Pensieve header: Computations in the ybax algebra using the Drinfel'd double (at $\epsilon^2=0$. Continues ExpDoubleEpsilonSquare5@@.nb.

The double at $\epsilon^2 = 0$

Utilities

Canonical Form:

ybax

In[ ]:=
```
CF[sd_SeriesData] := MapAt[CF, sd, 3];
CF[ℰ_] := ExpandDenominator@
   ExpandNumerator@Together[Expand[ℰ] //. e^x_ e^y_ :> e^(x+y) /. e^x_ :> e^CF[x]];
```

The Kronecker δ:

ybax

In[ ]:=
```
Kδ /: Kδ_{i_,j_} := If[i === j, 1, 0];
```

Equality and multiplication of perturbed Gaussians; $\mathbb{E}[L, Q, P]$ stands for $e^{L+Q} P$:

ybax

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 and Bind

ybax

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

ybax

In[ ]:=
```
expand[sd_SeriesData] := MapAt[expand, sd, 3];
expand[ℰ_] := Expand[ℰ];
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 $\mathbb{E}(L, Q, P) = \mathrm{P}e^{L+Q}$. Such zips regard the L variables as scalars.

ybax

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};
```

Upper to lower and lower to Upper:

ybax

In[∘]:=
```
U2l = {B_{i_}^{p_.} → e^{-p b_i}, B^{p_.} → e^{-p b}, T_{i_}^{p_.} → e^{p t_i}, T^{p_.} → e^{p t}, 𝒜_{i_}^{p_.} → e^{p α_i}, 𝒜^{p_.} → e^{p α}};
l2U = {e^{c_. b_{i_}+d_.} :→ B_i^{-c} e^d, e^{c_. b+d_.} :→ B^{-c} e^d,
  e^{c_. t_{i_}+d_.} :→ T_i^c e^d, e^{c_. t+d_.} :→ T^c e^d,
  e^{c_. α_{i_}+d_.} :→ 𝒜_i^c e^d, e^{c_. α+d_.} :→ 𝒜^c e^d,
  e^{ℰ_} :→ e^{Expand@ℰ}};
```

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 α and the ζ’s are β and a.

ybax

In[∘]:=
```
LZip_{ζs_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 /. U2l /. zrule) /. Alternatives @@ zs → 0;
  simp /@ 𝔼[L2, Q2, Det[lt] e^{-L2-Q2} Zip_{ζs}[e^{L1+Q1} (P /. U2l /. zrule)]] //. l2U ];
LZip_{ζs_List} := LZip_{ζs,CF};
```

ybax

In[∘]:=
```
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[{ξ_{n@i},y_{n@i}},{i,{is}}]} ];
B_{L_List}[L_, R_] := Bind_L[L, R]; B_{is___}[L_, R_] := Bind_{{is}}[L, R];
```

The two halves

ybax

```
In[]:= am_{i_,j_->k_} := 𝔼[(α_i + α_j) a_k, (e^{-α_j} ξ_i + ξ_j) x_k, 1 + O[ϵ]^2]
aΔ_{i_->j_,k_} := 𝔼[α_i (a_j + a_k), ξ_i (x_j + x_k), 1 + ϵ ξ_i x_k (-a_j + ξ_i x_j/2) + O[ϵ]^2]
aS_{i_} := 𝔼[-α_i a_i, - e^{α_i} ξ_i x_i, 1 - ϵ e^{α_i} ξ_i x_i (a_i + e^{α_i} ξ_i x_i/2) + O[ϵ]^2]
aSi_{i_} := 𝔼[-α_i a_i, -e^{α_i} ξ_i x_i, 1 - ϵ e^{α_i} ξ_i x_i (a_i - 1 + e^{α_i} ξ_i x_i/2) + O[ϵ]^2]
```

ybax

```
In[]:= bm_{i_,j_→k_} := 𝔼[(β_i + β_j) b_k, (η_i + η_j) y_k, 1 - ϵ η_j y_k β_i + O[ϵ]^2]
bΔ_{i_->j_,k_} := 𝔼[β_i (b_j + b_k), η_i (e^{-b_k} y_j + y_k), 1 + ϵ η_i^2 y_j y_k e^{-b_k}/2 + O[ϵ]^2]
bS_{i_} := 𝔼[-β_i b_i, -e^{b_i} η_i y_i, 1 - ϵ e^{b_i} η_i y_i (β_i + e^{b_i} η_i y_i/2) + O[ϵ]^2]
bSi_{i_} := 𝔼[-β_i b_i, -e^{b_i} η_i y_i, 1 - ϵ e^{b_i} η_i y_i (β_i - 1 + e^{b_i} η_i y_i/2) + O[ϵ]^2]
```

First check that on the generators this agrees with our conventions in the handout:

```
In[]:= {{"[a,x]" → ((𝔼[0, 0, a_2 x_1] ~B_{1,2}~ am_{1,2→1})〚3〛 - (𝔼[0, 0, a_1 x_2] ~B_{1,2}~ am_{1,2→1})〚3〛),
    "[b,y]" → ((𝔼[0, 0, y_2 b_1] ~B_{1,2}~ bm_{1,2→1})〚3〛 - (𝔼[0, 0, y_1 b_2] ~B_{1,2}~ bm_{1,2→1})〚3〛)} /. z_1 → z,
 {"Δ[y]" → Last[𝔼[0, 0, y_1] ~B_1~ bΔ_{1→1,2}],
  "Δ[b]" → Last[𝔼[0, 0, b_1] ~B_1~ bΔ_{1→1,2}],
  "Δ[a]" → Last[𝔼[0, 0, a_1] ~B_1~ aΔ_{1→1,2}],
  "Δ[x]" → Last[𝔼[0, 0, x_1] ~B_1~ aΔ_{1→1,2}]},
 {
   "S(a)" → ((𝔼[0, 0, a_1] ~B_1~ aS_1)〚3〛),
   "S(x)" → ((𝔼[0, 0, x_1] ~B_1~ aS_1)〚3〛),
   "S(b)" → ((𝔼[0, 0, b_1] ~B_1~ bS_1)〚3〛),
   "S(y)" → ((𝔼[0, 0, y_1] ~B_1~ bS_1)〚3〛)
 } /. z_1 → z}
```

Hopf algebra axioms on both sides separately

(co)-associativity on both sides

```
In[]:= {(aΔ_{1→1,2} ~B_2~ aΔ_{2→2,3}) ≡ (aΔ_{1→1,3} ~B_1~ aΔ_{1→1,2}), (bΔ_{1→1,2} ~B_2~ bΔ_{2→2,3}) ≡ (bΔ_{1→1,3} ~B_1~ bΔ_{1→1,2}),
 (am_{1,2→1} ~B_1~ am_{1,3→1}) ≡ (am_{2,3→2} ~B_2~ am_{1,2→1}), (bm_{1,2→1} ~B_1~ bm_{1,3→1}) ≡ (bm_{2,3→2} ~B_2~ bm_{1,2→1})}
```

Δ is an algebra morphism

```
In[]:= {am_{1,2→1} ~B_1~ aΔ_{1→1,2} ≡ (aΔ_{1→1,3} aΔ_{2→2,4}) ~B_{1,2,3,4}~ (am_{3,4→2} am_{1,2→1}),
 bm_{1,2→1} ~B_1~ bΔ_{1→1,2} ≡ (bΔ_{1→1,3} bΔ_{2→2,4}) ~B_{1,2,3,4}~ (bm_{3,4→2} bm_{1,2→1})}
```

S is convolution inverse of id

```
In[]:= {(aΔ_{1→1,2} ~B_1~ aS_1) ~B_{1,2}~ am_{1,2→1}, (aΔ_{1→1,2} ~B_2~ aS_2) ~B_{1,2}~ am_{1,2→1}}
{(bΔ_{1→1,2} ~B_1~ bS_1) ~B_{1,2}~ bm_{1,2→1}, (bΔ_{1→1,2} ~B_2~ bS_2) ~B_{1,2}~ bm_{1,2→1}}
```

Si is the inverse of S

```
In[]:= {aSi_1 ~B_1~ aS_1 ≡ 𝔼[a_1 α_1, x_1 ξ_1, 1], aS_1 ~B_1~ aSi_1 ≡ 𝔼[a_1 α_1, x_1 ξ_1, 1]}
{bSi_1 ~B_1~ bS_1 ≡ 𝔼[b_1 β_1, y_1 η_1, 1], bS_1 ~B_1~ bSi_1 ≡ 𝔼[b_1 β_1, y_1 η_1, 1]}
```

S is an algebra anti-(co)morphism

In[∘]:=
```
{am1,2→1 ~ B1 ~ aS1 ≡ (aS1 aS2) ~ B1,2 ~ am2,1→1, bm1,2→1 ~ B1 ~ bS1 ≡ (bS1 bS2) ~ B1,2 ~ bm2,1→1}
{aS1 ~ B1 ~ aΔ1→1,2 ≡ aΔ1→2,1 ~ B1,2 ~ (aS1 aS2), bS1 ~ B1 ~ bΔ1→1,2 ≡ bΔ1→2,1 ~ B1,2 ~ (bS1 bS2)}
```

Pairing

ybax

In[∘]:=
```
tP_{i_,j_} := 𝔼[β_i α_j, η_i ξ_j, 1 + ϵ η_i^2 ξ_j^2 / 4]
```

In[∘]:=
```
qfac[k_, q_] := (1 - q)^-k QPochhammer[q, q, k] // FunctionExpand
qfe[k_] := Normal[Series[qfac[k, E^ρ], {ρ, 0, 1}]] /. {ρ → ϵ}
Table[𝔼[0, 0, y1^r b1^s a2^t x2^u] ~ B1,2 ~ tP1,2 ≡ 𝔼[0, 0, KΔr,u KΔs,t qfe[r] s!],
   {r, 0, 4}, {s, 0, 4}, {t, 0, 4}, {u, 0, 4}] // Flatten // Union
```

Pairing axioms

In[∘]:=
```
{(bm1,2→1 𝔼[α3 a3, ξ3 x3, 1]) ~ B1,3 ~ tP1,3 ≡
   (𝔼[β1 b1, η1 y1, 1] 𝔼[β2 b2, η2 y2, 1] aΔ3→4,5) ~ B1,4 ~ tP1,4 ~ B2,5 ~ tP2,5
 , (bΔ1→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] am3,4→3) ~ B1,3 ~ tP1,3 }
```

In[∘]:=
```
{(bS1 𝔼[α2 a2, ξ2 x2, 1]) ~ B1,2 ~ tP1,2 ≡ (𝔼[β1 b1, η1 y1, 1] aS2) ~ B1,2 ~ tP1,2,
 (bSi1 𝔼[α2 a2, ξ2 x2, 1]) ~ B1,2 ~ tP1,2 ≡ (𝔼[β1 b1, η1 y1, 1] aSi2) ~ B1,2 ~ tP1,2}
```

The Double

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

ybax

In[∘]:=
```
Block[{i, j, k}, dm_{i_,j_→k_} =
   (𝔼[β_i b_i + α_j a_j, η_i y_i + ξ_j x_j, 1] (aΔ_{i→h1,h2} ~ B_h2 ~ aΔ_{h2→h2,h3}) (bΔ_{j→t1,t2} ~ B_t2 ~ bΔ_{t2→t2,t3})) ~
    B_h3 ~ aSi_h3 ~ B_{t1,h3} ~ (tP_{t1,h3}) ~ B_{t3,h1} ~ (tP_{t3,h1}) ~ B_{h2,j,i,t2} ~ (am_{h2,j→k} bm_{i,t2→k})]
```

ybax

In[∘]:=
```
Block[{i}, dS_{i_} = 𝔼[β_i b1 + α_i a2, η_i y1 + ξ_i x2, 1] ~ B1,2 ~ (bSi1 aS2) ~ B1,2 ~ dm_{2,1→i}]
```

ybax

In[∘]:=
```
Block[{i, j, k}, dΔ_{i_→j_,k_} = (bΔ_{i→3,1} aΔ_{i→2,4}) ~ B1,2,3,4 ~ (dm_{3,4→k} dm_{1,2→j})]
```

First check the double formulas on the generators agree with SL2Portfolio.pdf:

```
In[∘]:= {
      "[a,y]" → ((𝔼[0, 0, y2 a1] ~ B1,2 ~ dm1,2→1)〚3〛 - (𝔼[0, 0, y1 a2] ~ B1,2 ~ dm1,2→1)〚3〛),
      "[b,x]"  → ((𝔼[0, 0, x2 b1] ~ B1,2 ~ dm1,2→1)〚3〛 - (𝔼[0, 0, x1 b2] ~ B1,2 ~ dm1,2→1)〚3〛),
      "xy-qyx" → ((𝔼[0, 0, x1 y2] ~ B1,2 ~ dm1,2→1)〚3〛 - (1 + ϵ) (𝔼[0, 0, y1 x2] ~ B1,2 ~ dm1,2→1)〚3〛)
    } /. {z_1 → z} // Expand // Factor
   {
    "Δ(a)" → ((𝔼[0, 0, a1] ~ B1 ~ dΔ1→1,2)〚3〛),
    "Δ(x)" → ((𝔼[0, 0, x1] ~ B1 ~ dΔ1→1,2)〚3〛),
    "Δ(b)" → ((𝔼[0, 0, b1] ~ B1 ~ dΔ1→1,2)〚3〛),
    "Δ(y)" → ((𝔼[0, 0, y1] ~ B1 ~ dΔ1→1,2)〚3〛)
   } // Simplify
   {
     "S(a)" → ((𝔼[0, 0, a1] ~ B1 ~ dS1)〚3〛),
     "S(x)" → ((𝔼[0, 0, x1] ~ B1 ~ dS1)〚3〛),
     "S(b)" → ((𝔼[0, 0, b1] ~ B1 ~ dS1)〚3〛),
     "S(y)" → ((𝔼[0, 0, y1] ~ B1 ~ dS1)〚3〛)
   } /. {z_1 → z} // Simplify
```

Hopf algebra axioms on double

(co)-associativity

```
In[∘]:= {(dΔ1→1,2 ~ B2 ~ dΔ2→2,3) ≡ (dΔ1→1,3 ~ B1 ~ dΔ1→1,2),
   (dm1,2→1 ~ B1 ~ dm1,3→1) ≡ (dm2,3→2 ~ B2 ~ dm1,2->1)}
```

Δ is an algebra morphism

```
In[∘]:= dm1,2→1 ~ B1 ~ dΔ1→1,2 ≡ (dΔ1→1,3 dΔ2→2,4) ~ B1,2,3,4 ~ (dm3,4→2 dm1,2→1)
```

S is convolution inverse of id

```
In[∘]:= {(dΔ1→1,2 ~ B1 ~ dS1) ~ B1,2 ~ dm1,2→1, (dΔ1→1,2 ~ B2 ~ dS2) ~ B1,2 ~ dm1,2→1}
```

S is a (co)-algebra anti-morphism

```
In[∘]:= {dm1,2→1 ~ B1 ~ dS1 ≡ (dS1 dS2) ~ B1,2 ~ dm2,1→1, dS1 ~ B1 ~ dΔ1→1,2 ≡ dΔ1→2,1 ~ B1,2 ~ (dS1 dS2)} // Expand
```

R-matrix

$$\text{In[∘]:= } \mathbb{e}_{q\_,k\_}[x\_] := \mathbb{e}\,\hat{}\left(\sum_{j=1}^{k+1} \frac{(1-q)^j x^j}{j\,(1-q^j)}\right)$$

ybax

$$\text{In[∘]:= } R_{i\_,j\_} := \mathbb{E}\left[b_i a_j,\ y_i x_j,\ 1 - \epsilon\, y_i^2 x_j^2 / 4 + O[\epsilon]^2\right]$$

```
In[∘]:= Series[𝕖q,1[z] /. {z → yi xj, q → 1 + ρ}, {ρ, 0, 1}] /. {ρ → ϵ}
```

Quasi-triangular axiom 1:

```
In[∘]:= R1,2 ~ B1 ~ dΔ1->1,3 ≡ (R1,4 R3,2) ~ B2,4 ~ dm2,4→2
```

Quasi-triangular axiom 2:

In[∘]:=
```
((dΔ_{1->1,2} R_{3,4}) ~ B_{1,2,3,4} ~ (dm_{1,3→1} dm_{2,4→2})) ≡
 ((dΔ_{1→2,1} R_{3,4}) ~ B_{1,2,3,4} ~ (dm_{3,1→1} dm_{4,2→2}))
```

Reidemeister 3:

In[∘]:=
```
((R_{1,2} R_{4,3} R_{5,6}) ~ B_{1,4} ~ dm_{1,4→1} ~ B_{2,5} ~ dm_{2,5→2} ~ B_{3,6} ~ dm_{3,6→3}) ≡
 ((R_{1,6} R_{2,3} R_{4,5}) ~ B_{1,4} ~ dm_{1,4→1} ~ B_{2,5} ~ dm_{2,5→2} ~ B_{3,6} ~ dm_{3,6→3})
```

ybax

In[∘]:=
```
Block[{i, j}, R̄_{i_,j_} = Expand /@ R_{i,j} ~ B_j ~ dS_j]
```

Reidemeister 2

In[∘]:=
```
{(R̄_{1,2} R_{3,4}) ~ B_{1,2,3,4} ~ (dm_{1,3→1} dm_{2,4→2}), (R_{1,2} R̄_{3,4}) ~ B_{1,2,3,4} ~ (dm_{1,3→1} dm_{2,4→2})}
```

Deriving the Drinfeld element u and its inverse ui

ybax

In[∘]:=
```
Block[{i}, {
  u_{i_} = R_{1,2} ~ B_1 ~ dS_1 ~ B_{1,2} ~ dm_{2,1→i},
  ui_{i_} := R_{1,2} ~ B_2 ~ dS_2 ~ B_2 ~ dS_2 ~ B_{1,2} ~ dm_{2,1→i}
 }]
```

u and ui are inverses

In[∘]:=
```
(u_1 ui_2) ~ B_{1,2} ~ dm_{1,2→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 ~ dS_1) ui_2) ~ B_{1,2} ~ dm_{1,2→1}
```

(* Needs fixing! *) So in our case S(u) = u z so S(u)u = $u^2\,z$ and v = u$z^{\frac{1}{2}}$ and finally C = uv^{-1} = $z^{-\frac{1}{2}} = e^{t_1/2}(1 - \epsilon\, a_1)$.

ybax

In[∘]:=
```
Block[{i},
 {CC_{i_} = 𝔼[0, 0, B_i^{1/2} e^{-ϵ a_i/2} + O[ϵ]^2],
  C̄C̄_{i_} = 𝔼[0, 0, B_i^{-1/2} e^{ϵ a_i/2} + O[ϵ]^2]
 }]
```

ybax

In[∘]:=
```
Block[{i, j}, {
  Kink_{i_} = (R_{1,3} C̄C̄_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→i},
  K̄īn̄k̄_{j_} = (R̄_{1,3} CC_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→j}
 }]
```

In[∘]:=
```
k2 = (R_{3,1} CC_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→i} /. e → E;
k4 = (R̄_{3,1} C̄C̄_2) ~ B_{1,2} ~ dm_{1,2→1} ~ B_{1,3} ~ dm_{1,3→j} /. e → E;
Simplify@{Kink_i ≡ k2, K̄īn̄k̄_j ≡ k4, (Kink_i K̄īn̄k̄_j) ~ B_{i,j} ~ dm_{i,j→1}}
```

Reidemeister 2:

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

Cyclic Reidemeister 2:

In[∘]:= $\left(R_{1,4}\,\overline{R}_{5,2}\,\overline{CC}_3\right) \sim B_{2,4} \sim dm_{2,4\to 2} \sim B_{1,3} \sim dm_{1,3\to 1} \sim B_{1,5} \sim dm_{1,5\to 1} \equiv \overline{CC}_1$

Trefoil

ybax

In[∘]:=
```
Z = R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10;
Do[Z = Z~B1,r~dm1,r→1, {r, 2, 10}];
Simplify /@ Z
```

```
Timing[
 Z = R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10;
 Do[Z = Z~B1,r~dm1,r→1, {r, 2, 10}];
 Simplify /@ Z ]
```

---

ybax

In[∘]:=
```
b2ti_ := 𝔼[αi ai − βi ti, ξi xi + ηi yi, 1 + ϵ βi ai + O[ϵ]^2]
t2bi_ := 𝔼[αi ai − τi bi, ξi xi + ηi yi, 1 + ϵ τi ai + O[ϵ]^2]
```

In[∘]:= `R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10`

In[∘]:= `(R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10)~BRange[10]~Product[b2ti, {i, 10}]`

In[∘]:=
```
Z = (((R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10)~BRange[10]~Product[b2ti, {i, 10}]) /. T_ → T1)~
  BRange[10]~Product[t2bi, {i, 10}]
```

```
Timing[
 Do[Z = Z~B1,r~dm1,r→1, {r, 2, 10}];
 Simplify@Z[[3]] ]
```

In[∘]:=
```
Timing[
 Z = R1,5 R6,2 R3,7 CC4 Kink8 Kink9 Kink10 /. B_ → B1;
 Do[Print["doing ", r]; Z = Z~B1,r~dm1,r→1 /. B_ → B1, {r, 2, 10}];
 Simplify@Z[[3]] ]
```