

Free Lie Algebras Routines

Lazy Evaluation Version

Pensieve header: A free-Lie calculator with lazy evaluation for series; forks pensieve://2013-04/.

Global Definitions

```
$SeriesShowDegree = 3; $SeriesCompareDegree = 3;
```

Words and Lyndon Words

A Lyndon word is a word lexicographically smaller than all of its proper right factors; see <http://katlas-math.toronto.edu/drorbn/AcademicPensieve/Projects/FreeLie/index.html>

```

LyndonQ[AW[w_String]] := And @@ (
  OrderedQ[{w, #}] & /@ Table[StringDrop[w, i], {i, 1, StringLength[w] - 1}]
);
AllWords[0, _List] = {AW[""]};
AllWords[n_ /; n > 0, ab_List] := AllWords[n, ab] = AW /@ Flatten[Outer[
  StringJoin[#1, #2] &,
  First /@ AllWords[n - 1, ab],
  ab
]];
AllLyndonWords[n_Integer, ab_List] :=
  LW @@@ Select[AllWords[n, ab /. LW[w_String] => w], LyndonQ];
AllLyndonWords[{n_}, ab_List] :=
  Join@@Table[AllLyndonWords[k, ab /. LW[w_String] => w], {k, n}];
LyndonFactorization[LW[w_String] /; StringLength[w] == 1] := LW[w];
LyndonFactorization[LW[w_String] /; StringLength[w] > 1] := Module[
  {rf},
  rf = First[Sort[Table[StringDrop[w, i], {i, 1, StringLength[w] - 1}]]];
  LW /@ {StringDrop[w, -StringLength[rf]], rf}
];
LW[s_Symbol] := LW[ToString[s]];
LW[LW[w_]] := LW[w];
LW /: LW[x_] ≤ LW[y_] := OrderedQ[{x, y}];
LW /: x_LW ≥ y_LW := y ≤ x;
LW /: x_LW > y_LW := !(x ≤ y);
LW /: x_LW < y_LW := !(y ≤ x);
Format[LW[w_], StandardForm] := Defer[⟨w⟩];
BracketForm[w_LW] /; Deg[w] == 1 := w[[1]];
BracketForm[w_LW] := BracketForm[w] = StringJoin[Flatten[{
  "[",
  BracketForm /@ LyndonFactorization[w],
  "]"
}]];
topbracketform[w_LW] /; Deg[w] == 1 := w[[1]];
topbracketform[w_LW] := topbracketform[w] = Overscript[
  Row[Riffle[topbracketform /@ LyndonFactorization[w], " "]],
  ─
];
TopBracketForm[w_LW] /; Deg[w] == 1 := Overscript[w[[1]], ─];
TopBracketForm[w_LW] := topbracketform[w];
TopBracketForm[CW[w_String]] := Overscript[w, ─];
TopBracketForm[expr_] := expr /. w_LW | w_CW => TopBracketForm[w];
⟨w_⟩ := LW[w];
LW[is_Integer] := LW[StringJoin@@
  (StringTake["1234567890abcdefghijklmnopqrstuvwxyz", {#}] & /@ {is})];
Deg[LW[x_]] := StringLength[x];

{LyndonQ[AW["abba"], LyndonQ[AW["ababb"]]}
{False, True}

```

```

{AllWords[3, {"1", "2"}], AllLyndonWords[{3}, {"1", "2"}]}
{{AW[111], AW[112], AW[121], AW[122], AW[211], AW[212], AW[221], AW[222]},
 {<1>, <2>, <12>, <112>, <122>}}

Table[Length[AllLyndonWords[k, {"1", "2"}]], {k, 10}]
{2, 1, 2, 3, 6, 9, 18, 30, 56, 99}

Table[Length[AllLyndonWords[k, {"1", "2", "3"}]], {k, 10}]
{3, 3, 8, 18, 48, 116, 312, 810, 2184, 5880}

BracketForm[LW["12122"]]
[[12][[12]2]]

```

The Bracket for Lie Elements

```

b[0, _] = 0; b[_ , 0] = 0;
b[c_ * (x_AW | x_LW), y_] := Expand[c b[x, y]];
b[x_, c_ * (y_AW | y_LW)] := Expand[c b[x, y]];
b[x_Plus, y_] := b[#, y] & /@ x;
b[x_, y_Plus] := b[x, #] & /@ y;
b[w_LW, z_LW] := LWAdjoint[w][z];
ad[x_][y_] := b[x, y];

LWAdjoint[w_] := LWAdjoint[w] = Module[{u},
  u = Unique[LWAct];
  u[z_] := u[z] = Which[
    w === z, 0,
    z < w, Expand[-b[z, w]],
    Deg[w] == 1, LW[First[w] <> First[z]],
    True, Module[{x, y},
      {x, y} = LyndonFactorization[w];
      If[y ≥ z,
        LW[First[w] <> First[z]],
        b[x, LWAdjoint[y][z]] + b[LWAdjoint[x][z], y]
      ]
    ]
  ];
  u
];

b[LW["112"], LW["122"]]
<112122> + <112212>

```

```

Outer[b, AllLyndonWords[{3}, {"1", "2"}],
  AllLyndonWords[{3}, {"1", "2"}]] // MatrixForm

$$\begin{pmatrix} 0 & \langle 12 \rangle & \langle 112 \rangle & \langle 1112 \rangle & \langle 1122 \rangle \\ -\langle 12 \rangle & 0 & -\langle 122 \rangle & -\langle 1122 \rangle & -\langle 1222 \rangle \\ -\langle 112 \rangle & \langle 122 \rangle & 0 & -\langle 11212 \rangle & \langle 12122 \rangle \\ -\langle 1112 \rangle & \langle 1122 \rangle & \langle 11212 \rangle & 0 & \langle 112122 \rangle + \langle 112212 \rangle \\ -\langle 1122 \rangle & \langle 1222 \rangle & -\langle 12122 \rangle & -\langle 112122 \rangle & -\langle 112212 \rangle \end{pmatrix}$$

Union[Flatten[Outer[(b[#1, #2] + b[#2, #1]) &,
  AllLyndonWords[{6}, {"1", "2"}], AllLyndonWords[{6}, {"1", "2"}]
  ]]]
{0}

Outer[(b[#1, b[#2, #3]] + b[#2, b[#3, #1]] + b[#3, b[#1, #2]]) &,
  AllLyndonWords[{5}, {"1", "2"}],
  AllLyndonWords[{5}, {"1", "2"}], AllLyndonWords[{5}, {"1", "2"}]
] // Flatten // Union
{0}

```

LieSeries

```

LieSeries[ser_Symbol][{dd_Integer}] :=
  TopBracketForm[LS@@Table[ser[d], {d, dd}]];
LieSeries[ser_Symbol][e___] := ser[e];
Format[s_LieSeries, StandardForm] := s[{$SeriesShowDegree}];
ShowLieSeries[d_Integer][s_LieSeries] := s[{d}];
MakeLieSeries[s_LieSeries] := s;
MakeLieSeries[expr_] :=
  MakeLieSeries[expr] = MakeLieSeries[Unique[MakeLieSeries], expr];
MakeLieSeries[ser_Symbol, expr_] := (
  ser[] = Hold[MakeLieSeries[ser, expr]];
  ser[d_Integer] := ser[d] = Expand[expr /. w_LW /. Deg[w] ≠ d → 0];
  LieSeries[ser]
);
MakeLieSeries[ab_List, coefs_] :=
  MakeLieSeries[Unique[MakeLieSeries], ab, coefs];
MakeLieSeries[ser_Symbol, ab_List, coefs_Symbol] := (
  ser[] = Hold[MakeLieSeries[a, coefs]];
  ser[d_Integer] := ser[d] =
    Plus @@ MapIndexed[({coefs@@Prepend[#2, d]} * #1) &, AllLyndonWords[d, ab]];
  LieSeries[ser]
);
s1_LieSeries == s2_LieSeries := Module[{res = True, k},
  For[k = 1, res && k <= $SeriesCompareDegree, ++k, res = res && (s1[k] == s2[k])];
  res
];
RandomLieSeries[ab_List, opts__Rule] :=
  RandomLieSeries[Unique[RandomLieSeries], ab, opts];
RandomLieSeries[ser_Symbol, ab_List, opts__Rule] := Module[
  {rand = Randomizer /. {opts} /.
    Randomizer → (
      
$$\frac{\text{RandomInteger}[-2 \text{Deg}[\#]!, 2 \text{Deg}[\#]!]}{\text{Deg}[\#]!} \&$$

    )},
  ser[] = Hold[RandomLieSeries[a, opts]];
  ser[d_Integer] := ser[d] = Plus @@ ((rand[#] * #) & /@ AllLyndonWords[d, ab]);
  LieSeries[ser]
];
Print /@ {ts1 = <"1122"> // MakeLieSeries, ts1[], ts1 /@ Range[6]};
LS[0, 0, 0]
Hold[MakeLieSeries[MakeLieSeries$5040, <1122>]]
{0, 0, 0, <1122>, 0, 0}
α = RandomLieSeries[{"a", "b"}]
LS[-2 a + 2 b,  $\frac{\overline{ab}}{2}$ ,  $\frac{3}{2} \overline{aab} - \frac{7}{6} \overline{abb}$ ]

```

$\alpha\{5\}$

$$\text{LS} \left[-2 \langle a \rangle + 2 \langle b \rangle, \frac{\langle ab \rangle}{2}, \frac{3 \langle aab \rangle}{2} - \frac{7 \langle abb \rangle}{6}, \frac{11 \langle aaab \rangle}{8} + \frac{23 \langle aabb \rangle}{24} + \frac{19 \langle abbb \rangle}{12}, \right. \\ \left. - \frac{7 \langle aaaab \rangle}{15} + \frac{13 \langle aaabb \rangle}{120} + \frac{181 \langle aabab \rangle}{120} - \frac{163 \langle aabbb \rangle}{120} - \frac{7 \langle ababb \rangle}{60} + \frac{5 \langle abbbb \rangle}{24} \right]$$

```

AddLieSeries[ss__LieSeries] := AddLieSeries[ss] = Module[{ser},
  ser = Unique[AddLieSeries];
  ser[] = Hold[AddLieSeries[ss]];
  ser[d_Integer] := ser[d] = Plus @@ ((#[d]) & /@ {ss});
  LieSeries[ser]
];

LieSeries /: Plus[ss__LieSeries] := AddLieSeries[ss];
ScaleLieSeries[c_, s_LieSeries] := ScaleLieSeries[c, s] = Module[{ser},
  ser = Unique[ScaleLieSeries];
  ser[] = Hold[ScaleLieSeries[c, s]];
  ser[d_Integer] := ser[d] = Expand[c*s[d]];
  LieSeries[ser]
];

LieSeries /: c_*s_LieSeries := ScaleLieSeries[c, s];
b[s_LieSeries, y_] := b[s, MakeLieSeries[y]];
b[x_, s_LieSeries] := b[MakeLieSeries[x], s];

b[s1_LieSeries, s2_LieSeries] := b[s1, s2] = Module[{ser},
  ser = Unique[b];
  ser[] = Hold[b[s1, s2]];
  ser[d_Integer] := ser[d] = Sum[
    b[s1[k], s2[d-k]],
    {k, 1, d-1}
  ];
  LieSeries[ser]
];

b[s_LieSeries, y_] := b[s, MakeLieSeries[y]];
b[x_, s_LieSeries] := b[MakeLieSeries[x], s];

{ts2 = <"122"> + <"11122"> // MakeLieSeries,
  ts3 = b[ts1, ts2], ts3[], ts3 /@ Range[10]}

{LS[0, 0, <122>], LS[0, 0, 0], Hold[b[LS[0, 0, 0], LS[0, 0, <122>]]],
  {0, 0, 0, 0, 0, 0, <1122122>, 0, -<111221122>, 0}}

LieSeries /: EulerE[s_LieSeries] := Module[{ser},
  ser = Unique[EulerE];
  ser[] = Hold[EulerE[s]];
  ser[d_Integer] := ser[d] = Expand[d*s[d]];
  LieSeries[ser]
];

```

```
{ts4 = EulerE[ts3], ts4[], ts4 /@ Range[10]}
{LS[0, 0, 0], Hold[EulerE[LS[0, 0, 0]]],
 {0, 0, 0, 0, 0, 0, 7 <1122122>, 0, -9 <111221122>, 0}}
```

adPower, adSeries, and Ad

```
adPower[0, x_LieSeries][ψ_LieSeries] := adPower[0, x][ψ] = Module[{ser},
  ser = Unique[adPower];
  ser[] = Hold[adPower[0, x][ψ]];
  ser[d_Integer] := ser[d] = ψ[d];
  LieSeries[ser]
];
adPower[n_Integer, x_LieSeries][ψ_LieSeries] := adPower[n, x][ψ] = Module[{ser},
  ser = Unique[adPower];
  ser[] = Hold[adPower[n, x][ψ]];
  ser[d_Integer] := ser[d] = b[x, adPower[n-1, x][ψ]][d];
  LieSeries[ser]
];
adSeries[f_, x_LieSeries][ψ_LieSeries] := adSeries[f, x][ψ] = Module[{ser},
  ser = Unique[adSeries];
  ser[] = Hold[adSeries[f, x][ψ]];
  ser[d_Integer] := ser[d] = Module[{c},
    Expand[Sum[
      c = SeriesCoefficient[f, {ad, 0, k}];
      If[c === 0, 0, c * adPower[k, x][ψ][d]],
      {k, 0, d-1}
    ]]
  ];
  LieSeries[ser]
];
adSeries[f_, x_][ψ_] := adSeries[f, MakeLieSeries[x]][MakeLieSeries[ψ]];
Ad[x_] := adSeries[E^ad, x];

{xs = MakeLieSeries[LW["x"]], ys = MakeLieSeries[LW["y"]],
 ts5 = adPower[0, xs][ys], ts5[], ts5 /@ Range[5]}
{LS[<x>, 0, 0], LS[<y>, 0, 0], LS[<y>, 0, 0],
 Hold[adPower[0, LS[<x>, 0, 0]][LS[<y>, 0, 0]]], {<y>, 0, 0, 0, 0}}
```

adPower[3, xs][ys] /@ Range[5]

```
{0, 0, 0, <xxxxy>, 0}
```

{adSeries[E^(-ad), xs][ys] /@ Range[5], adSeries[E^(-ad), ys][xs] /@ Range[5]}

$$\left\{ \left\{ \langle y \rangle, -\langle xy \rangle, \frac{\langle xxy \rangle}{2}, -\frac{\langle xxxxy \rangle}{6}, \frac{\langle xxxxy \rangle}{24} \right\}, \left\{ \langle x \rangle, \langle xy \rangle, \frac{\langle xyy \rangle}{2}, \frac{\langle xyyy \rangle}{6}, \frac{\langle xyyyy \rangle}{24} \right\} \right\}$$

Ad[xs][ys][5]

$$\frac{\langle xxxxy \rangle}{24}$$

```
Ad[xs][ys][]
Hold[adSeries[e-ad, LS[⟨x⟩, 0, 0]][LS[⟨y⟩, 0, 0]]]
```

LieDerivation, DerivationPower, DerivationSeries

```
LieDerivation[der_][es___] := der[es];
LieDerivation[rules__Rule] := LieDerivation[{rules}];
LieDerivation[rules_List] :=
  LieDerivation[rules] = LieDerivation[Unique[LieDerivation], rules];
LieDerivation[der_Symbol, rules_List] := (
  der[] = Hold[LieDerivation[der, rules]];
  (der[w_LW] /; Deg[w] == 1) :=
    (der[w] = MakeLieSeries[w /. Append[rules, _LW → 0]]);
  der[w_LW] := der[w] = Module[{x, y},
    {x, y} = LyndonFactorization[w];
    AddLieSeries[b[der[x], y], b[x, der[y]]]
  ];
  der[s_LieSeries] := der[s] = Module[{ser},
    ser = Unique[LieDerivationOnLieSeries];
    ser[] = Hold[der[s]];
    ser[d_] := ser[d] = Sum[
      der[s[k]][d],
      {k, 1, d}
    ];
    LieSeries[ser]
  ];
  der[as_ASeries] := der[as] = Module[{ser},
    ser = Unique[LieDerivationOnASeries];
    ser[] = Hold[der[as]];
    ser[d_] := ser[d] = Sum[
      Expand[as[k] /. AW[w_] => Sum[
        NonCommutativeMultiply[
          AW[StringTake[w, j - 1]],
          ⌊[der[LW[StringTake[w, {j}]]][d - k + 1]],
          AW[StringDrop[w, j]]
        ],
        {j, k}
      ]],
      {k, 1, d}
    ];
    ASeries[ser]
  ];
  der[cws_CWSeries] := der[cws] = Module[{ser},
    ser = Unique[LieDerivationOnCWSeries];
    ser[] = Hold[der[cws]];
    ser[d_] := ser[d] = Sum[
      Expand[cws[k] /. CW[w_] => Sum[
        tr[NonCommutativeMultiply[
          AW[StringTake[w, j - 1]],
```



```

      ⌊[der[LW[StringTake[w, {j}]]][d - k + 1]],
      AW[StringDrop[w, j]]
    ]],
    {j, k}
  ]],
  {k, 1, d}
];
CWSeries[ser]
];
der[expr_][d_] :=
  Expand[expr /. {w_LW => der[w][d], s_LieSeries => der[s][d]}];
LieDerivation[der]
);

Print /@ {
  ld1 = LieDerivation[{{<1> → b[<3>, <1>]}],
  ld1[],
  (# → ld1[#][{4}]) & /@ AllLyndonWords[{3}, {"1", "2"}],
  (<"112"> // ld1 // ld1)[{5}]
};

LieDerivation[LieDerivation$5120]

Hold[LieDerivation[LieDerivation$5120, {{<1> → -<13>}}]

{{<1> → LS[0, -<13>, 0, 0], <2> → LS[0, 0, 0, 0], <12> → LS[0, 0, -<132>, 0],
  <112> → LS[0, 0, 0, -<1132> + <1213>], <122> → LS[0, 0, 0, -<1322>]}
LS[0, 0, 0, 0, <11332> - <12133> + 2 <13132>]

```

```

DerivationPower[0, der_LieDerivation][ψ_LieSeries] :=
  DerivationPower[0, der][ψ] = Module[{ser},
    ser = Unique[DerivationPower];
    ser[] = Hold[DerivationPower[0, der][ψ]];
    ser[d_Integer] := ser[d] = ψ[d];
    LieSeries[ser]
  ];
DerivationPower[n_Integer, der_LieDerivation][ψ_LieSeries] :=
  DerivationPower[n, x][ψ] = Module[{ser},
    ser = Unique[DerivationPower];
    ser[] = Hold[DerivationPower[n, der][ψ]];
    ser[d_Integer] := ser[d] = der[DerivationPower[n-1, der][ψ]][d];
    LieSeries[ser]
  ];
DerivationSeries[___][0] = 0;
DerivationSeries[f_, ld_LieDerivation][ψ_LieSeries] :=
  DerivationSeries[f, ld][ψ] = Module[{ser},
    ser = Unique[DerivationSeries];
    ser[] = Hold[DerivationSeries[f, ld][ψ]];
    ser[d_Integer] := ser[d] = Module[{c},
      Expand[Sum[
        c = SeriesCoefficient[f, {der, 0, k}];
        If[c == 0, 0, c * DerivationPower[k, ld][ψ][d]],
        {k, 0, d}
      ]]
    ];
    LieSeries[ser]
  ];
DerivationExp[ld_LieDerivation] := DerivationSeries[E^der, ld];

<"112"> // MakeLieSeries // DerivationExp[LieDerivation[{{<1> → b[<3>, <1>}}]] //
  ShowLieSeries[6]
LS[0, 0, <112>, -<1132> + <1213>,  $\frac{\langle 11332 \rangle}{2} - \frac{\langle 12133 \rangle}{2} + \langle 13132 \rangle,$ 
 $-\frac{\langle 113332 \rangle}{6} + \frac{\langle 121333 \rangle}{6} - \frac{\langle 131332 \rangle}{2} + \frac{\langle 132133 \rangle}{2}$ ]

<"122"> // MakeLieSeries // DerivationExp[LieDerivation[{{<1> → b[<3>, <1>}}]] //
  ShowLieSeries[6]
LS[0, 0, <122>, -<1322>,  $\frac{\langle 13322 \rangle}{2}, -\frac{\langle 133322 \rangle}{6}$ ]

```

LieMorphism

```

LieMorphism[mor_][es___] := mor[es];
LieMorphism[rules_List] :=
  LieMorphism[Unique[LieMorphism], rules];
LieMorphism[rules_Rule] := LieMorphism[{rules}];
LieMorphism[mor_Symbol, rules_List] := (
  mor[] = Hold[LieMorphism[mor, rules]];
  (mor[w_LW] /; Deg[w] == 1) := (mor[w] = MakeLieSeries[w /. rules]);
  mor[w_LW] := (mor[w] = b @@ (mor /@ LyndonFactorization[w]));
  mor[AW[""]] = MakeASeries[AW[""]];
  (mor[AW[w_]] /; StringLength[w] == 1) :=
    (mor[w] =  $\iota$ [MakeLieSeries[LW[w] /. rules]));
  mor[AW[w_]] := mor[w] = Module[{w1, w2},
    w1 = StringTake[w, Floor[StringLength[w] / 2]];
    w2 = StringDrop[w, Floor[StringLength[w] / 2]];
    (mor[AW[w1]]) ** (mor[AW[w2]])
  ];
  mor[CW[w_]] := tr[mor[AW[w]]];
  mor[s_LieSeries] := mor[s] = Module[{ser},
    ser = Unique[LieMorphismOnLieSeries];
    ser[] = Hold[mor[s]];
    ser[d_] := ser[d] = Sum[
      mor[s[k]][d],
      {k, 1, d}
    ];
    LieSeries[ser]
  ];
  mor[cws_CWSeries] := mor[cws] = Module[{ser},
    ser = Unique[LieMorphismOnCWSeries];
    ser[] = Hold[mor[s]];
    ser[d_] := ser[d] = Sum[
      mor[cws[k]][d],
      {k, 1, d}
    ];
    CWSeries[ser]
  ];
  mor[expr_][d_] := Expand[expr /. (w_LW | w_AW | w_CW)  $\Rightarrow$  mor[w][d]];
  LieMorphism[mor]
);

Print /@ {
  lm0 = LieMorphism[{LW["x"]  $\rightarrow$  LW["y"]}],
  LW["x"] // lm0,
  AW["x"] // lm0,
  CW["x"] // lm0};

```

```

LieMorphism[LieMorphism$8978]
LS[⟨y⟩, 0, 0]
ℓ[LS[⟨y⟩, 0, 0]]
tr[ℓ[LS[⟨y⟩, 0, 0]]]

Print /@ {
  lm1 = LieMorphism[{LW["x"] → Ad[LW["y"]][LW["x"]]}],
  lm1[],
  lm1[LW["y"]],
  lm1[LW["x"]],
  lm1[LW["x"]][4],
  lm1[⟨"xy"⟩],
  lm1[⟨"xy"⟩][8],
  lm1[AW["x"]],
  lm1[CW["x"]]
};

LieMorphism[LieMorphism$8979]

Hold[LieMorphism[LieMorphism$8979, {⟨x⟩ → LS[⟨x⟩, ⟨xy⟩,  $\frac{\langle xyY \rangle}{2}$ ]}]]]
LS[⟨y⟩, 0, 0]
LS[⟨x⟩, ⟨xy⟩,  $\frac{\langle xyY \rangle}{2}$ ]
 $\frac{\langle xyYY \rangle}{6}$ 
LS[0, 0, ⟨xy⟩]
 $\frac{\langle xxyYYYY \rangle}{120} + \frac{\langle xyxyYYYY \rangle}{30} + \frac{\langle xyYxyYYY \rangle}{24}$ 
ℓ[LS[⟨x⟩, ⟨xy⟩,  $\frac{\langle xyY \rangle}{2}$ ]]]
tr[ℓ[LS[⟨x⟩, ⟨xy⟩,  $\frac{\langle xyY \rangle}{2}$ ]]]]]

```

StableApply

```

StableApply[mor_LieMorphism, (type : (LieSeries | ASeries | CWSeries))][s_] := (
  StableApply[mor, type[s]] = Module[{ser},
    ser = Unique[StableApply];
    ser[] = Hold[StableApply[mor, type[s]]];
    ser[d_] := ser[d] = Nest[mor, type[s], d][d];
    type[ser]
  ]
);

```

BCH

```

BCHBase = Module[{bch},
  bch = Unique["BCHBase"];
  bch[] = Hold[BCHBase];
  bch[1] = <"x"> + <"y">;
  bch[d_Integer] := bch[d] = Expand[Plus[
    adSeries[E^(-ad), MakeLieSeries[<"y">]][MakeLieSeries[<"x">]][d],
    -adSeries[(1 - E^(-ad)) / ad - 1, LieSeries[bch]][
      EulerE[LieSeries[bch]][d]
    ] / d];
  LieSeries[bch]
];
BCH[x_, y_] := LieMorphism[{LW["x"] -> x, LW["y"] -> y}][BCHBase];

```



```
{BCH[LW["Y"], LW["z"]], BCH[LW["Y"], LW["z"]][6]}
{LS[⟨y⟩ + ⟨z⟩,  $\frac{\langle yz \rangle}{2}, \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ],
-  $\frac{\langle YYYZZZ \rangle}{1440} + \frac{\langle YYYzyz \rangle}{720} + \frac{\langle YYYzzz \rangle}{360} + \frac{\langle YYzyzz \rangle}{240} - \frac{\langle YYzzzz \rangle}{1440}$ }]
{LS[⟨y⟩ + ⟨z⟩,  $\frac{\langle yz \rangle}{2}, \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ],
-  $\frac{\langle YYYZZZ \rangle}{1440} + \frac{\langle YYYzyz \rangle}{720} + \frac{\langle YYYzzz \rangle}{360} + \frac{\langle YYzyzz \rangle}{240} - \frac{\langle YYzzzz \rangle}{1440}$ }]
{LS[⟨y⟩ + ⟨z⟩,  $\frac{\langle yz \rangle}{2}, \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ],
-  $\frac{\langle YYYZZZ \rangle}{1440} + \frac{\langle YYYzyz \rangle}{720} + \frac{\langle YYYzzz \rangle}{360} + \frac{\langle YYzyzz \rangle}{240} - \frac{\langle YYzzzz \rangle}{1440}$ }]
{LieSeries[LieMorphismOnLieSeries$101],
-  $\frac{\langle YYYZZZ \rangle}{1440} + \frac{\langle YYYzyz \rangle}{720} + \frac{\langle YYYzzz \rangle}{360} + \frac{\langle YYzyzz \rangle}{240} - \frac{\langle YYzzzz \rangle}{1440}$ }]
```

```
{
  t1 = BCH[LW["x"], BCH[LW["Y"], LW["z"]]],
  t2 = BCH[BCH[LW["x"], LW["Y"]], LW["z"]],
  t1 == t2,
  Table[t1[d] == t2[d], {d, 10}]
} // Timing
```

```
{4.056, {LS[⟨x⟩ + ⟨y⟩ + ⟨z⟩,  $\frac{\langle xy \rangle}{2} + \frac{\langle xz \rangle}{2} + \frac{\langle yz \rangle}{2}$ ,
 $\frac{\langle xxy \rangle}{12} + \frac{\langle xxz \rangle}{12} + \frac{\langle xyy \rangle}{12} + \frac{\langle xyz \rangle}{3} + \frac{\langle xzy \rangle}{6} + \frac{\langle xzz \rangle}{12} + \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ], LS[⟨x⟩ + ⟨y⟩ + ⟨z⟩,
 $\frac{\langle xy \rangle}{2} + \frac{\langle xz \rangle}{2} + \frac{\langle yz \rangle}{2}$ ,  $\frac{\langle xxy \rangle}{12} + \frac{\langle xxz \rangle}{12} + \frac{\langle xyy \rangle}{12} + \frac{\langle xyz \rangle}{3} + \frac{\langle xzy \rangle}{6} + \frac{\langle xzz \rangle}{12} + \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ],
LS[⟨x⟩ + ⟨y⟩ + ⟨z⟩,  $\frac{\langle xy \rangle}{2} + \frac{\langle xz \rangle}{2} + \frac{\langle yz \rangle}{2}$ ,
 $\frac{\langle xxy \rangle}{12} + \frac{\langle xxz \rangle}{12} + \frac{\langle xyy \rangle}{12} + \frac{\langle xyz \rangle}{3} + \frac{\langle xzy \rangle}{6} + \frac{\langle xzz \rangle}{12} + \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ] == LS[⟨x⟩ + ⟨y⟩ + ⟨z⟩,
 $\frac{\langle xy \rangle}{2} + \frac{\langle xz \rangle}{2} + \frac{\langle yz \rangle}{2}$ ,  $\frac{\langle xxy \rangle}{12} + \frac{\langle xxz \rangle}{12} + \frac{\langle xyy \rangle}{12} + \frac{\langle xyz \rangle}{3} + \frac{\langle xzy \rangle}{6} + \frac{\langle xzz \rangle}{12} + \frac{\langle YYz \rangle}{12} + \frac{\langle Yzz \rangle}{12}$ ],
{True, True, True, True, True, True, True, True, True, True, True}}}
```

AW, ASeries, ι , σ

```

Unprotect[NonCommutativeMultiply];
x_**0 = 0; 0**y_ = 0;
(c_*x_AW)**y_ := Expand[c(x**y)];
x_** (c_*y_AW) := Expand[c(x**y)];
x_Plus**y_ := (#**y) & /@ x;
x_**y_Plus := (x**#) & /@ y;
Deg[AW[w_]] := StringLength[w];
AW[AW[w_]] := AW[w];
AW[w1_String]**AW[w2_String] := AW[w1<>w2];
b[w_AW, z_AW] := w**z - z**w;

ASeries[ser_Symbol][{dd_Integer}] := AS@@Table[ser[d], {d, 0, dd}];
ASeries[as_Symbol][es___] := as[es];
Format[s_ASeries, StandardForm] := s[{$SeriesShowDegree}];
MakeASeries[as_CWSeries] := as;
MakeASeries[expr_] :=
  MakeASeries[expr] = MakeCWSeries[Unique[MakeASeries], expr];
MakeASeries[ser_Symbol, expr_] := (
  ser[] = Hold[MakeASeries[ser, expr]];
  ser[d_Integer] := ser[d] = Expand[expr /. w_AW /; Deg[w] ≠ d → 0];
  ASeries[ser]
);
(s1_ASeries**s2_ASeries) := (s1**s2) = Module[{ser},
  ser = Unique[NonCommutativeMultiply];
  ser[] = Hold[s1**s2];
  ser[d_Integer] := ser[d] = Sum[
    s1[k]**s2[d-k],
    {k, 0, d}
  ];
  ASeries[ser]
];

ι[w_LW] /; Deg[w] == 1 := AW@@w;
ι[w_LW] := ι[w] = b@@(ι /@ LyndonFactorization[w]);
ι[expr_] := Expand[expr /. w_LW => ι[w]];
ι[ls_LieSeries] := ι[ls] = Module[{as},
  as = Unique[ι];
  as[] = Hold[ι[ls]];
  as[d_] := as[d] = ι[ls[d]];
  ASeries[as]
];

ι[BCHBase[3]]
AW[xyy] / 12 - AW[xyx] / 6 + AW[xyy] / 12 + AW[yxx] / 12 - AW[yxy] / 6 + AW[yyx] / 12

```


{as = \mathcal{L} [BCHBase], as[5]}

Power::infy : Infinite expression $\frac{1}{0}$ encountered. >>

Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. >>

$$\left\{ \text{AS} \left[\text{Indeterminate}, \text{AW}[x] + \text{AW}[y], \frac{\text{AW}[xy]}{2} - \frac{\text{AW}[yx]}{2}, \right. \right. \\ \left. \frac{\text{AW}[xxy]}{12} - \frac{\text{AW}[xyx]}{6} + \frac{\text{AW}[xyy]}{12} + \frac{\text{AW}[yxx]}{12} - \frac{\text{AW}[yxy]}{6} + \frac{\text{AW}[yyx]}{12} \right], \\ - \frac{\text{AW}[xxxxy]}{720} + \frac{\text{AW}[xxxxyx]}{180} + \frac{\text{AW}[xxxxyy]}{180} - \frac{\text{AW}[xxyxx]}{120} - \frac{\text{AW}[xxyxy]}{120} - \frac{\text{AW}[xxyyx]}{120} + \\ \frac{\text{AW}[xxyyy]}{180} + \frac{\text{AW}[xyxxx]}{180} - \frac{\text{AW}[xyxyy]}{120} + \frac{\text{AW}[xyxyx]}{30} - \frac{\text{AW}[xyxyy]}{120} - \frac{\text{AW}[xyyxx]}{120} - \\ \frac{\text{AW}[xyyxy]}{120} + \frac{\text{AW}[xyyyx]}{180} - \frac{\text{AW}[xyyyy]}{720} - \frac{\text{AW}[yxxxx]}{720} + \frac{\text{AW}[yxxxxy]}{180} - \frac{\text{AW}[yxxxxy]}{120} - \\ \frac{\text{AW}[yxxxy]}{120} - \frac{\text{AW}[yxyxx]}{120} + \frac{\text{AW}[yxyxy]}{30} - \frac{\text{AW}[yxyyx]}{120} + \frac{\text{AW}[yxyyy]}{180} + \frac{\text{AW}[yyxxx]}{180} - \\ \frac{\text{AW}[yyxxy]}{120} - \frac{\text{AW}[yyxyx]}{120} - \frac{\text{AW}[yyxyy]}{120} + \frac{\text{AW}[yyyxx]}{180} + \frac{\text{AW}[yyyxy]}{180} - \frac{\text{AW}[yyyyx]}{720} \left. \right\}$$

```

σ[y_LW, w_LW] /; Deg[y] == 1 := σ[y, w] = Which[
  y === w, AW[""],
  Deg[w] === 1, 0,
  True, Module[{w1, w2},
    {w1, w2} = LyndonFactorization[w];
     $\mathcal{L}[w1] ** \sigma[y, w2] - \mathcal{L}[w2] ** \sigma[y, w1]$ 
  ]
];
σ[y_, ls_LieSeries] := σ[y, ls] = Module[{as},
  as = Unique[σ];
  as[] = Hold[σ[y, ls]];
  as[d_] := as[d] = σ[LW[y], ls[d+1]];
  ASeries[as]
];
σ[y_, expr_] := Expand[expr /. w_LW => σ[LW[y], w]];

```

```

(# -> σ[1, #]) & /@ AllLyndonWords[{5}, {"1", "2"}]
{<1> -> AW[], <2> -> 0, <12> -> -AW[2], <112> -> -2 AW[12] + AW[21], <122> -> AW[22],
<1112> -> -3 AW[112] + 3 AW[121] - AW[211], <1122> -> 2 AW[212] - AW[221],
<1222> -> -AW[222], <11112> -> -4 AW[1112] + 6 AW[1121] - 4 AW[1211] + AW[2111],
<11122> -> -AW[1122] + 4 AW[1212] - AW[1221] - 2 AW[2121] + AW[2211],
<11212> -> -AW[1122] + 4 AW[1212] - AW[1221] - 3 AW[2112] + AW[2121],
<11222> -> -2 AW[1222] + 3 AW[2122] - 3 AW[2212] + AW[2221],
<12122> -> 2 AW[1222] - 3 AW[2122] + AW[2212], <12222> -> AW[2222]}

```

$$\{\sigma["x", \text{BCHBase}][5], \sigma["y", \text{BCHBase}][5]\}$$

$$\left\{ \begin{array}{l} -\frac{\text{AW}[yxxxxy]}{360} + \frac{\text{AW}[yxyxyx]}{240} + \frac{\text{AW}[yxxyyy]}{240} - \frac{\text{AW}[yxyxxx]}{360} - \frac{\text{AW}[yxxyxy]}{60} + \frac{\text{AW}[yxyyx]}{240} - \frac{\text{AW}[xyyyy]}{360} + \\ \frac{\text{AW}[yyxxx]}{1440} + \frac{\text{AW}[yyxxy]}{240} + \frac{\text{AW}[yyxyx]}{240} + \frac{\text{AW}[yyxyy]}{240} - \frac{\text{AW}[yyyxx]}{360} - \frac{\text{AW}[yyyxy]}{360} + \frac{\text{AW}[yyyxyx]}{1440}, \\ -\frac{\text{AW}[xxxxy]}{1440} + \frac{\text{AW}[xxxxyx]}{360} + \frac{\text{AW}[xxxxyy]}{360} - \frac{\text{AW}[xxyxxx]}{240} - \frac{\text{AW}[xxyxy]}{240} - \frac{\text{AW}[xxyyx]}{240} - \frac{\text{AW}[xxyyy]}{1440} + \\ \frac{\text{AW}[xyxxx]}{360} - \frac{\text{AW}[xyxxy]}{240} + \frac{\text{AW}[xyxyx]}{60} + \frac{\text{AW}[xyxyy]}{360} - \frac{\text{AW}[xyyxx]}{240} - \frac{\text{AW}[xyyxy]}{240} + \frac{\text{AW}[xyyyx]}{360} \end{array} \right\}$$

CW, CWSeries, tr, div

```

Deg[CW[w_]] := StringLength[w];
AllCyclicWords[d_Integer, ab_List] :=
  AllCyclicWords[d, ab] = Union[tr[AW[StringJoin@#] & /@ Tuples[ab, d]]];
CWSeries[cws_Symbol][es_] := cws[es];
CWSeries[ser_Symbol][{dd_Integer}] :=
  TopBracketForm[CWS @@ Table[ser[d], {d, dd}]];
Format[s_CWSeries, StandardForm] := s[{$SeriesShowDegree}];
MakeCWSeries[cws_CWSeries] := cws;
MakeCWSeries[expr_] :=
  MakeCWSeries[expr] = MakeCWSeries[Unique[MakeCWSeries], expr];
MakeCWSeries[ser_Symbol, expr_] := (
  ser[] = Hold[MakeCWSeries[ser, expr]];
  ser[d_Integer] := ser[d] = Expand[expr /. w_CW /; Deg[w] ≠ d → 0];
  CWSeries[ser]
);
MakeCWSeries[ab_List, coefs_] := MakeCWSeries[Unique[MakeCWSeries], ab, coefs];
MakeCWSeries[ser_Symbol, ab_List, coefs_Symbol] := (
  ser[] = Hold[MakeCWSeries[a, coefs]];
  ser[d_Integer] := ser[d] =
    Plus @@ MapIndexed[({coefs @@ Prepend[#2, d]} * #1) &, AllCyclicWords[d, ab]];
  CWSeries[ser]
);
RandomCWSeries[ab_List, opts_Rule] :=
  RandomCWSeries[Unique[RandomCWSeries], ab, opts];
RandomCWSeries[ser_Symbol, ab_List, opts_Rule] := Module[
  {rand = Randomizer /. {opts} /.
    Randomizer → (
      RandomInteger[{-2 Deg[#]!, 2 Deg[#]!}]
      Deg[#]!
    ) &},
  ser[] = Hold[RandomCWSeries[a, opts]];
  ser[d_Integer] := ser[d] = Plus @@ ((rand[#] * #) & /@ AllCyclicWords[d, ab]);
  CWSeries[ser]
];
s1_CWSeries ≡ s2_CWSeries := Module[{res = True, k},
  For[k = 1, res && k ≤ $SeriesCompareDegree, ++k, res = res && (s1[k] == s2[k])];

```

```

    res
  ];
AddCWSeries[ss__CWSeries] := AddCWSeries[ss] = Module[{ser},
  ser = Unique[AddCWSeries];
  ser[] = Hold[AddCWSeries[ss]];
  ser[d_Integer] := ser[d] = Plus @@ ((#[d]) & /@ {ss});
  CWSeries[ser]
];
CWSeries /: Plus[ss__CWSeries] := AddCWSeries[ss];
ScaleCWSeries[c_, s_CWSeries] := ScaleCWSeries[c, s] = Module[{ser},
  ser = Unique[ScaleCWSeries];
  ser[] = Hold[ScaleCWSeries[c, s]];
  ser[d_Integer] := ser[d] = Expand[c*s[d]];
  CWSeries[ser]
];
CWSeries /: c_*s_CWSeries := ScaleCWSeries[c, s];
IntegrateCWSeries[cws_CWSeries, {s_, s0_, s1_}] :=
  IntegrateCWSeries[cws, {s, s0, s1}] = Module[{ser},
  ser = Unique[IntegrateCWSeries];
  ser[] = Hold[IntegrateCWSeries[cws, {s, s0, s1}]];
  ser[d_Integer] := ser[d] = Expand[Integrate[cws[d], {s, s0, s1}]];
  CWSeries[ser]
];
CWSeries /: Integrate[cws_CWSeries, {s_, s0_, s1_}] :=
  IntegrateCWSeries[cws, {s, s0, s1}];

tr[w_AW] := tr[w] = CW[RotateToMinimal@@w];
tr[expr_] := expr /. aw_AW => tr[aw];
tr[as_ASeries] := tr[as] = Module[{cws},
  cws = Unique[tr];
  cws[] = Hold[tr[as]];
  cws[d_] := cws[d] = tr[as[d]];
  CWSeries[cws]
];

tr[AW["yxyxyx"]]
CW[xxyxyy]

t1 = σ["y", BCHBase] // tr
CW[
   $\frac{CW[x]}{2}, \frac{CW[xx]}{12} - \frac{CW[xy]}{12}, -\frac{CW[xxy]}{24}$ 
]

t1[5]
 $\frac{CW[xxxxy]}{1440} - \frac{CW[xxxxy]}{180} + \frac{CW[xxyxy]}{120} + \frac{CW[xyyyy]}{480} - \frac{CW[xyxyy]}{720}$ 

```

```

div[y_LW, w_LW] /; Deg[y] == 1 := div[y, w] = tr[(AW@@y) ** σ[y, w]];
div[y_, ls_LieSeries] := div[y, ls] = Module[{cws},
  cws = Unique[div];
  cws[] = Hold[div[y, ls]];
  cws[d_] := cws[d] = div[LW[y], ls[d]];
  CWSeries[cws]
];
div[y_, expr_] := Expand[expr /. w_LW => div[LW[y], w]];
div_y_[expr_] := div[y, expr];

```

```
{div["x", BCHBase][7], div["y", BCHBase][7]}
```

$$\left\{ \begin{aligned} & -\frac{CW[xxxxxxy]}{30\,240} + \frac{CW[xxxxxyy]}{2520} - \frac{CW[xxxxyxy]}{1008} - \frac{19\,CW[xxxxyyy]}{15\,120} + \\ & \frac{CW[xxxxyxy]}{2520} + \frac{CW[xxxxyyy]}{504} + \frac{CW[xxxxyxy]}{504} + \frac{19\,CW[xxxxyyy]}{15\,120} + \\ & \frac{CW[xxyxyxy]}{1680} - \frac{CW[xxyxyxy]}{280} - \frac{CW[xxyxyyy]}{504} - \frac{CW[xxyxyyy]}{1680} - \frac{CW[xxyxyxy]}{504} - \\ & \frac{CW[xxyxyyy]}{2520} + \frac{CW[xyxyxyy]}{280} + \frac{CW[xyxyyyy]}{1008} - \frac{CW[xyxyyyy]}{2520} + \frac{CW[xxyxyyy]}{30\,240}, \\ & \frac{CW[xxxxxxy]}{30\,240} - \frac{CW[xxxxxyy]}{2520} + \frac{CW[xxxxyxy]}{1008} + \frac{19\,CW[xxxxyyy]}{15\,120} - \frac{CW[xxxxyxy]}{2520} - \\ & \frac{CW[xxxxyxy]}{504} - \frac{CW[xxxxyyy]}{504} - \frac{19\,CW[xxxxyyy]}{15\,120} - \frac{CW[xxyxyxy]}{1680} + \\ & \frac{CW[xxyxyxy]}{280} + \frac{CW[xxyxyyy]}{504} + \frac{CW[xxyxyyy]}{1680} + \frac{CW[xxyxyxy]}{504} + \\ & \frac{CW[xxyxyyy]}{2520} - \frac{CW[xyxyxyy]}{280} - \frac{CW[xyxyyyy]}{1008} + \frac{CW[xyxyyyy]}{2520} - \frac{CW[xxyxyyy]}{30\,240} \end{aligned} \right\}$$

```
t1 = MakeCWSeries[CW["xyxyyyy"]] //
```

```
LieDerivation[{LW["x"] -> MakeLieSeries[b[LW["x"], LW["z"]]]}]
```

```
CWS[0, 0, 0]
```

```
t1 /@ Range[10]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, -CW[xyxyyyyz] + CW[xyxzyyyy] - CW[xyyyyxyz] + CW[xyyyyxzy], 0, 0}
```

The Meta-Cocycle JA

```

JA[-1, ___] = MakeCWSeries[0];
JA[n_, y_LW, μ_LieSeries, ss_] := JA[n, y, μ, ss] = Module[
  {s, sμ, μs},
  sμ = ScaleLieSeries[s, μ];
  μs = StableApply[LieMorphism[{y → Ad[ScaleLieSeries[1, sμ]][LW[z]]}], μ];
  μs = μs // LieMorphism[{LW[z] → y}];
  IntegrateCWSeries[
    AddCWSeries[
      JA[n-1, y, μ, s] // LieDerivation[{y → b[μs, y]}],
      div[y, μs]
    ],
    {s, 0, ss}
  ]
];
JA[y_LW, μ_LieSeries] := JA[y, μ] = Module[{cws, s},
  cws = Unique[JA];
  cws[] = Hold[JA[y, μ]];
  cws[d_Integer] := cws[d] = JA[d-1, y, μ, s][d] /. s → 1;
  CWSeries[cws]
];
Print /@ {y0 = LW["y"], μ0 = BCHBase,
  JA[0, y0, μ0, s],
  JA[1, y0, μ0, s],
  JA[2, y0, μ0, s],
  JA[y0, μ0]
};

```

⟨y⟩

$$\text{LS} \left[\langle x \rangle + \langle y \rangle, \frac{\langle xy \rangle}{2}, \frac{\langle xxy \rangle}{12} + \frac{\langle xyy \rangle}{12} \right]$$

$$\text{CWS} \left[s \text{CW}[y], \frac{1}{2} s \text{CW}[xy] + \frac{1}{2} s^2 \text{CW}[xy], \right. \\ \left. \frac{1}{12} s \text{CW}[xxy] + \frac{1}{4} s^2 \text{CW}[xxy] + \frac{1}{6} s^3 \text{CW}[xxy] - \frac{1}{12} s \text{CW}[xyy] - \frac{1}{4} s^2 \text{CW}[xyy] - \frac{1}{6} s^3 \text{CW}[xyy] \right]$$

$$\text{CWS} \left[s \text{CW}[y], \frac{1}{2} s \text{CW}[xy] + \frac{1}{2} s^2 \text{CW}[xy], \right. \\ \left. \frac{1}{12} s \text{CW}[xxy] + \frac{1}{4} s^2 \text{CW}[xxy] + \frac{1}{6} s^3 \text{CW}[xxy] - \frac{1}{12} s \text{CW}[xyy] - \frac{1}{4} s^2 \text{CW}[xyy] - \frac{1}{6} s^3 \text{CW}[xyy] \right]$$

$$\text{CWS} \left[s \text{CW}[y], \frac{1}{2} s \text{CW}[xy] + \frac{1}{2} s^2 \text{CW}[xy], \right. \\ \left. \frac{1}{12} s \text{CW}[xxy] + \frac{1}{4} s^2 \text{CW}[xxy] + \frac{1}{6} s^3 \text{CW}[xxy] - \frac{1}{12} s \text{CW}[xyy] - \frac{1}{4} s^2 \text{CW}[xyy] - \frac{1}{6} s^3 \text{CW}[xyy] \right]$$

$$\text{CWS} \left[\text{CW}[y], \text{CW}[xy], \frac{\text{CW}[xxy]}{2} - \frac{\text{CW}[xyy]}{2} \right]$$

$$\text{CWS} \left[s \text{CW}["Y"], \frac{1}{2} s \text{CW}["xy"] + \frac{1}{2} s^2 \text{CW}["xy"], \frac{1}{12} s \text{CW}["xxy"] + \frac{1}{4} s^2 \text{CW}["xxy"] + \frac{1}{6} s^3 \text{CW}["xxy"] - \frac{1}{12} s \text{CW}["xyy"] - \frac{1}{4} s^2 \text{CW}["xyy"] - \frac{1}{6} s^3 \text{CW}["xyy"] \right] /. s \rightarrow 1$$

$$\text{CWS} \left[\text{CW}["Y"], \text{CW}["xy"], \frac{\text{CW}["xxy"]}{2} - \frac{\text{CW}["xyy"]}{2} \right]$$

`$SeriesCompareDegree = $SeriesShowDegree = 8;`

`JA[3, y0, μ0, s] ≡ JA[4, y0, μ0, s]`

True

`JA[y0, μ0][6]`

$$\frac{\text{CW}["xxxxxy"]}{120} + \frac{31 \text{CW}["xxxxyy"]}{48} - \frac{11 \text{CW}["xxxxyxy"]}{6} + \frac{109 \text{CW}["xxxxyyy"]}{36} +$$

$$\frac{7 \text{CW}["xxyxxy"]}{8} - \frac{23 \text{CW}["xxyxyy"]}{4} - \frac{23 \text{CW}["xxyyxy"]}{4} + \frac{31 \text{CW}["xxyyyy"]}{48} +$$

$$\frac{28 \text{CW}["xyxyxy"]}{3} - \frac{11 \text{CW}["xyxyyy"]}{6} + \frac{7 \text{CW}["xyyxxy"]}{8} + \frac{\text{CW}["xyyyyy"]}{120}$$