Aw-Calculus Programs for the WKO4 Paper

Pensieve header: Aw-calculus programs for the WKO4 paper.

“d” is “ht”: along tube strands, heads appear before tails.

Utilities

```
⟨{λ___}⟩ := ⟨λ⟩;
Support[⟨λ___⟩] := First /@ {λ};
λ_ \ key_ := DeleteCases[λ, key → _];
λ_ \ keys_List := Fold[#1 \ #2 &, λ, keys];
⟨λ___⟩s_ := s /. {λ};
(λ_AngleBracket)[d_] := ⟨Table[u → λu[d], {u, Support[λ]}]⟩;
Crop[λ_AngleBracket, d_] := MapAt[Crop[#, d] &, λ, {All, 2}];
AngleBracket /:
 λ1_AngleBracket ≡ λ2_AngleBracket /; Support[λ1] == Support[λ2] :=
 ⟨Table[s → λ1s ≡ λ2s, {s, Support[λ1]}]⟩
AngleBracket /: Plus[λs__AngleBracket] := ⟨Table[
     u → Total[#u & /@ {λs}],
     {u, Union @@ (Support[#] & /@ {λs})}
    ]⟩;
AngleBracket /: c_ * λ_AngleBracket := ⟨Table[
     u → Expand[c λu],
     {u, Support[λ]}
    ]⟩;
AngleBracket /: λ_AngleBracket // DerivationSeries[f_, ld_LieDerivation] :=
  MapAt[DerivationSeries[f, ld], λ, {All, 2}];
AngleBracket /: λ_AngleBracket // mor_LieMorphism := MapAt[mor, λ, {All, 2}];
AngleBracket /: Integrate[λ_AngleBracket, {s_, s0_, s1_}] :=
  ⟨Table[u → Integrate[λu, {s, s0, s1}], {u, Support[λ]}]⟩;
AngleBracket /: CC[⟨λ___⟩] := CC[LW /@ First /@ {λ}, Last /@ {λ}];
λ_AngleBracket // DegreeScale[h_] := MapAt[DegreeScale[h], λ, {All, 2}];
deg /: (h_)^deg := DegreeScale[h];
```

Tangential Derivations, the Tangential Bracket tb, div and j

```
TangentialDerivation[⟨λ___⟩] :=
  LieDerivation[{λ} /. (s_ → λs_) ⧴ (LW[s] → b[LW[s], λs])];
tb[λ1_AngleBracket, λ2_AngleBracket] /; Support[λ1] == Support[λ2] := ⟨Table[
    s →
     b[λ1s, λ2s] + TangentialDerivation[λ1][λ2s] - TangentialDerivation[λ2][λ1s],
    {s, Support[λ1]}
   ]⟩;
div[λ_AngleBracket] := Sum[divs[λs], {s, Support[λ]}];
j[λ_AngleBracket] :=
  DerivationSeries[(ⅇ^der - 1)/der, TangentialDerivation[λ]][div[λ]];
```

Evaluating Lie Series in ⟨…⟩

```
LieMorphism[rules__Rule, keys_AngleBracket, br_] :=
  LieMorphism[{rules}, keys, br];
LieMorphism[rules_List, keys_AngleBracket, br_] := New[LieMorphism[mor],
   mor[Support] = First /@ rules;
   (mor[w_LW] /; Deg[w] == 1) := (mor[w] = w /. rules);
   mor[w_LW] := (mor[w] = br @@ (mor /@ LyndonFactorization[w]));
   mor[expr_][d_] := Expand[expr /. w_LW ⧴ mor[w][d]];
   mor[ls_LieSeries] := mor[ls] = Module[{ser},
      ⟨Table[
        ReleaseHold[Hold[
            ser[] = Hold[AngleBracketFromLieMorphism[mor, ls, ss]];
            ser[d_Integer] := ser[d] = (Sum[mor[ls[k]][d], {k, 1, d}])_ss ;
            ss → LieSeries[ser]
           ] /. {ss → s, ser → Unique[LieSeriesInAngleBracket]}],
        {s, List @@ keys}
       ]⟩]
  ];
BCH[x_, y_, keys_AngleBracket, br_] :=
  LieMorphism[{LW["x"] → x, LW["y"] → y}, keys, br][BCHBase];
adSeries[f_, λ1_AngleBracket, br_][λ2_AngleBracket] /;
   Support[λ1] == Support[λ2] :=
  LieMorphism[{LW["x"] → λ1, LW["y"] → λ2}, ⟨Support[λ1]⟩, br][
   adSeries[f, LW["x"]][LW["y"]]];
```

The AT Presentation E_l of A^w

```
El /: El[λ1_, ω1_] ⋃ El[λ2_, ω2_] /; Support[λ1] ⋂ Support[λ2] == {} :=
  El[λ1 ⋃ λ2, ω1 + ω2];
El /: El[λ1_, ω1_] ** El[λ2_, ω2_] /; Support[λ1] == Support[λ2] :=
  El[BCH[λ1, λ2, 〈Support[λ1]〉, tb],
   ω1 + DerivationExp[TangentialDerivation[λ1]][ω2]];
El[λ_, ω_] // dA := El[-λ, DerivationExp[TangentialDerivation[λ]][ω] - j[λ]];
El[λ_, ω_] // dS := El[
   -λ // (-1)^deg,
   (DerivationExp[TangentialDerivation[λ]][ω] - j[λ]) // (-1)^deg
  ];
```

The KBH Presentation E_s of A^w

```
Es /: Es[λ1_, ω1_] ⋃ Es[λ2_, ω2_] /; Support[λ1] ⋂ Support[λ2] == {} :=
  Es[λ1 ⋃ λ2, ω1 + ω2];

tσ[us_List → vs_List][ser_LieSeries | ser_CWSeries | ser_AngleBracket] :=
  ser // LieMorphism[Thread[(LW /@ us) → (LW /@ vs)]];
tσ[u_, v_] := tσ[{u} → {v}];
tσ[us_List → vs_List][ξ_Es] := tσ[us → vs] /@ ξ;
hσ[xs_List → ys_List][λ_AngleBracket] :=
  Union[λ \ xs, 〈Thread[ys → Table[λ_x, {x, xs}]]〉];
hσ[x_, y_] := hσ[{x} → {y}];
hσ[xs_List → ys_List][Es[λ_, ω_]] := Es[λ // hσ[xs → ys], ω];
dσ[as_List → bs_List][ξ_] := ξ // tσ[as → bs] // hσ[as → bs];
dσ[a_, b_][ξ_] := ξ // tσ[a, b] // hσ[a, b];

tm[u_, v_, w_][λ_AngleBracket] := λ // LieMorphism[LW@u → LW@w, LW@v → LW@w];
tm[u_, v_, w_][Es[λ_, ω_]] := LieMorphism[LW@u → LW@w, LW@v → LW@w] /@ Es[λ, ω];
hm[x_, y_, z_][λ_AngleBracket] := Union[λ \ {x, y}, 〈z → BCH[λ_x, λ_y]〉];
hm[x_, y_, z_][Es[λ_, ω_]] := Es[λ // hm[x, y, z], ω];
tha[u_LW, x_][λ_AngleBracket] := λ // RC_u[λ_x];
tha[u_LW, x_][Es[λ_, ω_]] := Es[λ // tha[u, x], (ω + J_u[λ_x]) // RC_u[λ_x]];
dm[a_, b_, c_][ξ_] := ξ // tha[LW@a, b] // tm[LW@a, LW@b, LW@c] // hm[a, b, c];
dm[a_, b_, rest__, c_][ξ_] := ξ // dm[b, rest, b] // dm[a, b, c];

Es /: Es[λ1_, ω1_] ** Es[λ2_, ω2_] /; Support[λ1] == Support[λ2] := Module[
   {supp, temps, ξ},
   supp = Support[λ1];
   temps = Complement[Characters["0123456789abcdefghijklmnopqrstuvwxyz"],
      ToString /@ supp][[1 ;; Length@supp]];
   ξ = Es[λ1, ω1] ⋃ (Es[λ2, ω2] // dσ[supp → temps]);
   MapThread[(ξ = ξ // dm[#1, #2, #1]) &, {supp, temps}] // Last
  ];
```

```
tA[u_][expr_] := expr;
hA[x_][Es[λ_, ω_]] := Es[Union[λ \ x, ⟨x → -λx⟩], ω];
dA[a_][μ_] := μ // hA[a] // tha[LW@a, a];
dA[a_, rest__][μ_] := μ // dA[a] // dA[rest];
Es[λ_, ω_] // dA := Es[λ, ω] // (dA @@ Support[λ])

tS[u_][λ_AngleBracket] :=
  ⟨Table[x → LieMorphism[LW@u → -LW@u][λx], {x, Support[λ]}]⟩;
tS[u_][Es[λ_, ω_]] := Es[λ // tS[u], ω // LieMorphism[LW@u → -LW@u]];
hS[x_][Es[λ_, ω_]] := Es[Union[λ \ x, ⟨x → -λx⟩], ω];
dS[a_][μ_] := μ // tS[a] // hS[a] // tha[LW@a, a];
dS[a_, rest__][μ_] := μ // dS[a] // dS[rest];
Es[λ_, ω_] // dS := Es[λ, ω] // (dS @@ Support[λ])
```

The $E_l \leftrightarrow E_s$ Conversions

```
Γ[-1, λ_AngleBracket, _] := ⟨Table[s → MakeLieSeries[0], {s, Support[λ]}]⟩;
Γ[n_, λ_AngleBracket, t_] := Γ[n, λ, t] = Module[{τ, Γ0},
    Γ0 = Γ[n - 1, λ, τ];
    ⟨Table[
      s → ∫_0^t (λs // DerivationExp[-τ TangentialDerivation[λ]] //
            adSeries[ad/(e^ad - 1), Γ0s]) dτ,
      {s, Support[λ]}
     ]⟩
   ];
Γ[λ_AngleBracket, t_] := Γ[λ, t] = ⟨Table[
     s → New[LieSeries[ser],
       With[{s = s}, ser[d_Integer] := ser[d] = (Γ[d - 1, λ, t]s)[d]];
      ],
     {s, Support[λ]}
    ]⟩;
Γ[λ_AngleBracket] := Γ[λ, 1];
Γi[λ_AngleBracket] := Γi[λ] = ⟨Table[With[{s = s},
      s → New[LieSeries[ser],
        ser[d_Integer] := ser[d] = λs[d] - Γ[Crop[Γi[λ], d - 1]]s[d]
       ]],
     {s, Support[λ]}
    ]⟩;
Γ /: Γ^-1 = Γi;
Es[El[λ_, ω_]] := Es[Γ[λ], ω];
El[Es[λ_, ω_]] := El[Γ^-1[λ], ω];
```

```
Λ[-1, λ_AngleBracket, _] := 〈Table[s → MakeLieSeries[0], {s, Support[λ]}]〉;
Λ[n_, λ_AngleBracket, t_] := Λ[n, λ, t] = Module[{τ, Λ0},
    Λ0 = Λ[n - 1, λ, τ];
    (* I'm confused why the first
     two options below do not give the same answer *)
    ∫_0^t (λ // DerivationExp[TangentialDerivation[Λ0]] //
        adSeries[ad/(e^ad - 1), Λ0, tb]) dτ
    (*ϵ1∫_0^t (λ // CC[ϵ2 τ λ] // adSeries[ϵ3 ad/(e^(ϵ3 ad) - 1), ϵ4 Λ0, tb]) dτ*)
    (*ϵ1∫_0^t (λ // DerivationExp[ϵ2 TangentialDerivation[Λ0]] //
         adSeries[ϵ3 ad/(e^(ϵ3 ad) - 1), ϵ4 Λ0, tb]) dτ*)
    (*ϵ1∫_0^t (λ // DerivationExp[ϵ2 τ TangentialDerivation[λ]] //
         adSeries[ϵ3 ad/(e^(ϵ3 ad) - 1), ϵ4 Λ0, tb]) dτ*)
   ];
Λ[λ_AngleBracket, t_] := Λ[λ, t] = 〈Table[
      s → New[LieSeries[ser],
        With[{s = s}, ser[d_Integer] := ser[d] = (Λ[d - 1, λ, t]_s)[d]];
       ],
      {s, Support[λ]}
     ]〉;
Λ[λ_AngleBracket] := Λ[λ, 1];

(*Λ[λ_AngleBracket] :=
 Module[{τ}, ∫_0^1 (λ // DerivationExp[τ TangentialDerivation[λ]] //
       adSeries[ad/(e^ad - 1), -τ λ, tb]) dτ]*)
```