Pensieve header: The Objects, for the lazy evaluation version.

Program

The Objects, for LazyEval

Program

"Define" code

Program

Define[lhs = rhs, ...] defines the lhs to be rhs, except that rhs is computed only once for each value of $k. Fancy Mathematica not for the faint of heart. Most readers should ignore.

Program

```
SetAttributes[Define, HoldAll];
Define[def_, defs__] := (Define[def]; Define[defs];);
Define[op_is__ = ℰ_] := Module[{SD, ii, jj, kk, isp, nis, nisp, sis}, Block[{i, j, k},
   ReleaseHold[Hold[
      SD[op_nisp,$k_Integer, PP_Boot@Block[{i, j, k}, op_isp,$k = Zip@ℰ; op_nis,$k]];
      SD[op_isp, op_{is},$k]; SD[op_sis__, op_{sis}];
     ] /. {SD → SetDelayed,
      isp → {is} /. {i → i_, j → j_, k → k_},
      nis → {is} /. {i → ii, j → jj, k → kk},
      nisp → {is} /. {i → ii_, j → jj_, k → kk_}
     }] ]]
```

Program

Booting Up QU

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, 1]_$k,
 bm_{i,j→k} = 𝔼_{{i,j}→{k}}[(β_i + β_j) b_k, (η_i + η_j) y_k, e^((e^-ϵ β_i -1) η_j y_k)]_$k]
```

Program

```
Define[R_{i,j} = 𝔼_{{}→{i,j}}[ħ a_j b_i, ħ x_j y_i, e^Sum[($k+1)/k (1 - e^γ ϵ ħ)^k (ħ y_i x_j)^k / (1 - e^k γ ϵ ħ), {k, 2, $k+1}]]_$k,
 R̄_{i,j} = 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})$k[-1] -
      Zip[((R̄_{{i,j},0})$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})][-1]]],
 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})$k[-1] -
      Zip[R_{1,2} // ((P_{{1,j},0})$k (P_{{i,2},$k-1})$k)][-1]]]]
```

Program

```
Define[aS_j = R̄_{i,j} ~B_i~ P_{i,j},
 āS_i = 𝔼_{{i}→{i}}[-a_i α_i, -x_i 𝒜_i ξ_i, 1 + If[$k == 0, 0, (āS_{{i},$k-1})$k[-1] -
      Zip[(āS_{{i},0})$k ~B_i~ aS_i ~B_i~ (āS_{{i},$k-1})$k][-1]]]]
```

Program

```
In[ ]:= Define[bS_i = R_{i,1} ~ B_1 ~ aS_1 ~ B_1 ~ P_{i,1},
  \overline{bS}_i = R_{i,1} ~ B_1 ~ \overline{aS}_1 ~ B_1 ~ P_{i,1},
  a\Delta_{i→j,k} = (R_{1,j} R_{2,k}) // bm_{1,2→3} // P_{3,i},
  b\Delta_{i→j,k} = (R_{j,1} R_{k,2}) // am_{1,2→3} // P_{i,3}]
```

Program

```
In[ ]:= Define[dm_{i,j→k} = (\mathbb{E}_{{i,j}→{i,j}}[\beta_i b_i + \alpha_j a_j, \eta_i y_i + \xi_j x_j, 1]
      (a\Delta_{i→1,2} // a\Delta_{2→2,3} // \overline{aS}_3) (b\Delta_{j→-1,-2} // b\Delta_{-2→-2,-3})) // (P_{-1,3} P_{-3,1} am_{2,j→k} bm_{i,-2→k}),
  dS_i = \mathbb{E}_{{i}→{1,2}}[\beta_i b_1 + \alpha_i a_2, \eta_i y_1 + \xi_i x_2, 1] // (\overline{bS}_1 aS_2) // dm_{2,1→i},
  d\Delta_{i→j,k} = (b\Delta_{i→3,1} a\Delta_{i→2,4}) // (dm_{3,4→k} dm_{1,2→j})]
```

Program

```
In[ ]:= Define[C_i = \mathbb{E}_{{}→{i}}[0, 0, B_i^{1/2} e^{-\hbar \epsilon a_i/2}]_{$k},
  \overline{C}_i = \mathbb{E}_{{}→{i}}[0, 0, B_i^{-1/2} e^{\hbar \epsilon a_i/2}]_{$k},
  Kink_i = (R_{1,3} \overline{C}_2) // dm_{1,2→1} // dm_{1,3→i},
  \overline{Kink}_i = (\overline{R}_{1,3} C_2) // dm_{1,2→1} // dm_{1,3→i}]
```

Program

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

Program

```
In[ ]:= Define[b2t_i = \mathbb{E}_{{i}→{i}}[\alpha_i a_i - \beta_i t_i / \gamma, \xi_i x_i + \eta_i y_i, e^{\epsilon \beta_i a_i/\gamma}]_{$k},
  t2b_i = \mathbb{E}_{{i}→{i}}[\alpha_i a_i - \tau_i \gamma b_i, \xi_i x_i + \eta_i y_i, e^{\epsilon \tau_i a_i}]_{$k}]
```

Program

The CU Definitions

Program

```
Define[cm_{i,j→k} = \mathbb{E}_{{i,j}→{k}}[
    a_k (\alpha_i + \alpha_j) + b_k (\beta_i + \beta_j),
    y_k (\eta_i + \eta_j / \mathcal{A}_i) + \gamma b_k \eta_j \xi_i + x_k (\xi_i / \mathcal{A}_j + \xi_j),
    e^{y_k \eta_j (e^{-\epsilon \beta_i}/(\mathcal{A}_i + \gamma \epsilon \mathcal{A}_i \eta_j \xi_i) - 1/\mathcal{A}_i) + \xi_i (x_k (e^{-\epsilon \beta_j}/(\mathcal{A}_j + \gamma \epsilon \mathcal{A}_j \eta_j \xi_i) - 1/\mathcal{A}_j) - \gamma b_k \eta_j)} (1 + \gamma \epsilon \eta_j \xi_i)^{a_k/\gamma + b_k/\epsilon}
  ]_{$k}]
```

Program

```
In[ ]:= Define[c\Delta_{i→j,k} = \mathbb{E}_{{i}→{j,k}}[(a_j + a_k) \alpha_i + (b_j + b_k) \beta_i, (y_j + y_k) \eta_i + (x_j + x_k) \xi_i, 1]_{$k}]
```

Program

```
In[ ]:= Define[cS_i = \mathbb{E}_{{i}→{1,2,3,4}}[-\beta_i b_2 - \alpha_i a_3, -\eta_i y_1 - \xi_i x_4, 1] // cm_{4,3→i} // cm_{i,2→i} // cm_{i,1→i}]
```

Program

The Knot Tensors

Program

In[•]:=

```
Define[kR_{i,j} = R_{i,j} // (b2t_i b2t_j) /. t_{i|j} → t,
 kR̄_{i,j} = R̄_{i,j} // (b2t_i b2t_j) /. {t_{i|j} → t, T_{i|j} → T},
 km_{i,j→k} = (t2b_i t2b_j) // dm_{i,j→k} // b2t_k /. {t_k → t, T_k → T, τ_{i|j} → 0},
 kC_i = C_i // b2t_i /. T_i → T,
 kC̄_i = C̄_i // b2t_i /. T_i → T,
 kKink_i = Kink_i // b2t_i /. {t_i → t, T_i → T},
 kKink‾_i = Kink‾_i // b2t_i /. {t_i → t, T_i → T}]
```