Pensieve header: μ-calculus programs, continued pensieve://2013-11/DoubleTree/.

Program

```
Domain[f_List] := First /@ f;
f_ \ key_ := DeleteCases[f, key → _];
f_ \ keys_List := Fold[#1 \ #2 &, f, keys];
f1_List ≡ f2_List := Domain[f1] === Domain[f2] && (And @@ (
        ((# /. f1) ≡ (# /. f2)) & /@ Domain[f1]
      ));
LieMorphism[mor_][f_List] := MapAt[LieMorphism[mor], f, {All, 2}];
Domain[M[f_List, _]] := Domain[f];
M[λ1_, ω1_] ≡ M[λ2_, ω2_] := (λ1 ≡ λ2) && (ω1 ≡ ω2);
M[_, ω_][W] := ω;
M[λ_, _][x_] := (x /. λ);

CC[u_, γ_LieSeries] := LieMorphism[⟨u⟩ → Ad[γ][⟨u⟩]];
CC_u_[γ_] := CC[u, γ];
RC[u_, γ_LieSeries, ub_][ser_] :=
  StableApply[LieMorphism[⟨u⟩ → Ad[γ][⟨ub⟩]], ser];
RC[u_, γ_LieSeries][ser_] := ser // RC[u, γ, ⟨υ⟩] // LieMorphism[⟨υ⟩ → ⟨u⟩];
RC_u_[γ_] := RC[u, γ];

J[u_LW, γ_] := Module[{s}, ∫_0^1 (γ // RC_u[s γ] // div_u // CC_u[-s γ]) ⅆs];
J[u_, γ_] := J[⟨u⟩, γ];
J_u_[γ_] := J[u, γ];

M /: M[λ1_, ω1_] ⋃ M[λ2_, ω2_] := M[λ1 ⋃ λ2, ω1 + ω2];
M /: M[λ1_, ω1_] M[λ2_, ω2_] := M[λ1 ⋃ λ2, ω1 + ω2];

tσ[us_List → vs_List][ser_LieSeries | ser_CWSeries | ser_List] :=
  ser // LieMorphism[Thread[(LW /@ us) → (LW /@ vs)]];
tσ[u_, v_] := tσ[{u} → {v}];
tσ[uvs__][μ_M] := tσ[uvs] /@ μ;
hσ[xs_List → ys_List][λ_List] := Union[λ \ xs, Thread[ys → (xs /. λ)]];
hσ[x_, y_] := hσ[{x} → {y}];
hσ[xys__][M[λ_, ω_]] := M[λ // hσ[xys], ω];
dσ[abs__][μ_] := μ // tσ[abs] // hσ[abs];

hη[x_][M[λ_, ω_]] := M[λ \ x, ω];

tm[u_, v_, w_][λ_List] := λ // LieMorphism[⟨u⟩ → ⟨w⟩, ⟨v⟩ → ⟨w⟩];
tm[u_, v_, w_][M[λ_, ω_]] := LieMorphism[⟨u⟩ → ⟨w⟩, ⟨v⟩ → ⟨w⟩] /@ M[λ, ω];
hm[x_, y_, z_][λ_List] := Union[λ \ {x, y}, {z → BCH[x /. λ, y /. λ]}];
hm[x_, y_, z_][M[λ_, ω_]] := M[λ // hm[x, y, z], ω];
tha[u_, x_][λ_List] := MapAt[RC[u, x /. λ], λ, {All, 2}];
tha[u_, x_][M[λ_, ω_]] :=
  M[λ // tha[u, x], (ω + J[u, x /. λ]) // RC[u, x /. λ]];
dm[a_, b_, c_][μ_] := μ // tha[⟨a⟩, b] // tm[⟨a⟩, ⟨b⟩, ⟨c⟩] // hm[a, b, c];
```

```
tΔ[u_, v_, w_][μ_M] := LieMorphism[⟨u⟩ → ⟨v⟩ + ⟨w⟩] /@ μ;
hΔ[x_, y_, z_][M[λ_, ω_]] := M[Union[λ \ x, {y → (x /. λ), z → (x /. λ)}], ω];
dΔ[a_, b_, c_][μ_M] := μ // tΔ[a, b, c] // hΔ[a, b, c];

tS[u_][μ_M] := LieMorphism[⟨u⟩ → -⟨u⟩] /@ μ;
hS[x_][M[λ_, ω_]] := M[Union[λ \ x, {x → -(x /. λ)}], ω];
dS[a_][μ_] := μ // tS[a] // hS[a] // tha[a, a] ;

tA[u_][expr_] := expr;
hA[x_][M[λ_, ω_]] := M[Union[λ \ x, {x → -(x /. λ)}], ω];
dA[a_][μ_] := μ // hA[a] // tha[a, a];

dc[a_][μ_] := μ // hS[a] // tha[a, a] // hS[a] // hη[a];

tϵ[u_] := M[{}, MakeCWSeries[0]];
hϵ[x_] := M[{x → MakeLieSeries[0]}, MakeCWSeries[0]];
dϵ[a_] := tϵ[a] ⋃ hϵ[a];
ρ⁺[u_, x_] := M[{x → MakeLieSeries[⟨u⟩]}, MakeCWSeries[0]];
ρ⁻[u_, x_] := M[{x → MakeLieSeries[-⟨u⟩]}, MakeCWSeries[0]];
R⁺[a_, b_] := ρ⁺[a, b] ⋃ M[{a → MakeLieSeries[0]}, MakeCWSeries[0]];
R⁻[a_, b_] := ρ⁻[a, b] ⋃ M[{a → MakeLieSeries[0]}, MakeCWSeries[0]];
R[a_, b_, p_] /; a ≠ b :=
  M[{a → MakeLieSeries[0], b → MakeLieSeries[p ⟨a⟩]}, MakeCWSeries[0]];
R[a_, a_, p_] := M[{a → MakeLieSeries[p ⟨a⟩]}, MakeCWSeries[0]];
Θ[a_, b_, p_] := (R[a, a, p / 2] // dΔ[a, a, b]) ** R[a, a, -p / 2] ** R[b, b, -p / 2];
Θ[a_, b_] := Θ[a, b, 1];

M /: μ1_M ** μ2_M := Module[{d1, d2, dom, μ1p = μ1, μ2p = μ2, s},
   {d1 = Domain[μ1], d2 = Domain[μ2], dom = Union[d1, d2]};
   Do[μ1p = μ1p ⋃ dϵ[a], {a, Complement[d2, d1]}];
   Do[μ2p = μ2p ⋃ dϵ[a], {a, Complement[d1, d2]}];
   s = Max[dom] - Min[dom] + 1;
   Do[μ2p = μ2p // dσ[a, a + s], {a, dom}];
   Fold[(#1 // dm[#2, #2 + s, #2]) &, μ1p ⋃ μ2p, dom]
  ];
```