Pensieve header: The Objects.

```
Echo@"Warning: On Sep 4 2019 I swapped the operations
   ϵ and η. Some incompatibilities may arise in older notebooks."
```

Program

The Objects

Program

Symmetric Algebra Objects

Program

$$sm_{i\_,j\_\to k\_} := \mathbb{E}_{\{i,j\}\to\{k\}}\left[b_k\,(\beta_i+\beta_j) + t_k\,(\tau_i+\tau_j) + a_k\,(\alpha_i+\alpha_j) + y_k\,(\eta_i+\eta_j) + x_k\,(\xi_i+\xi_j)\right];$$
$$s\Delta_{i\_\to j\_,k\_} := \mathbb{E}_{\{i\}\to\{j,k\}}\left[\beta_i\,(b_j+b_k) + \tau_i\,(t_j+t_k) + \alpha_i\,(a_j+a_k) + \eta_i\,(y_j+y_k) + \xi_i\,(x_j+x_k)\right];$$
$$sS_{i\_} := \mathbb{E}_{\{i\}\to\{i\}}[-\beta_i\,b_i - \tau_i\,t_i - \alpha_i\,a_i - \eta_i\,y_i - \xi_i\,x_i];$$
$$s\eta_{i\_} := \mathbb{E}_{\{\}\to\{i\}}[0];$$
$$s\epsilon_{i\_} := \mathbb{E}_{\{i\}\to\{\}}[0];$$

Program

$$s\sigma_{i\_\to j\_} := \mathbb{E}_{\{i\}\to\{j\}}\left[\beta_i\,b_j + \tau_i\,t_j + \alpha_i\,a_j + \eta_i\,y_j + \xi_i\,x_j\right];$$
$$s\Upsilon_{i\_\to j\_,k\_,l\_,m\_} := \mathbb{E}_{\{i\}\to\{j,k,l,m\}}\left[\beta_i\,b_k + \tau_i\,t_k + \alpha_i\,a_l + \eta_i\,y_j + \xi_i\,x_m\right];$$

Program

The CU Definitions

Program

$$c\Lambda = \left(\eta_i + \frac{e^{-\gamma\,\alpha_i-\epsilon\,\beta_i}\,\eta_j}{1+\gamma\,\epsilon\,\eta_j\,\xi_i}\right) y_k + \left(\beta_i + \beta_j + \frac{\text{Log}[1+\gamma\,\epsilon\,\eta_j\,\xi_i]}{\epsilon}\right) b_k +$$
$$\left(\alpha_i + \alpha_j + \frac{\text{Log}[1+\gamma\,\epsilon\,\eta_j\,\xi_i]}{\gamma}\right) a_k + \left(\frac{e^{-\gamma\,\alpha_j-\epsilon\,\beta_j}\,\xi_i}{1+\gamma\,\epsilon\,\eta_j\,\xi_i} + \xi_j\right) x_k;$$
$$\text{Define}\left[cm_{i,j\to k} = \mathbb{E}_{\{i,j\}\to\{k\}}[c\Lambda]\right]$$

Program

$$\text{Define}\left[c\sigma_{i\to j} = s\sigma_{i,j}\ /.\ \tau_i \to 0,\ c\epsilon_i = s\epsilon_i,\ c\eta_i = s\eta_i,\ c\Delta_{i\to j,k} = s\Delta_{i\to j,k},\right.$$
$$\left. cS_i = sS_i\ //\ s\Upsilon_{i\to 1,2,3,4}\ //\ cm_{4,3\to i}\ //\ cm_{i,2\to i}\ //\ cm_{i,1\to i}\right];$$

Program

Booting Up QU

Program

$$\text{Define}\left[a\sigma_{i\to j} = \mathbb{E}_{\{i\}\to\{j\}}[a_j\,\alpha_i + x_j\,\xi_i],\ b\sigma_{i\to j} = \mathbb{E}_{\{i\}\to\{j\}}[b_j\,\beta_i + y_j\,\eta_i]\right]$$

Program

```
Define[am_{i,j→k} = 𝔼_{{i,j}→{k}}[(α_i + α_j) a_k + (𝒜_j^{-1} ξ_i + ξ_j) x_k],
  bm_{i,j→k} = 𝔼_{{i,j}→{k}}[(β_i + β_j) b_k + (η_i + ⅇ^{-ϵ β_i} η_j) y_k]]
```

Three types of inverses appear below!

$\overline{R}$ is the inverse of R in the algebra $\mathbb{B} \otimes \mathbb{A}$.

P is the inverse of R as a quadratic form, like how an element of $V^* \otimes V^*$ can be the inverse of an element of $V \otimes V$.

$\overline{\mathrm{aS}}$ is the inverse of aS as an operator, like how an element of $V^* \otimes V$ can be the inverse of another element of $V^* \otimes V$.

Program

```
Define[R_{i,j} = 𝔼_{{}→{i,j}}[ℏ a_j b_i + Sum_{k=1}^{$k+1} ((1 - ⅇ^{γ ϵ ℏ})^k (ℏ y_i x_j)^k) / (k (1 - ⅇ^{k γ ϵ ℏ}))],
  R̄_{i,j} = CF@𝔼_{{}→{i,j}}[-ℏ a_j b_i, -ℏ x_j y_i / B_i, 1 + If[$k == 0, 0, (R̄_{{i,j},$k-1})_{$k}[3] -
      (((R̄_{{i,j},0})_{$k} R_{1,2} (R̄_{{3,4},$k-1})_{$k}) // (bm_{i,1→i} am_{j,2→j}) // (bm_{i,3→i} am_{j,4→j}))[3]]],
  P_{i,j} = 𝔼_{{i,j}→{}}[β_i α_j / ℏ, η_i ξ_j / ℏ, 1 + If[$k == 0, 0, (P_{{i,j},$k-1})_{$k}[3] -
      (R_{1,2} // ((P_{{1,j},0})_{$k} (P_{{i,2},$k-1})_{$k}))[3]]]]
```

Program

```
Define[aS_i = (aσ_{i→2} R̄_{1,i}) // P_{1,2},
  aS̄_i = 𝔼_{{i}→{i}}[-a_i α_i, -x_i 𝒜_i ξ_i, 1 + If[$k == 0, 0, (aS̄_{{i},$k-1})_{$k}[3] -
      ((aS̄_{{i},0})_{$k} // aS_i // (aS̄_{{i},$k-1})_{$k})[3]]]]
```

(was $\mathrm{aS}_j = \overline{R}_{i,j} \sim B_i \sim P_{i,j}$).

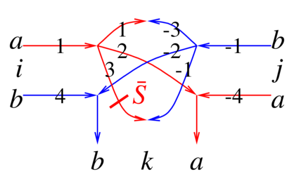
Program

```
Define[bS_i = bσ_{i→1} R_{i,2} // aS_2 // P_{1,2},
  bS̄_i = bσ_{i→1} R_{i,2} // aS̄_2 // P_{1,2},
  aΔ_{i→j,k} = (R_{1,j} R_{2,k}) // bm_{1,2→3} // P_{3,i},
  bΔ_{i→j,k} = (R_{j,1} R_{k,2}) // am_{1,2→3} // P_{i,3}]
```

(was $\mathrm{bS}_i = R_{i,1} \sim B_1 \sim \mathrm{aS}_1 \sim B_1 \sim P_{i,1}$, $\overline{\mathrm{bS}}_i = R_{i,1} \sim B_1 \sim \overline{\mathrm{aS}}_1 \sim B_1 \sim P_{i,1}$).

Program

The Drinfel'd double:



Program

```
Define[
  dm_{i,j→k} = ((sΥ_{i→4,4,1,1} // aΔ_{1→1,2} // aΔ_{2→2,3} // aS̄_3) (sΥ_{j→-1,-1,-4,-4} // bΔ_{-1→-1,-2} // bΔ_{-2→-2,-3})) //
    (P_{-1,3} P_{-3,1} am_{2,-4→k} bm_{4,-2→k})]
```

Program

```
Define[dσi→j = aσi→j bσi→j,
 dϵi = sϵi, dηi = sηi,
 dSi = sϒi→1,1,2,2 // (bS̄1 aS2) // dm2,1→i,
 dS̄i = sϒi→1,1,2,2 // (bS1 aS̄2) // dm2,1→i,
 dΔi→j,k = (bΔi→3,1 aΔi→2,4) // (dm3,4→k dm1,2→j)]
```

Program

In[∘]:=

```
Define[Ci = 𝔼{}→{i}[0, 0, Bi^(1/2) ⅇ^(-ℏ ϵ ai/2)]$k,
 C̄i = 𝔼{}→{i}[0, 0, Bi^(-1/2) ⅇ^(ℏ ϵ ai/2)]$k,
 Kinki = (R1,3 C̄2) // dm1,2→1 // dm1,3→i,
 K̄inki = (R̄1,3 C2) // dm1,2→1 // dm1,3→i]
```

Program

Note. $t == \epsilon a - \gamma b$ and $b == -t/\gamma + \epsilon a/\gamma$.

Program

```
Define[b2ti = 𝔼{i}→{i}[αi ai + βi (ϵ ai - ti) / γ + ξi xi + ηi yi],
 t2bi = 𝔼{i}→{i}[αi ai + τi (ϵ ai - γ bi) + ξi xi + ηi yi]]
```

Program

The Knot Tensors

Program

```
Define[kRi,j = (Ri,j // (b2ti b2tj)) /. ti|j → t,
 k̄Ri,j = (R̄i,j // (b2ti b2tj)) /. {ti|j → t, Ti|j → T},
 kmi,j→k = ((t2bi t2bj) // dmi,j→k // b2tk) /. {tk → t, Tk → T, τi|j → 0},
 kCi = (Ci // b2ti) /. Ti → T,
 k̄Ci = (C̄i // b2ti) /. Ti → T,
 kKinki = (Kinki // b2ti) /. {ti → t, Ti → T},
 k̄Kinki = (K̄inki // b2ti) /. {ti → t, Ti → T}]
```