Pensieve header: Experiments with RC, JA, and J; much borrowed from pensieve://Projects/KBH, and hopefully will be returned.

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\2013-02"];
```

We load FreeLie.m:

LoadFreeLie

```
<< FreeLie.m
$SeriesShowDegree = 3;
$SeriesCompareDegree = 6;
```

Some set theoretic definitions:

SetTheory

```
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]
     ));
(* LieDerivation[der_][f_List] := MapAt[LieDerivation[der], f, {All, 2}]; *)
LieMorphism[mor_][f_List] := MapAt[LieMorphism[mor], f, {All, 2}];
M /: M[λ1_, ω1_] ⋃ M[λ2_, ω2_] := M[λ1 ⋃ λ2, ω1 + ω2];
M[λ1_, ω1_] ≡ M[λ2_, ω2_] := (λ1 ≡ λ2) && (ω1 ≡ ω2);
```

MGADefs

```
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];
RC[u_, λx_LieSeries, ub_][ser_] := StableApply[
   LieMorphism[⟨u⟩ → Ad[λx][⟨ub⟩]],
   ser
  ];
RC[u_, λx_LieSeries][ser_] :=
  ser // RC[⟨u⟩, λx, ⟨"υ"⟩] // LieMorphism[⟨"υ"⟩ → ⟨u⟩];
J[u_, λx_] := Module[{s},
   IntegrateCWSeries[
    div[⟨u⟩, λx // RC[⟨u⟩, s λx]] // LieMorphism[⟨u⟩ → Ad[-s λx][⟨u⟩]],
    {s, 0, 1}
   ]
  ];
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];
R^+[u_, x_] := M[{x → MakeLieSeries[⟨u⟩], u → MakeLieSeries[0]}, MakeCWSeries[0]];
R^-[u_, x_] := M[{x → MakeLieSeries[-⟨u⟩], u → MakeLieSeries[0]}, MakeCWSeries[0]];
```