Pensieve header: Quantum sl_n.

References.

[1] Rosso “An Analogue of P.B.W. Theorem and the Universal R-matrix for $U_\hbar(sl(N+1))$”.

[2] Yamane “A Poincare-Birkhoff-Witt Theorem for Quantized Universal Enveloping Algebras of Type A_N”.

```
SetDirectory@"C:\\drorbn\\AcademicPensieve\\Projects\\SolvablePBW";
<< KnotTheory`
<< ../Profile/Profile.m
```

```
Loading KnotTheory` version of October 29, 2024, 10:29:52.1301.
Read more at http://katlas.org/wiki/KnotTheory.
This is Profile.m of http://www.drorbn.net/AcademicPensieve/Projects/Profile/.
This version: April 2020. Original version: July 1994.
```

```
δij__ := KroneckerDelta[ij];
SetAttributes[{SS}, HoldAll];
SS[ℰ_, op_] := PPSS@Collect[PPNormal@Normal@PPSeries@Series[ℰ, {ϵ, 0, $k}], ϵ, op];
SS[ℰ_] := SS[ℰ, Identity];
$N = 9; (* We hereby commit that we will never go higher than sl(9) *)
x[i_, j_] := x <> i <> j; a[i_] := a <> i; A[i_] := A <> i;
ξ[i_, j_] := ξ <> i <> j; α[i_] := α <> i; 𝔄[i_] := 𝔄 <> i;
y[i_, j_] := y <> i <> j; b[i_] := b <> i; B[i_] := B <> i;
η[i_, j_] := η <> i <> j; β[i_] := β <> i;
τ[s_Symbol] := First[Characters[SymbolName[s]]];
ι[s_Symbol] := FromDigits /@ Rest[Characters[SymbolName[s]]];
τ[s_[_]] := τ[s]; ι[s_[_]] := ι[s];
```

Greek - Latin duality:

```
Do[As[i, 1, $N - 1], As[j, i + 1, $N], x[i, j]* = ξ[i, j];
 ξ[i, j]* = x[i, j]; y[i, j]* = η[i, j];
 η[i, j]* = y[i, j]
]
Do[As[i, 1, $N - 1], A[i]* = 𝔄[i]; 𝔄[i]* = A[i]; a[i]* = α[i]; α[i]* = a[i];
 b[i]* = β[i]; β[i]* = b[i]]
(vs_List)* := (v ↦ v*) /@ vs;
(u_αβ__)* := (u*)αβ;
f[u_i_] := f[ui] = ToExpression["f" <> ToString[u]]i (*May conflict with τ,
  which assumes that generator names are a single letter (plus indices) *)
```

```
LatinVars = Flatten@{Table[As[i, 1, $N - 1], As[j, i + 1, $N], {x[i, j], y[i, j]}],
    Table[As[i, 1, $N - 1], {a[i], b[i]}]};
GreekVars = LatinVars*;
LatinVarPatten = (Alternatives @@ LatinVars)_;
GreekVarPatten = (Alternatives @@ GreekVars)_;
VarPattern = (Alternatives @@ (LatinVars ⋃ GreekVars))_;
```

```
Module[{i, s, p, c, d, χ, f, v},
  U2lRule = Union @@ Table[As[i, 1, $N - 1], {𝒜[i]^{p_-.}_{s_-} → 𝕖^{p α[i]_s}, B[i]^{p_-.}_{s_-} → 𝕖^{-p ℏ b[i]_s}}];
  l2URule =
   Append[Union @@ Table[As[i, 1, $N - 1], {𝕖^{d_-. + c_-. b[i]_{s_-}} → B[i]^{-c/ℏ}_s 𝕖^d, 𝕖^{d_-. + c_-. α[i]_{s_-}} → 𝒜[i]^c_s 𝕖^d}],
    𝕖^{χ_-} :↦ 𝕖^{Expand[l2U[χ]]}];
  U2l[ℰ_] := ℰ /. U2lRule;
  l2U[ℰ_] := ℰ //. l2URule;
  Do[As[i, 1, $N - 1], With[{αi = α[i], 𝒜i = 𝒜[i], bi = b[i], Bi = B[i]},
    D_{αi_s_}[f_] := ∂_{αi_s} f + 𝒜i_s ∂_{𝒜i_s} f;
    D_{bi_s_}[f_] := ∂_{bi_s} f - ℏ Bi_s ∂_{Bi_s} f;
   ]];
  D_{v_}[f_] := ∂_v f;
 ];
```

```
CF[ℰ_] := PP_CF@Module[{s1},
   s1 = If[$k == ∞,
     Expand[ℰ] /. 𝕖^{p_-} :↦ 𝕖^{Expand@p},
     Expand[ℰ] /. 𝕖^{p_-} :↦ 𝕖^{Expand@p} /. q^{n_-.} :↦ Sum_{k=0}^{$k} (ℏ ϵ γ n)^k / k! /. ϵ^{k_-.} /; k > $k :↦ 0
    ]
   (*Expand@Collect[s1,Cases[s1,VarPattern,∞],CCF]/.CCF→Factor*)
  ]
```

Weights: (Need confirmation!!!)

```
Do[As[i, 1, $N - 1], As[j, i + 1, $N],
  Wt[x[i, j]] = {1, i - j}; Wt[y[i, j]] = {0, -(i - j)};
  Wt[ξ[i, j]] = {0, -(i - j)}; Wt[η[i, j]] = {1, (i - j)};
 ];
Do[As[i, 1, $N - 1],
  Wt[a[i]] = {1, 0}; Wt[b[i]] = {0, 0};
  Wt[α[i]] = {0, 0}; Wt[β[i]] = {1, 0};
 ];
Wt[u_] := Wt[u];
```

```
(* q~𝕖^{ℏ γ ϵ}; Ai[n]~𝕖^{ℏ ϵ ai}; *)
```

```
$n = 5; $k = ∞; $E := {$n, $k}
```

```
$EPBWGens := Flatten@{
   Table[As[i, 1, $n - 1], As[j, i + 1, $n], x[i, j]],
   Table[As[i, 1, $n - 1], {a[i], A[i][_]}]
  }
$EPBWRule := Module[{k = 0}, Flatten@Table[{g → ++k, g_i_ → {i, k}}, {g, $EPBWGens}]];
x_ ≼ y_ := OrderedQ[{x, y} /. $EPBWRule]; x_ ≺ y_ := ! OrderedQ[{y, x} /. $EPBWRule];
$PBWGens := DeleteCases[$EPBWGens, _[_]]
```

```
UCF[ℰ_] := PP_UCF@Module[{F, k}, Expand@Collect[
      ℰ //. {U_i_[l___, _[0], r___] → U_i[l, r]} //. If[$k == ∞, {}, {
        U_i_[l___, AB_[n_], r___] :↦ Sum[As[k, 0, $k],
          (-2)^k ħ^k ϵ^k n^k / k! U_i[l, Sequence @@ Table[a @@ ι[AB], k], r]]
        }],
      U_[___], F
     ] /. F → CF]
```

```
m_s__[0] = 0;
m_s__[x_Plus] := UCF[m_s /@ x];
m_i_→j_[ℰ_] := ℰ /. U_i → U_j;
m_i_,j_→k_[c_. U_i_[x___] U_j_[]] := c U_k[x];
m_i_,j_→k_[c_. U_i_[] U_j_[y___]] := c U_k[y];
m_i_,j_→k_[c_. U_i_[xx___, x_[n1_]] U_j_[x_[n2_], yy___]] := UCF[c U_i[xx, x[n1 + n2]] U_j[yy]] // m_i,j→k;
m_i_,j_→k_[c_. U_i_[xx___, x_] U_j_[y_, yy___]] := If[x ≤ y,
   c U_k[xx, x, y, yy],
   ((c U_i[xx] RR_j[x, y] // UCF // m_i,j→i) U_j[yy] // UCF // m_i,j→k) // UCF
  ];
m_s__[ℰ_] := m_s[UCF[ℰ]];
```

```
Supp[ℰ_] := Union@Cases[{ℰ}, U_i_[___] :↦ i, ∞];
```

```
Unprotect[NonCommutativeMultiply];
NonCommutativeMultiply[x_] := x;
x_ ⊠ y_ := PP_NCM@Module[{i, is = Supp[x] ⋂ Supp[y], σ, z, k},
    z = x; Do[z = m_i→σ@i[z], {i, is}];
    z = Expand[y z];
    If[$k < ∞, z = z /. ϵ^k_. /; k > $k :↦ 0];
    Do[z = m_σ@i,i→i[z], {i, is}]; z];
UB[x_, y_] := PP_UB@UCF[x ⊠ y - y ⊠ x];
```

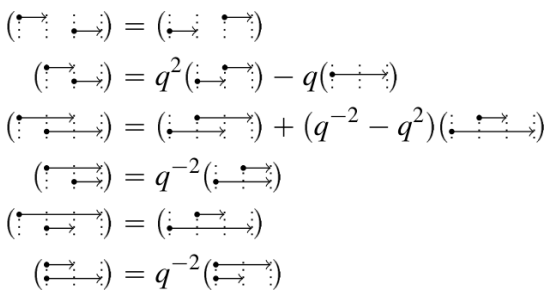
```
Δ_i_→j_,k_[ℰ_] := UCF@Module[{j1, k1}, ℰ /. {
      U_i[] → U_j[] U_k[],
      U_i[f_, fs___] :↦ m_k,k1→k@m_j,j1→j[Δ_j,k[f] Δ_i→j1,k1[U_i[fs]]]
     }];
Δ_i_→j_,k_,l__[ℰ_] := ℰ // Δ_i→j,k // Δ_k→k,l
```

```
S_i_[ℰ_] := UCF@Module[{i1}, ℰ /. {
      U_i[f_, fs___] :↦ m_i1,i→i[S_i[f] S_i1[U_i1[fs]]]
     }];
S̄_i_[ℰ_] := UCF@Module[{i1}, ℰ /. {
      U_i[f_, fs___] :↦ m_i1,i→i[S̄_i[f] S̄_i1[U_i1[fs]]]
     }];
```

The 𝔸-algebra

```
RR_k_[ai_, aj_] /; τ[ai] == τ[aj] == "a" := U_k[aj, ai];
RR_k_[ai_, Aj_[n_]] /; τ[ai] == "a" ∧ τ[Aj] == "A" := U_k[Aj[n], ai];
RR_k_[Aj_[n_], ai_] /; τ[ai] == "a" ∧ τ[Aj] == "A" := U_k[ai, Aj[n]];
RR_k_[Ai_[m_], Aj_[n_]] /; τ[Ai] == τ[Aj] == "A" := U_k[Aj[n], Ai[m]];
RR_k_[am_, xij_] /; τ[am] == "a" ∧ τ[xij] == "x" := Module[{i, j, m},
   {i, j} = ι@xij; {m} = ι@am;
   U_k[xij, am] + γ (δ_m,i - δ_m+1,i - δ_m,j + δ_m+1,j) U_k[xij] / 2
  ];
RR_k_[Am_[n_], xij_] /; τ[Am] == "A" ∧ τ[xij] == "x" := Module[{i, j, m},
  {i, j} = ι@xij; {m} = ι@Am; q^-n (δ_m,i-δ_m+1,i-δ_m,j+δ_m+1,j) U_k[xij, Am[n]]
 ]
```

We take the reduction rules RR for xij's from Yamane page 509, rewritten in Monoblog/260928:



```
RR_k_[xmn_, xij_] /; τ[xij] == τ[xmn] == "x" := Module[{i, j, m, n},
  {i, j} = ι@xij; {m, n} = ι@xmn;
  Which[
   (i == m ∧ j < n) ∨ (i < m < j ∧ n == j), q^-2 U_k[xij, xmn], (* I∨III *)
   (i < m < j ∧ n < j) ∨ (j < m), U_k[xij, xmn], (* II∨VI *)
   i < m < j ∧ n > j, U_k[xij, xmn] + (q^-2 - q^2) U_k[x[i, n], x[m, j]], (* IV *)
   j == m, q^2 U_k[xij, xmn] - q U_k[x[i, n]] (* V *)
  ]
 ]
```

The Co-Product, Co-Associativity, and the Hopf Axiom

According to DeltaUpTo9.nb, the following formulas produce an easy to see pattern.

```
Do[As[i, 1, $n - 1],
  Δ_j_,k_[a[i]] = U_j[a[i]] U_k[] + U_j[] U_k[a[i]];
  Δ_j_,k_[A[i][n_]] = U_j[A[i][n]] U_k[A[i][n]];
  Δ_j_,k_[x[i, i + 1]] = U_j[x[i, i + 1]] U_k[A[i][1]] + U_j[A[i][-1]] U_k[x[i, i + 1]]
 ];
Do[As[i, 1, $n - 2], As[n, i + 2, $n],
 Δ_j_,k_[x[i, n]] = m_k,k1→k[m_j,j1→j[
    q Δ_j,k[x[i, n - 1]] Δ_j1,k1[x[n - 1, n]] - q^-1 Δ_j,k[x[n - 1, n]] Δ_j1,k1[x[i, n - 1]]
   ]]
]
```

The Antipode

```
Do[As[i, 1, $N - 1],
  S_j_[a[i]] = -U_j[a[i]];
  S_j_[A[i][n_]] = U_j[A[i][-n]];
  S_j_[x[i, i + 1]] = -q^-2 U_j[x[i, i + 1]]; (*?*)
  Sb_j_[a[i]] = -U_j[a[i]];
  Sb_j_[A[i][n_]] = U_j[A[i][-n]];
  Sb_j_[x[i, i + 1]] = -q^2 U_j[x[i, i + 1]]; (*?*)
 ];
Do[As[i, 1, $N - 2], As[n, i + 2, $N],
 S_j_[x[i, n]] = m_{j1,j->j}[
   q S_j[x[i, n - 1]] S_j1[x[n - 1, n]] - q^-1 S_j[x[n - 1, n]] S_j1[x[i, n - 1]]];
 Sb_j_[x[i, n]] = m_{j1,j->j}[
   q Sb_j[x[i, n - 1]] Sb_j1[x[n - 1, n]] - q^-1 Sb_j[x[n - 1, n]] Sb_j1[x[i, n - 1]]]
]
```

The Generating Functions Engine

Canonical Forms:

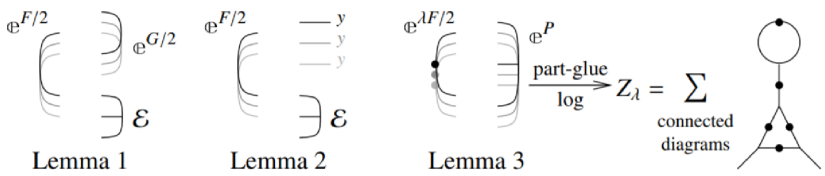
```
LogReduce[E_] := E /. c_ * Log[a_] :> Log@Factor[a^c] //. Log[a_] + Log[b_] :> Log@Factor[a b];
CF[E_IE] := CF /@ MapAt[LogReduce, E, 1];
CF[E : IE__[___]] := CF /@ E;
```

𝔼 operations:

```
E_IE[$] := Length[E] - 1; IE_[Es___][$] := IE[Es][$];
E_IE[k_Integer] := E[[k + 1]]; IE_[Es___][k_Integer] := {Es}[[k + 1]];
IE /: E1_IE ≡ E2_IE := Inner[CF@#1 == CF@#2 &, E1, E2, And];
IE_{d1_->r1_}[E1s___] ≡ IE_{d2_->r2_}[E2s___] ^:= (d1 == d2) ∧ (r1 == r2) ∧ (IE[E1s] ≡ IE[E2s]);
IE /: E1_IE * E2_IE := IE @@ Table[CF[E1[kk] + E2[kk]], {kk, 0, Min[E1[$], E2[$]]}];
IE_{d1_->r1_}[E1s___] IE_{d2_->r2_}[E2s___] ^:= IE_{(d1⋃d2)->(r1⋃r2)} @@ (IE[E1s] × IE[E2s]);
```

```
IE_{d1_->r1_}[E1s___] // IE_{d2_->r2_}[E2s___] := Module[{is = r1 ⋂ d2, lvs, llvs},
  lvs = Cases[{E1s}, (Alternatives @@ LatinVars)_i_ /; MemberQ[is, i], ∞] ⋃
    Cases[{E2s}, (Alternatives @@ GreekVars)_i_ /; MemberQ[is, i], ∞]*;
  llvs = lvs /. u_i_ :> u_$@i;
  IE_{(d1⋃Complement[d2,is])->(r2⋃Complement[r1,is])} @@ (Zip_{llvs⋃llvs*}[{(F /@ llvs*).(F /@ llvs), Times[
      IE[E1s] /. Thread[lvs -> llvs],
      IE[E2s] /. Thread[lvs* -> llvs*]
     ]}])
 ]
```

Zipping! Lemmas 2 and 3 are combined, yet they must be applied first to the middle weight variables and then to the heavy and light variables.



Comment. Zip3 of the outer variables must occur after all other operations are completed, because we must allow for gluings of the weight *n* variables in perturbations with the weight 0 variables in the coefficients of *Q*.

```
Zip_vs_[{ℱ_, ℰ_}] := PP_Zip@Module[{dvs = Select[vs, (Wt[#] == {0, 0} ∨ Wt[#] == {1, 0}) &]},
    {ℱ, ℰ} // Zip1_vs(*//Zip2_Complement[vs,dvs]*) // Zip2_dvs // Zip3_Complement[vs,dvs] // Zip3_dvs
   ] // Last
```

Getting rid of the

quadratic.

Lemma 1. With convergences left to the reader,

```
Zip1_{} = Identity;
Zip1_vs_@{ℱ_, 𝔼[Q_, P___]} := PP_Zip1@Module[{ℐ, F, G, u, v},
   (*Echo[{vs,ℱ,Q,P}];*)
   ℐ = IdentityMatrix@Length@vs;
   F = Table[If[Wt[u] + Wt[v] == {1, 0}, ∂_{F[u],F[v]}ℱ, 0], {u, vs}, {v, vs}];
   G = Table[If[Wt[u] + Wt[v] == {1, 0}, ∂_{u,v}Q, 0], {u, vs}, {v, vs}];
   (*Echo@MatrixForm[ℐ-G.F];*)
   {CF[(F /@ vs).(F.Inverse[ℐ - G.F]).(F /@ vs) / 2],
    𝔼[CF[Q - PowerExpand@Log[Det[ℐ - G.F]] / 2 - vs.G.vs / 2], P]}
  ]
```

Getting rid of linear

terms.

```
Zip2_{} = Identity;
Zip2_vs_@{ℱ_, 𝔼[Q_, P___]} := PP_Zip2@Module[{F, Y, u, v},
   (*EchoLabel["Zip2"][{vs,ℱ,Q,P}];*)
   F = Table[If[Wt[u] + Wt[v] == {1, 0}, CF[∂_{F[u],F[v]}ℱ], 0], {u, vs}, {v, vs}];
   Y = Table[∂_v Q, {v, vs}] /. (*AlsoUpper@*)Table[v → 0, {v, vs}];
   CF /@ ({ℱ, 𝔼[Q - Y.vs + Y.F.Y / 2, P]} /. (*AlsoUpper@*)Thread[vs → vs + F.Y])
  ]
```

Dealing with Feynman

diagrams.

Lemma 3. With an extra variable λ, $Z_\lambda := \log[\lambda F : \mathfrak{e}^P]_B$ satisfies
and is determined by the following PDE / IVP:

$$Z_0 = P \quad \text{and} \quad \partial_\lambda Z_\lambda = \frac{1}{2}\sum_{i,j\in B} F_{ij}\left(\partial_{z_i}\partial_{z_j}Z_\lambda + (\partial_{z_i}Z_\lambda)(\partial_{z_j}Z_\lambda)\right).$$

```
Zip3vs_@{ℱ_, ℰ_𝔼} := PPZip3@Module[{F, u, v, Z, kk, jj, $m = 0, m, n},
  (*Echo[{"Zip3",vs,ℱ,ℰ}];*)
  Do[Z[0, kk] = ℰ⟦kk + 1⟧, {kk, 0, ℰ@$}];
  F[u_, v_] := F[u, v] = CF@If[Wt[u] + Wt[v] == {1, 0}, ∂F[u],F[v] ℱ, 0];
  Z[m_, kk_, u_] := Z[m, kk, u] = Du[Z[m, kk]];
  Z[m_, kk_, u_, v_] := Z[m, kk, u, v] = Dv[Z[m, kk, u]];
  For[m = 0, m ≤ 2 $m, ++m, For[kk = 0, kk ≤ ℰ@$, ++kk,
    Z[m + 1, kk] = PPZip3[m,kk]@CF@Sum[
        If[F[u, v] === 0, 0,
          F[u, v]/(2 (m + 1)) (Z[m, kk, u, v] + Sum[Z[n, jj, u] * Z[m - n, kk - jj, v], {n, 0, m},
              {jj, 0, kk}])],
        {u, vs}, {v, vs}];
    If[Z[m + 1, kk] =!= 0, $m = m + 1]
   ]];
  CF /@ ({
      ℱ - Sum[F[u, v] * f[u] * f[v] / 2, {u, vs}, {v, vs}],
      𝔼 @@ Table[Sum[Z[m, kk], {m, 0, $m}], {kk, 0, ℰ@$}]
     } /. (*AlsoUpper@*)Table[v → 0, {v, vs}])
 ]
```

“Define” Code

```
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[opnisp,$E_, PPBoot@Block[{i, j, k}, opisp,$E = ℰ; opnis,$E]];
    SD[opisp, o{is},$E]; SD[opsis__, o{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_}
   }] ]]
```

The Generating Functions for the 𝔸 operations

Ad, TT, Seepage

```
PotQ[0] = True;
PotQ[c_. Ui_[aj_] (EUs : U_[] ...)] /; (τ[aj] == "a") ∧ FreeQ[c, U] = True;
PotQ[x_Plus] := PotQ /@ (And @@ x);
```

```
Ad[x_][y_] /; PotQ[x] ∧ (Head[y] =!= Plus) := If[y === 0, 0, PPAdPoty[e^UB[y,x]/y y]];
Ad[x_][y_Plus] /; PotQ[x] := PPAdPotPlus[Ad[x] /@ y];
```

```
Ad[x_][y_] /; ¬ TrueQ[PotQ[x]] := PP_AdNil@Module[{k = 1, s = y, t = y},
   While[(t = UB[t, x]) =!= 0, s += t / k!; ++k];
   UCF[s]
  ]
TT_λ[] = 0; (* Translated Tangent *)
TT_λ[xs___, x_] := PP_TT@UCF[Ad[x][TT_λ[xs]] + ∂_λ x];
TT_λ[xs_List] := TT_λ @@ xs
𝔒_i[poly_] := PP_𝔒@Total[
    CoefficientRules[poly, #_i & /@ $PBWGens] /.
     (p_ → c_) :> c U_i @@ Flatten@MapThread[ConstantArray, {$PBWGens, p}]
   ];
𝔒_{i_,is__}[poly_] := 𝔒_i[𝔒_is[poly]];
ℙ_i[ℰ_] := ℰ /. U_i[xs___] :> Times @@ (#_i & /@ {xs});
ℙ_{i_,is__}[poly_] := ℙ_i[ℙ_is[poly]];
Seepage_i[{}, c_] = c;
Seepage_i[ℱs_List, P_] := PP_Seepage@Module[{lx},
    lx = $PBWGens[[Length@ℱs]];
    CF@Collect[P, lx_i, ℙ_i[Ad[Last[ℱs]] U_i[lx]][𝔒_i[Seepage_i[Most@ℱs, #]]]
       &]
   ];
Seepage_{i_,is_}[ℱs_List, ℱss__List, P_] := Seepage_i[ℱs, Seepage_is[ℱss, P]]
Invert_is__[ℰ_] := PP_Invert@Module[{A, B, U1, j}, (* ℰ=A U[]+ϵ B *)
   U1 = Times @@ (U_#[] & /@ {is});
   A = Coefficient[ℰ, U1] /. ϵ → 0;
   B = UCF[(ℰ - A U1)];
   UCF[A^-1 (U1 + Sum[As[j, 1, $k], (-1)^j NonCommutativeMultiply @@ ConstantArray[B / A, j]])]
  ]
TriangularDSolve[eqns_, inits_, fs_, λ_, CF_] :=
 PP_TriangularDSolve@Module[{ss = {}, reqns = eqns, rinits = inits, rfs = fs, s, p},
   While[rfs =!= {},
    {p} = FirstPosition[Count[#, Alternatives @@ rfs, ∞, Heads → True] & /@ reqns, 1];
    {{s}} = DSolve[reqns[[p]] ∧ rinits[[p]], rfs[[p]], λ];
    AppendTo[ss, s = s /. F_Function :> ReplacePart[F, 2 → CF[F[[2]]]]];
    reqns = Delete[reqns, p] /. s /. e_ == 0 :> CF[e] == 0;
    rinits = Delete[rinits, p];
    rfs = Delete[rfs, p];
   ];
   ss
  ]
```

$L_\lambda = \mathbb{E}(\lambda \xi_{i_1} x)\,\mathbb{E}(\lambda \xi_{i_2} x)$ Seepage is defined by: $\mathbb{E}(f_i x_i)\,\mathbb{O}(\text{Seepage}(\{f_i\}, A)) = \mathbb{O}(e^{f_i x_i} A)$

$TT(L_\lambda) = L^{-1} \partial_\lambda L_\lambda$ $S(B) = \text{Seepage}(\{f_i\}, B)$.

$R_\lambda = \mathbb{E}(f+g)$, $f = f_i x_i$ $g = \sum_{|w|>1} g_w w$ $e^f = F$ $e^g = G$ $R_\lambda = \mathbb{O}(FG)$ $\partial_\lambda F = F \partial_\lambda f$

$TT(R_\lambda) = R_\lambda^{-1}(\partial_\lambda R_\lambda) = \mathbb{O}(FG)^{-1}\mathbb{O}(\partial_\lambda FG) = S(G)^{-1}\mathbb{O}(F)^{-1}\mathbb{O}(F(\partial_\lambda f G + \partial_\lambda G)) = S(G)^{-1}S(\partial_\lambda f G + \partial_\lambda G)$

```
Define[amt_{i,j→k} = PP"amt"[$k,$n]@Module[{(*TTL, *)},
    ES[s_][ℰ_] := (EchoLabel[s][Short[ℰ, 2]]; ℰ); ES[s_][ℰ_] := ℰ;
    TTL =
     ES["TTL"]@
      PP"TTL"@TT_λ@Join[l1 = Table[If[τ[g] === "x", λ, 1] (g*)_i U_k[g], {g, $PBWGens}], l1 /. i → j];
    mons[-1] = {};
    mons[p_] := mons[p - 1] ⋃ Union@Cases[Coefficient[TTL, ϵ, p], U_k[___], ∞];
    ff = Table[
       Sum[ϵ^p f_{p,u_k}[λ], {p, 0, $k}],
       {u, $PBWGens}
      ] /. f_{p_,u_k}[λ] /; τ[u] == "a" :↦ If[p == 0, (u*)_i + (u*)_j, 0];
    fs = Flatten[Table[
       f_{p,ℙ_k[m]},
       {p, 0, $k}, {m, mons[p]}
      ]];
    gg = Sum[
      ϵ^p f_{p,ℙ_k[u]}[λ] ℙ_k[u],
      {p, 0, $k}, {u, Complement[mons[p], U_k /@ $PBWGens]}
     ];
    G = Series[e^gg, {ϵ, 0, $k}];
    𝒮[B_] := UCF[𝒪_k[Seepage_k[ff, UCF@B]]];
    𝒮G = ES["𝒮G"]@PP"𝒮G"@𝒮[G];
    TTR = ES["TTR"]@PP"TTR"@UCF[Invert_k[𝒮G] ⊠ 𝒮[(∂_λ (ff.(#_k & /@ $PBWGens)) G + ∂_λ G)]];
    diff = UCF[TTL - TTR];
    eqns = Flatten[Table[
       Coefficient[Coefficient[diff, m], ϵ, p] == 0,
       {p, 0, $k}, {m, mons[p]}
      ]];
    inits = (#[0] == 0) & /@ fs;
    ss = TriangularDSolve[eqns, inits, fs, λ, CF];
    𝔼_{i,j}→{k} @@ Table[As[p, 0, $k], Plus[
       If[p == 0, Sum[As[m, 1, $n - 1], (α[m]_i + α[m]_j) a[m]_k], 0],
       Total[Cases[fs, f_{p,_}]] /. f_{p_,u_} :↦ u (f_{p,u} /. ss)[λ] /. λ | ħ | γ → 1]
     ]
   ]
]
```

```
Define[aΔt_{i→j,k} = PP_"aΔt"[$k,$n]@Module[{(*TTL, *)},
    ES[s_][ℰ_] := (EchoLabel[s][Short[ℰ, 2]]; ℰ);
    TTL = TT_λ@Table[UCF[If[τ[g] === "x", λ, 1] (g^*)_i Δ_{i→j,k}[U_i[g]]], {g, $PBWGens}];
    mons[-1] = {};
    mons[p_] :=
     mons[p - 1] ⋃ Union@Cases[Coefficient[TTL, ϵ, p], c_. U_j[x___] U_k[y___] :↦ U_j[x] U_k[y], ∞];
    ffj = Table[
       Sum[ϵ^p f_{p,u_j}[λ], {p, 0, $k}],
       {u, $PBWGens}
      ] /. f_{p_,u_j}[λ] /; τ[u] == "a" :↦ If[p == 0, (u^*)_i, 0];
    ffk = ffj /. j → k;
    fs = Flatten[Table[
       f_{p,𝕡_{j,k}[m]},
       {p, 0, $k}, {m, mons[p]}
      ]];
    gg = Sum[
      ϵ^p f_{p,𝕡_{j,k}[u]}[λ] 𝕡_{j,k}[u],
      {p, 0, $k}, {u, Complement[mons[p], U_j[] (U_k /@ $PBWGens) ⋃ U_k[] (U_j /@ $PBWGens)]}
     ];
    G = Series[e^gg, {ϵ, 0, $k}];
    𝒮[B_] := UCF[𝒪_{j,k}[Seepage_{j,k}[ffj, ffk, UCF@B]]];
    𝒮G = PP_"𝒮G"@𝒮[G];
    BB = PP_"BB"[∂_λ (ffj.(#_j & /@ $PBWGens) + ffk.(#_k & /@ $PBWGens)) G + ∂_λ G];
    𝒮BB = PP_"𝒮BB"@𝒮[BB];
    TTR = PP_"TTR"@UCF[Invert_{j,k}[𝒮G] ⊠ 𝒮BB];
    diff = UCF[TTL - TTR];
    eqns = Flatten[Table[
       Coefficient[Coefficient[diff, m], ϵ, p] == 0,
       {p, 0, $k}, {m, mons[p]}
      ]];
    inits = (#[0] == 0) & /@ fs;
    ss = TriangularDSolve[eqns, inits, fs, λ, CF];
    𝔼_{{i}→{j,k}} @@ Table[As[p, 0, $k], Plus[
       If[p == 0, Sum[As[m, 1, $n - 1], α[m]_i (a[m]_j + a[m]_k)], 0],
       Total[Cases[fs, f_{p,_}]] /. f_{p_,u_} :↦ u (f_{p,u} /. ss)[λ] /. λ | ℏ | γ → 1]
     ]
   ]]
```

```
Define[aSt_i = Module[{(*TTL, *)},
   ES[s_][ℰ_] := (EchoLabel[s][Short[ℰ, 2]]; ℰ);
   TTL = TT_λ@Reverse@Table[UCF[If[τ[g] === "x", λ, 1] (g^*)_i S_i[U_i[g]]], {g, $PBWGens}];
   mons[-1] = {};
   mons[p_] := mons[p - 1] ⋃ Union@Cases[Coefficient[TTL, ϵ, p], U_i[x___] :> U_i[x], ∞];
   ff = Table[
      Sum[ϵ^p f_{p,u_i}[λ], {p, 0, $k}],
      {u, $PBWGens}
     ] /. f_{p_,u_i}[λ] /; τ[u] == "a" :> If[p == 0, -(u^*)_i, 0];
   fs = Flatten[Table[
      f_{p,ℙ_i[m]},
      {p, 0, $k}, {m, mons[p]}
     ]];
   gg = Sum[
     ϵ^p f_{p,ℙ_i[u]}[λ] ℙ_i[u],
     {p, 0, $k}, {u, Complement[mons[p], (U_i /@ $PBWGens)]}
    ];
   G = Series[e^{gg}, {ϵ, 0, $k}];
   𝒮[B_] := UCF[𝒪_i[Seepage_i[ff, UCF@B]]];
   𝒮G = 𝒮[G];
   TTR = UCF[Invert_i[𝒮G] ⊠ 𝒮[(∂_λ (ff.(#_i & /@ $PBWGens)) G + ∂_λ G)]];
   diff = UCF[TTL - TTR];
   eqns = Flatten[Table[
      Coefficient[Coefficient[diff, m], ϵ, p] == 0,
      {p, 0, $k}, {m, mons[p]}
     ]];
   inits = (#[0] == 0) & /@ fs;
   ss = TriangularDSolve[eqns, inits, fs, λ, CF];
   𝔼_{{i}→{i}} @@ Table[As[p, 0, $k], Plus[
      If[p == 0, Sum[As[m, 1, $n - 1], -α[m]_i (a[m]_i)], 0],
      Total[Cases[fs, f_{p,_}]] /. f_{p_,u_} :> u (f_{p,u} /. ss)[λ] /. λ | ħ | γ → 1]
    ]
  ]]
```

```
Define[aSt_i = Module[{(*TTL, *)},
   TTL = TT_λ@Reverse@Table[UCF[If[τ[g] === "x", λ, 1] (g^*)_i S_i[U_i[g]]], {g, $PBWGens}];
   mons[-1] = {};
   mons[p_] := mons[p - 1] ⋃ Union@Cases[Coefficient[TTL, ϵ, p], c_. U_i[x___] :> U_i[x], ∞];
   ff = Table[
      Sum[ϵ^p f_{p,u_i}[λ], {p, 0, $k}],
      {u, $PBWGens}
     ] /. f_{p_,u_i}[λ] /; τ[u] == "a" :> If[p == 0, -(u^*)_i, 0];
   fs = Flatten[Table[
      f_{p,ℙ_i[m]},
      {p, 0, $k}, {m, mons[p]}
     ]];
   gg = Sum[
     ϵ^p f_{p,ℙ_i[u]}[λ] ℙ_i[u],
     {p, 0, $k}, {u, Complement[mons[p], (U_i /@ $PBWGens)]}
    ];
   G = Series[e^gg, {ϵ, 0, $k}];
   𝒮[B_] := UCF[𝕆_i[Seepage_i[ff, UCF@B]]];
   𝒮G = 𝒮[G];
   TTR = UCF[Invert_i[𝒮G] ⊠ 𝒮[(∂_λ (ff. (#_i & /@ $PBWGens)) G + ∂_λ G)]];
   diff = UCF[TTL - TTR];
   eqns = Flatten[Table[
      Coefficient[Coefficient[diff, m], ϵ, p] == 0,
      {p, 0, $k}, {m, mons[p]}
     ]];
   inits = (#[0] == 0) & /@ fs;
   ss = TriangularDSolve[eqns, inits, fs, λ, CF];
   𝔼_{i}→{i} @@ Table[As[p, 0, $k], Plus[
      If[p == 0, Sum[As[m, 1, $n - 1], -α[m]_i a[m]_i], 0],
      Total[Cases[fs, f_{p,_}]] /. f_{p_,u_} :> u (f_{p,u} /. ss)[λ] /. λ | ħ | γ → 1]
    ]
  ]]
```

```
(*Splitter 𝔹⊗𝔸 *)
sϒ_{i_→j_,k_,l_,m_} := 𝔼_{i}→{j,k,l,m}[
  Sum[η[r, s]_i y[r, s]_j + ξ[r, s]_i x[r, s]_m, {r, 1, $n - 1}, {s, r + 1, $n}] +
   Sum[α[r]_i a[r]_l + β[r]_i b[r]_k, {r, 1, $n - 1}], Sequence @@ ConstantArray[0, $k]
 ]
```

$P : \mathbb{A}\otimes\mathbb{B} \to \mathbb{Q}$ and $R \in \mathbb{B}\otimes\mathbb{A}$:

```
P_{i_,j_} := 𝔼_{i,j}→{} @@ PadRight[
   {Sum[η[r, s]_j ξ[r, s]_i, {r, 1, $n - 1}, {s, r + 1, $n}] + Sum[β[r]_j α[r]_i, {r, 1, $n - 1}]}, $k + 1]
R_{i_,j_} := 𝔼_{}→{i,j} @@ PadRight[
   {Sum[y[r, s]_i x[r, s]_j, {r, 1, $n - 1}, {s, r + 1, $n}] + Sum[b[r]_i a[r]_j, {r, 1, $n - 1}]}, $k + 1]
```

```
Define[bmt_{j,k→i} = Module[{i0, i1, j1, k1}, (R_{i0,i1} // aΔt_{i1→j1,k1} // P_{j1,j} // P_{k1,k}) /. i0 → i],
 bΔt_{k→i,j} = Module[{i1, j1, k1, k2}, ((R_{i,i1} R_{j,j1} // amt_{i1,j1→k1}) // P_{k1,k2}) /. k2 → k],
 bSt_i = Module[{i1, i2}, (R_{i,i1} // aSt_{i1} // P_{i1,i2}) /. i2 → i],
 ¯bSt_i = Module[{i1, i2}, (R_{i,i1} // ¯aSt_{i1} // P_{i1,i2}) /. i2 → i]]
Define[¯R_{i,j} = R_{i,j} // bSt_i]
```

```
Define[
 dm_{i,j→k} = Module[{i1, i2, i3, i4},
    ((sϒ_{i→i4,i4,i1,i1} // aΔt_{i1→i1,i2} // aΔt_{i2→i2,i3} // ¯aSt_{i3})
       (sϒ_{j→-i1,-i1,-i4,-i4} // bΔt_{-i1→-i1,-i2} // bΔt_{-i2→-i2,-i3})) // (P_{i1,-i3} P_{i3,-i1} amt_{i2,-i4→k} bmt_{i4,-i2→k})]]
Define[dS_i = sϒ_{i→1,1,2,2} // (bSt_1 ¯aSt_2) // dm_{2,1→i},
 ¯dS_i = sϒ_{i→1,1,2,2} // (¯bSt_1 aSt_2) // dm_{2,1→i},
 dΔ_{i→j,k} = (bΔt_{i→1,3} aΔt_{i→4,2}) // (dm_{3,4→k} dm_{1,2→j})]
(*Notice we take co-opposite on the 𝔸-side! to make the double*)
```