

Define[lhs = rhs, ...] defines the lhs to be rhs, except that rhs is computed only once for each value of \$k. Fancy Mathematica notation for the faint of heart. Most readers should ignore.

```

SetAttributes[Define, HoldAll];
Define[def_, defs_] := (Define[def]; Define[defs]);
Define[op_is_ = ε_] :=
Module[{SD, ii, jj, kk, isp, nis, nisp, sis},
Block[{i, j, k},
ReleaseHold[Hold[
SD[op_nisp, $k_Integer, PPBoot@Block[{i, j, k}, op_isp, $k = ε;
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_}
}]]]

```

```

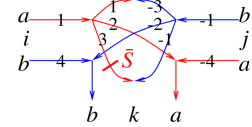
Define[aSi = (aσ_{i→2} R_{1,i}) // P_{1,2},
aSi = E[{i}→{i}] [-a_i α_i, -x_i A_i ε_i,
1 + If[$k == 0, 0, (aS_{i, $k-1}) $k [3] -
((aS_{i, 0}) $k // aSi // (aS_{i, $k-1}) $k) [3]]]]

```

```

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

```



The Drinfel'd double:

```

Define[
dm_{i,j→k} =
((SY_{i→4,4,1,1} // aΔ_{1→1,2} // aΔ_{2→2,3} // aS_3)
(SY_{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})

```

## The Objects

$\omega\epsilon\beta$ /objects

### Symmetric Algebra Objects

```

sm_{i,j→k} :=
E[{i,j}→{k}] [b_k (β_i + β_j) + t_k (τ_i + τ_j) + a_k (α_i + α_j) +
y_k (η_i + η_j) + x_k (ξ_i + ξ_j)];
sΔ_{i→j,k} :=
E[{i}→{j,k}] [β_i (b_j + b_k) + τ_i (t_j + t_k) + α_i (a_j + a_k) +
η_i (y_j + y_k) + ξ_i (x_j + x_k)];
sS_i := E[{i}→{i}] [-β_i b_i - τ_i t_i - α_i a_i - η_i y_i - ξ_i x_i];
se_i := E[{}→{i}] [0];
sη_i := E[{i}→{}] [0];
sσ_{i→j} := E[{i}→{j}] [β_i b_j + τ_i t_j + α_i a_j + η_i y_j + ξ_i x_j];
sY_{i→j,k,l,m} := E[{i}→{j,k,l,m}] [β_i b_k + τ_i t_k + α_i a_l + η_i y_j + ξ_i x_m];

```

### The CU Definitions

```

cλ = (η_i + (e^{-γ α_i - ε β_i} η_j) / (1 + γ ε η_j ξ_i)) y_k + (β_i + β_j + (Log[1 + γ ε η_j ξ_i] / ε)) b_k +
(α_i + α_j + (Log[1 + γ ε η_j ξ_i] / γ)) a_k + (e^{-γ α_j - ε β_j} ξ_i / (1 + γ ε η_j ξ_i) + ξ_j) x_k;
Define[cm_{i,j→k} = E[{i,j}→{k}] [cλ]]
Define[cσ_{i→j} = sσ_{i,j} /. τ_i → 0, cε_i = se_i, cη_i = sη_i,
cΔ_{i→j,k} = sΔ_{i,j,k},
cS_i = sS_i // sY_{i→1,2,3,4} // cm_{4,3→i} // cm_{i,2→i} // cm_{i,1→i}];

```

### Booting Up QU

```

Define[aσ_{i→j} = E[{i}→{j}] [a_j α_i + x_j ξ_i],
bσ_{i→j} = E[{i}→{j}] [b_j β_i + y_j η_i]]
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]]
Define[R_{i,j} = E[{}→{i,j}] [ħ a_j b_i + Σ_{k=1}^{$k+1} ((1 - e^{γ ε ħ})^k (ħ y_i x_j)^k) / (k (1 - e^{k γ ε ħ}))],
R_{i,j} = CF@E[{}→{i,j}] [-ħ a_j b_i, -ħ x_j y_i / B_i,
1 + If[$k == 0, 0, (R_{i,j, $k-1}) $k [3] -
((R_{i,j, 0}) $k R_{1,2} (R_{i,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_{i,1,3, 0}) $k (P_{i,2, $k-1}) $k)) [3]]]]]

```

```

Define[dσ_{i→j} = aσ_{i→j} bσ_{i→j},
de_i = se_i, dη_i = sη_i,
dS_i = sY_{i→1,1,2,2} // (bS_1 aS_2) // dm_{2,1→i},
dS_i = sY_{i→1,1,2,2} // (bS_1 aS_2) // dm_{2,1→i},
dΔ_{i→j,k} = (bΔ_{i→3,1} aΔ_{i→2,4}) // (dm_{3,4→k} dm_{1,2→j})]

```

```

Define[C_i = E[{}→{i}] [0, 0, B_i^{1/2} e^{-ħ ε a_i / 2}] $k,
C_i = E[{}→{i}] [0, 0, B_i^{-1/2} e^{ħ ε a_i / 2}] $k,
Kink_i = (R_{1,3} C_2) // dm_{1,2→1} // dm_{1,3→i},
Kink_i = (R_{1,3} C_2) // dm_{1,2→1} // dm_{1,3→i}]

```

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

```

Define[b2t_i = E[{i}→{i}] [α_i a_i + β_i (ε a_i - t_i) / γ + ξ_i x_i + η_i y_i],
t2b_i = E[{i}→{i}] [α_i a_i + τ_i (ε a_i - γ b_i) + ξ_i x_i + η_i y_i]]

```

### The Knot Tensors

```

Define[kR_{i,j} = R_{i,j} // (b2t_i b2t_j) /. t_i|j → t,
kR_{i,j} = R_{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,
kC_i = C_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}]

```

### Some of the Atoms.

$\omega\epsilon\beta$ /atoms

With  $A_i := e^{\alpha_i}$  and  $B_i = e^{-b_i}$ ,

```

PP_ := Identity; $k = 1; ħ = γ = 1;
Column[
(# → (ε = ToExpression[#];
Normal@Simplify[ε[[1]] + ε[[2]] + Log@ε[[3]]]) & @/
{"dm_{i,j→k}", "dΔ_{i→j,k}", "dS_i", "R_{i,j}", "P_{i,j}"}]

```