ taΔ2→2,3) ≡ (taΔ1→1,3 ~ B1 ~ taΔ1→1,2),
 (tbΔ1→1,2 ~ B2 ~ tbΔ2→2,3) ≡ (tbΔ1→1,3 ~ B1 ~ tbΔ1→1,2),
 (tam1,2→1 ~ B1 ~ tam1,3→1) ≡ (tam2,3→2 ~ B2 ~ tam1,2->1),
 (tbm1,2→1 ~ B1 ~ tbm1,3→1) ≡ (tbm2,3→2 ~ B2 ~ tbm1,2->1)}
```
Out[*]=
```
{True, True, True, True}
```

Δ is an algebra morphism

In[*]:=
```
{tam1,2→1 ~ B1 ~ taΔ1→1,2 ≡ (taΔ1→1,3 taΔ2→2,4) ~ B1,2,3,4 ~ (tam3,4→2 tam1,2→1),
 tbm1,2→1 ~ B1 ~ tbΔ1→1,2 ≡ (tbΔ1→1,3 tbΔ2→2,4) ~ B1,2,3,4 ~ (tbm3,4→2 tbm1,2→1)}
```
Out[*]=
```
{True, True}
```

S is convolution inverse of id

In[*]:=
```
{(taΔ1→1,2 ~ B1 ~ taS1) ~ B1,2 ~ tam1,2→1, (taΔ1→1,2 ~ B2 ~ taS2) ~ B1,2 ~ tam1,2→1}
{(tbΔ1→1,2 ~ B1 ~ tbS1) ~ B1,2 ~ tbm1,2→1, (tbΔ1→1,2 ~ B2 ~ tbS2) ~ B1,2 ~ tbm1,2→1}
```
Out[*]=
```
{𝔼[0, 0, 1], 𝔼[0, 0, 1]}
```
Out[*]=
```
{𝔼[0, 0, 1], 𝔼[0, 0, 1]}
```

Si is the inverse of S

In[*]:=
```
{taSi1 ~ B1 ~ taS1 ≡ 𝔼[a1 α1, x1 ξ1, 1], taS1 ~ B1 ~ taSi1 ≡ 𝔼[a1 α1, x1 ξ1, 1]}
{tbSi1 ~ B1 ~ tbS1 ≡ 𝔼[b1 β1, y1 η1, 1], tbS1 ~ B1 ~ tbSi1 ≡ 𝔼[b1 β1, y1 η1, 1]}
```
Out[*]=
```
{True, True}
```
Out[*]=
```
{True, True}
```

S is an algebra anti-morphism

In[*]:=
```
{tam1,2→1 ~ B1 ~ taS1 ≡ (taS1 taS2) ~ B1,2 ~ tam2,1→1, tbm1,2→1 ~ B1 ~ tbS1 ≡ (tbS1 tbS2) ~ B1,2 ~ tbm2,1→1}
{taS1 ~ B1 ~ taΔ1→1,2 ≡ taΔ1→2,1 ~ B1,2 ~ (taS1 taS2), tbS1 ~ B1 ~ tbΔ1→1,2 ≡ tbΔ1→2,1 ~ B1,2 ~ (tbS1 tbS2)}
```
Out[*]=
```
{True, True}
```
Out[*]=
```
{True, True}
```

The pairing.

In[*]:=
```
Table[𝔼[0, 0, y1^r b1^s a2^t x2^u] ~ B1,2 ~ tP1,2 ≡ 𝔼[0, 0, Kδr,u Kδs,t r! s!],
   {r, 0, 4}, {s, 0, 4}, {t, 0, 4}, {u, 0, 4}] // Flatten // Union
```
Out[*]=
```
{True}
```

Pairing

```
tPi_,j_ := 𝔼[βi αj, ηi ξj, 1]
```

Pairing axioms

In[∘]:=
```
(tbm1,2→1 𝔼[α3 a3, ξ3 x3, 1]) ~B1,3~tP1,3 ≡
  (𝔼[β1 b1, η1 y1, 1] 𝔼[β2 b2, η2 y2, 1] taΔ3→4,5) ~B1,4~tP1,4~B2,5~tP2,5
(tbΔ1→1,2 𝔼[α3 a3, ξ3 x3, 1] 𝔼[α4 a4, ξ4 x4, 1]) ~B1,3~tP1,3~B2,4~tP2,4 ≡
  (𝔼[β1 b1, η1 y1, 1] tam3,4→3) ~B1,3~tP1,3
```

Out[∘]= True

Out[∘]= True

In[∘]:=
```
(tbS1 𝔼[α2 a2, ξ2 x2, 1]) ~B1,2~tP1,2 ≡ (𝔼[β1 b1, η1 y1, 1] taS2) ~B1,2~tP1,2
```

Out[∘]= True

## The Double

The double multiplication (should really bind the a's and b's separately)

In[∘]:=
```
Simplify /@ Expand /@
   ((𝔼[βi bi + αj aj, ηi yi + ξj xj, 1] (taΔi→h1,h2~Bh2~taΔh2→h2,h3) (tbΔj→t1,t2~Bt2~tbΔt2→t2,t3)) ~
    Bh3~taSih3~Bt1,h3~(tPt1,h3)~Bt3,h1~(tPt3,h1)~Bh2,j,i,t2~(tamh2,j→k tbmt2,i→-k)) /. {u_-k → uk}
```

Out[∘]= $\mathbb{E}\left[a_k\,(\alpha_i+\alpha_j)+b_k\,(\beta_i+\beta_j),\ y_k\left(\eta_i+e^{-\alpha_i}\eta_j\right)+\left(1-e^{-b_k}\right)\eta_j\,\xi_i+x_k\left(e^{-\alpha_j}\xi_i+\xi_j\right),\ 1\right]$

In[∘]:=
```
(*Deriving tdS using tdm*)
((𝔼[α1 a1 + β2 b2, ξ1 x1 + η2 y2, 1] tbSi1 taS2) ~B2,1~tdm2,1→i) // Simplify
```

Out[∘]= $\mathbb{E}\left[a_i\,(\alpha_1-\alpha_2)+b_i\,(-\beta_1+\beta_2),\ y_i\left(-e^{b_i+\alpha_2}\eta_1+\eta_2\right)+e^{\alpha_2}\left(-1+e^{b_i}\right)\eta_1\,\xi_2+x_i\left(\xi_1-e^{-\alpha_1+\alpha_2}\xi_2\right),\ 1\right]$

In[∘]:=
```
(tbSi1 taS2) ~B1,2~tdm2,1→i
```

Out[∘]= $\mathbb{E}\left[-a_i\,\alpha_2-b_i\,\beta_1,\ -e^{b_i+\alpha_2}y_i\,\eta_1-e^{\alpha_2}x_i\,\xi_2-e^{\alpha_2}\eta_1\,\xi_2+e^{b_i+\alpha_2}\eta_1\,\xi_2,\ 1\right]$

In[∘]:=
```
((tbSi1 taS2) ~B1,2~tdm2,1→i) /. {z_1|2 → zi}
```

Out[∘]= $\mathbb{E}\left[-a_i\,\alpha_i-b_i\,\beta_i,\ -e^{b_i+\alpha_i}y_i\,\eta_i-e^{\alpha_i}x_i\,\xi_i-e^{\alpha_i}\eta_i\,\xi_i+e^{b_i+\alpha_i}\eta_i\,\xi_i,\ 1\right]$

In[∘]:=
```
(*Deriving tdΔ using tdm*)
(tbΔi→3,1 taΔi→2,4) ~B1,2,3,4~(tdm3,4→k tdm1,2→j)
```

Out[∘]= $\mathbb{E}\left[a_j\,\alpha_i+a_k\,\alpha_i+b_j\,\beta_i+b_k\,\beta_i,\ e^{-b_j}\left(e^{b_j}y_j\,\eta_i+y_k\,\eta_i+e^{b_j}x_j\,\xi_i+e^{b_j}x_k\,\xi_i\right),\ 1\right]$

In[∘]:=
```
tbSii taSi
```

Out[∘]= $\mathbb{E}\left[-a_i\,\alpha_i-b_i\,\beta_i,\ -e^{b_i}y_i\,\eta_i-e^{\alpha_i}x_i\,\xi_i,\ 1\right]$

In[∘]:=
```
tdm_{i_,j_→k_} := 𝔼[ak (αi + αj) + bk (βi + βj), yk (ηi + e^(-αi) ηj) + xk (e^(-αj) ξi + ξj) + (1 - e^(-bk)) ηj ξi, 1]
tdΔ_{i_→j_,k_} := 𝔼[αi (aj + ak) + βi (bj + bk), (yj + e^(-bj) yk) ηi + (xj + xk) ξi, 1]
tdS_{i_} := 𝔼[-ai αi - bi βi, -e^(bi+αi) yi ηi - e^(αi) xi ξi - e^(αi) ηi ξi + e^(bi+αi) ηi ξi, 1]
```

multiplication actually agrees with the double formula on the generators:

```
In[•]:= (#⟦3⟧ /. {e → E} // Expand) & /@
         {𝔼[0, 0, x1 b2] ~ B1,2 ~ tdm1,2→1,
          𝔼[0, 0, a1 y2] ~ B1,2 ~ tdm1,2→1,
          𝔼[0, 0, a1 b2] ~ B1,2 ~ tdm1,2→1,
          𝔼[0, 0, x1 y2] ~ B1,2 ~ tdm1,2→1}
```

$$\text{Out[•]=}\ \{b_1 x_1,\ -y_1 + a_1 y_1,\ a_1 b_1,\ 1 - e^{-b_1} + x_1 y_1\}$$

Check S on some generators

```
In[•]:= (#⟦3⟧ /. {e → E} // Expand) & /@ {𝔼[0, 0, y1] ~ B1 ~ tdS1,
          𝔼[0, 0, b1] ~ B1 ~ tdS1,
          𝔼[0, 0, a1] ~ B1 ~ tdS1,
          𝔼[0, 0, x1] ~ B1 ~ tdS1,
          𝔼[0, 0, x1 a1] ~ B1 ~ tdS1,
          𝔼[0, 0, y1 a1] ~ B1 ~ tdS1}
```

$$\text{Out[•]=}\ \{-e^{b_1} y_1,\ -b_1,\ -a_1,\ -x_1,\ -x_1 + a_1 x_1,\ -e^{b_1} y_1 + e^{b_1} a_1 y_1\}$$

Hopf algebra axioms on double

(co)-associativity

```
In[•]:= {(tdΔ1→1,2 ~ B2 ~ tdΔ2→2,3) ≡ (tdΔ1→1,3 ~ B1 ~ tdΔ1→1,2),
         (tdm1,2→1 ~ B1 ~ tdm1,3→1) ≡ (tdm2,3→2 ~ B2 ~ tdm1,2->1)}
Out[•]= {True, True}
```

Δ is an algebra morphism

```
In[•]:= tdm1,2→1 ~ B1 ~ tdΔ1→1,2 ≡ (tdΔ1→1,3 tdΔ2→2,4) ~ B1,2,3,4 ~ (tdm3,4→2 tdm1,2→1)
Out[•]= True
```

S is convolution inverse of id

```
In[•]:= (#⟦2⟧ // Expand // Simplify) & /@
         {(tdΔ1→1,2 ~ B1 ~ tdS1) ~ B1,2 ~ tdm1,2→1, (tdΔ1→1,2 ~ B2 ~ tdS2) ~ B1,2 ~ tdm1,2→1}
Out[•]= {0, 0}
```

S is a (co)-algebra anti-morphism

```
In[•]:= {tdm1,2→1 ~ B1 ~ tdS1 ≡ (tdS1 tdS2) ~ B1,2 ~ tdm2,1→1, tdS1 ~ B1 ~ tdΔ1→1,2 ≡ tdΔ1→2,1 ~ B1,2 ~ (tdS1 tdS2)}
Out[•]= {True, True}
```

$$\text{tR}_{i\_,j\_} := \mathbb{E}[b_i a_j,\ y_i x_j,\ 1]$$
$$\overline{\text{tR}}_{i\_,j\_} := \mathbb{E}\left[-a_j b_i,\ -e^{b_i} x_j y_i,\ 1\right] \quad \text{(*derived from tdS!*)}$$
$$\text{tC}_{i\_} := \mathbb{E}\left[0,\ 0,\ e^{\frac{-1}{2} b_i}\right]$$
$$\overline{\text{tC}}_{i\_} := \mathbb{E}\left[0,\ 0,\ e^{\frac{1}{2} b_i}\right]$$
$$\text{tKink}_{i\_} := \mathbb{E}\left[a_i b_i,\ x_i y_i,\ e^{\frac{b_i}{2}}\right]$$
$$\overline{\text{tKink}}_{i\_} := \mathbb{E}\left[-a_i b_i,\ -e^{b_i} x_i y_i,\ e^{\frac{-b_i}{2}}\right]$$

(*Deriving $\overline{R}$ formula*)

$tR_{i,j} \sim B_j \sim tdS_j$

Out[∘]= $\mathbb{E}\left[-a_j b_i, -e^{b_i} x_j y_i, 1\right]$

Kinks

In[∘]:= $k1 = \left(tR_{1,3}\, \overline{tC}_2\right) \sim B_{1,2} \sim tdm_{1,2\to 1} \sim B_{1,3} \sim tdm_{1,3\to 1}$

$k2 = \left(tR_{3,1}\, tC_2\right) \sim B_{1,2} \sim tdm_{1,2\to 1} \sim B_{1,3} \sim tdm_{1,3\to 1}$

$k3 = \left(\overline{tR}_{1,3}\, tC_2\right) \sim B_{1,2} \sim tdm_{1,2\to 1} \sim B_{1,3} \sim tdm_{1,3\to 1}$

$k4 = \left(\overline{tR}_{3,1}\, \overline{tC}_2\right) \sim B_{1,2} \sim tdm_{1,2\to 1} \sim B_{1,3} \sim tdm_{1,3\to 1}$

Out[∘]= $\mathbb{E}\left[a_1 b_1, x_1 y_1, e^{\frac{b_1}{2}}\right]$

Out[∘]= $\mathbb{E}\left[a_1 b_1, x_1 y_1, e^{\frac{b_1}{2}}\right]$

Out[∘]= $\mathbb{E}\left[-a_1 b_1, -e^{b_1} x_1 y_1, e^{-\frac{b_1}{2}}\right]$

Out[∘]= $\mathbb{E}\left[-a_1 b_1, -e^{b_1} x_1 y_1, e^{-\frac{b_1}{2}}\right]$

In[∘]:= $\left(tKink_1\, \overline{tKink_2}\right) \sim B_{1,2} \sim tdm_{1,2\to 1}$

Out[∘]= $\mathbb{E}[0, 0, 1]$

Reidemeister 2:

In[∘]:= $\left(tR_{1,2}\, \overline{tR}_{3,4}\right) \sim B_{1,3} \sim tdm_{1,3\to 1} \sim B_{2,4} \sim tdm_{2,4\to 2}$

Out[∘]= $\mathbb{E}[0, 0, 1]$

cyclic Reidemeister 2:

In[∘]:= $\left(tR_{1,4}\, \overline{tR}_{5,2}\, \overline{tC}_3\right) \sim B_{2,4} \sim tdm_{2,4\to 2} \sim B_{1,3} \sim tdm_{1,3\to 1} \sim B_{1,5} \sim tdm_{1,5\to 1} \equiv \overline{tC}_1$

Out[∘]= True

Reidemeister 3:

In[∘]:= $\left(tR_{1,2}\, tR_{4,3}\, tR_{5,6}\right) \sim B_{1,4} \sim tdm_{1,4\to 1} \sim B_{2,5} \sim tdm_{2,5\to 2} \sim B_{3,6} \sim tdm_{3,6\to 3} \equiv$
$\left(tR_{1,6}\, tR_{2,3}\, tR_{4,5}\right) \sim B_{1,4} \sim tdm_{1,4\to 1} \sim B_{2,5} \sim tdm_{2,5\to 2} \sim B_{3,6} \sim tdm_{3,6\to 3}$

Out[∘]= True

Trefoil

In[∘]:=
```
Z = tR_{1,5} tR_{6,2} tR_{3,7} tC_4 tKink_8 tKink_9 tKink_10;
Do[Z = Z ~ B_{1,r} ~ tdm_{1,r→1}, {r, 2, 10}]; Z
1 / (Z[[3]] /. e^{c_. b_1} → T^c) // Expand
```

Out[∘]= $\mathbb{E}\left[0, 0, \frac{e^{b_1}}{1 - e^{b_1} + e^{2 b_1}}\right]$

Out[∘]= $-1 + \frac{1}{T} + T$

Deriving the Drinfeld element u and its inverse ui

In[∘]:=
```
u_{i_} := tR_{1,2} ~ B_1 ~ tdS_1 ~ B_{1,2} ~ tdm_{2,1→i}
ui_{i_} := tR_{1,2} ~ B_2 ~ tdS_2 ~ B_2 ~ tdS_2 ~ B_{1,2} ~ tdm_{2,1→i}
```

In[∘]:= `{u_1, ui_1}`

Out[∘]= $\left\{\mathbb{E}\left[-a_1 b_1, -e^{b_1} x_1 y_1, e^{-b_1}\right], \mathbb{E}\left[a_1 b_1, x_1 y_1, e^{b_1}\right]\right\}$

u and ui are inverses

In[∘]:= `(u_1 ui_2) ~ B_{1,2} ~ tdm_{1,2→1}`

Out[∘]= $\mathbb{E}[0, 0, 1]$

The ribbon element v satisfies $v^2 = S(u)\, u$. The spinner $C=uv^{-1}$.
It is convenient to compute $z = S(u)\, u^{-1}$ which is something easy.

In[∘]:= `((u_1 ~ B_1 ~ tdS_1) ui_2) ~ B_{1,2} ~ tdm_{1,2→1}`

Out[∘]= $\mathbb{E}\left[0, 0, e^{b_1}\right]$

So in our case S(u) = u z so S(u)u = $u^2\, z$ and v = $uz^{\frac{1}{2}}$ and finally C = $z^{\frac{-1}{2}}$.