

Pensieve header: γ 2-checks for OneCo. Branched from OneCoComputations.nb.

The bracket

In the basis $\{a[f,i,j], \gamma[f,i,j], \gamma[f,i,j,k], \gamma a[f,i,j,k,l]\}$.

Generalities

```

DQ[is__] := (Sort[{is}] === Union[{is}]);

LSimp[ $\gamma[f_, j_, k_, k_]$ ] = 0;
LSimp[ $\gamma[f_, j_, k_, l_]$ ] /; OrderedQ[{l, k}]  $\wedge$  DQ[k, l] :=
   $\gamma[-f, j, l, k]$  // LSimp;
LSimp[ $\gamma a[f_, i_, j_, i_, k_]$ ] /; OrderedQ[{k, j}]  $\wedge$  DQ[k, j] :=
   $\gamma[b_i f, i, j, k] + \gamma a[f, i, k, i, j]$  // LSimp;
LSimp[ $\beta[f_]$ ] :=  $\beta[\text{Expand}[f]]$ ;
LSimp[a[i_, j_]] := a[i, j];
LSimp[a[f_, i_, j_]] := a[Expand[f], i, j];
LSimp[ $\gamma[f_, is_]$ ] :=  $\gamma[\text{Expand}[f], is]$ ;
LSimp[ $\gamma a[f_, is_]$ ] :=  $\gamma a[\text{Expand}[f], is]$ ;
LSimp[expr_] := expr /. (t_ $\beta$  | t_ $a$  | t_ $\gamma$  | t_ $\gamma a$ )  $\Rightarrow$  LSimp[t];

 $\beta[0]$  := 0;
 $\beta$  /:  $\beta[f_] + \beta[g_] := \beta[f+g]$ ;
 $\beta$  /:  $-\beta[f_] := \beta[-f]$  // LSimp;
a[0, __] := 0;
a /: a[f_, i_, j_] + a[g_, i_, j_] := a[f+g, i, j];
a /:  $-a[f_, is_] := a[-f, is]$  // LSimp;
 $\gamma[0, __] := 0$ ;
 $\gamma$  /:  $\gamma[f_, is_] + \gamma[g_, is_] := \gamma[f+g, is]$ ;
 $\gamma$  /:  $-\gamma[f_, is_] := \gamma[-f, is]$  // LSimp;
 $\gamma a[0, __] := 0$ ;
 $\gamma a$  /:  $\gamma a[f_, is_] + \gamma a[g_, is_] := \gamma a[f+g, is]$ ;
 $\gamma a$  /:  $-\gamma a[f_, is_] := \gamma a[-f, is]$  // LSimp;

B[0, _] = 0; B[_, 0] = 0;
B[x_, x_] = 0;
B[x_Plus, y_] := B[#, y] & /@ x;
B[x_, y_Plus] := B[x, #] & /@ y;

```

Specific Brackets

[a,a] brackets:

```

B[a[f_, j_, k_], a[g_, l_, m_]] := Plus[
  B[a[j, k], a[l, m]] /. {a[h_, is_] => a[f g h, is],  $\gamma$ [h_, is_] =>  $\gamma$ [f g h, is]},
   $\gamma$ a[f ( $\partial_{b_j} g - \partial_{b_k} g$ ), j, k, l, m] +  $\gamma$ a[g ( $\partial_{b_m} f - \partial_{b_l} f$ ), l, m, j, k]
] // LSimp;
(* tt *) B[a[j_, k_], a[j_, l_]] /; DQ[j, k, l] :=  $\gamma$ [1, j, k, l] // LSimp;
(* hh *)
B[a[j_, k_], a[i_, k_]] /; DQ[i, j, k] := a[b_i, j, k] - a[b_j, i, k] // LSimp;
(* th *) B[a[j_, k_], a[i_, j_]] /; DQ[i, j, k] :=
  a[b_j, i, k] - a[b_i, j, k] +  $\gamma$ [1, i, j, k] // LSimp;
(* ht *) B[a[j_, k_], a[k_, l_]] /; DQ[j, k, l] := -B[a[k, l], a[j, k]];
(* loc *) B[a[j_, k_], a[l_, m_]] /; DQ[j, k, l, m] := 0;

```

[a, β] brackets:

```
B[a[f_, j_, k_],  $\beta$ [g_]] :=  $\gamma$ [f ( $\partial_{b_j} g - \partial_{b_k} g$ ), j, k] // LSimp;
```

[a, γ_2] brackets: (only γ_2 's in output!)

```

(* tt $\gamma$  *) B[a[f_, j_, k_],  $\gamma$ [g_, j_, l_]] /; DQ[j, k, l] := 0;
(* hh $\gamma$  *)
B[a[f_, j_, k_],  $\gamma$ [g_, i_, k_]] /; DQ[i, j, k] := - $\gamma$ [b_j f g, i, k] // LSimp;
(* th $\gamma$  *)
B[a[f_, j_, k_],  $\gamma$ [g_, i_, j_]] /; DQ[i, j, k] :=  $\gamma$ [b_j f g, i, k] // LSimp;
(* ht $\gamma$  *) B[a[f_, j_, k_],  $\gamma$ [g_, k_, l_]] /; DQ[j, k, l] :=
   $\gamma$ [b_j f g, k, l] -  $\gamma$ [b_k f g, j, l] // LSimp;
(* loc $\gamma$  *) B[a[f_, j_, k_],  $\gamma$ [g_, l_, m_]] /; DQ[j, k, l, m] := 0;
B[a[f_, j_, k_],  $\gamma$ [g_, j_, k_]] :=  $\gamma$ [-b_j f g, j, k] // LSimp;

```

[a, γ_3] brackets:

```

(* ttγ3 *) B[a[f_, j_, k_], γ[g_, j_, l_, m_]] /; DQ[j, k, l, m] := 0;
(* thγ3-1 *) B[a[f_, j_, k_], γ[g_, i_, j_, l_]] /; DQ[i, j, k, l] :=
  γ[bjfg, i, k, l] + γa[fg, i, l, j, k] // LSimp;
(* thγ3-2 *) B[a[f_, l_, k_], γ[g_, i_, j_, l_]] /; DQ[i, j, k, l] :=
  γ[-blfg, i, k, j] + γa[-fg, i, j, l, k] // LSimp;
(* htγ3 *) B[a[f_, j_, k_], γ[g_, k_, l_, m_]] /; DQ[j, k, l, m] :=
  γ[-bkfg, j, l, m] + γ[bjfg, k, l, m] // LSimp;
(* hhγ3-1 *) B[a[f_, j_, k_], γ[g_, n_, i_, k_]] /; DQ[n, i, j, k] :=
  γ[-bjfg, n, i, k] + γa[fg, n, i, j, k] // LSimp;
(* hhγ3-2 *) B[a[f_, j_, i_], γ[g_, n_, i_, k_]] /; DQ[n, i, j, k] :=
  γ[bjfg, n, k, i] + γa[-fg, n, k, j, i] // LSimp;
(* a-th-1 *) B[a[f_, j_, k_], γ[g_, j_, k_, l_]] /; DQ[j, k, l] :=
  γa[-fg, j, k, j, l] // LSimp;
(* a-th-2 *) B[a[f_, j_, l_], γ[g_, j_, k_, l_]] /; DQ[j, k, l] :=
  γa[fg, j, l, j, k] // LSimp;
(* a-hh-1 *) B[a[f_, j_, k_], γ[g_, i_, j_, k_]] /; DQ[i, j, k] :=
  γ[-bjfg, i, j, k] + γa[fg, i, j, j, k] + γa[fg, i, k, j, k] // LSimp;
(* a-hh-2 *) B[a[f_, k_, j_], γ[g_, i_, j_, k_]] /; DQ[i, j, k] :=
  γ[bkfg, i, k, j] + γa[-fg, i, k, k, j] + γa[-fg, i, j, k, j] // LSimp;
(* locγ *) B[a[f_, i_, j_], γ[g_, k_, l_, m_]] /; DQ[i, j, k, l, m] := 0;

```

[a, γ] brackets:

```

B[x_a, γa[f_, i_, j_, m_, n_]] := Plus[
  B[x, γ[f, i, j]] /. γ[g_, k_, l_] => γa[g, k, l, m, n],
  B[x, a[l, m, n]] /. {a[g_, k_, l_] => γa[fg, i, j, k, l], _γ | _γa => 0}
] // LSimp

```

[β, a], [γ, a], [γ, γ] brackets:

```

B[x_β | x_γ | x_γa, y_a] := -B[y, x];
B[_β | _γ | _γa, _β | _γ | _γa] := 0;

```

γ_2 checks

```

βc[0, _] := 0;
βc /: βc[f_, i_] + βc[g_, i_] := βc[f+g, i];
βc /: -βc[f_, i_] := βc[Expand[-f], i];
δa[0, __] := 0;
δa /: δa[f_, is__] + δa[g_, is__] := δa[f+g, is];
δa /: -δa[f_, is__] := δa[Expand[-f], is];
B[a[f_, j_, k_], c[l_]] /; DQ[j, k, l] := 0;
B[a[f_, j_, k_], c[j_]] := γ[-f, j, k];
B[a[f_, j_, k_], c[k_]] := γ[f, j, k];
B[x_, βc[f_, i_]] := B[x, c[i]] /. γ[g_, l_, m_] => γ[f g, l, m];
(*B[x_, ca[f_, i_, j_, k_]] := Plus[
  B[x, c[i]] /. γ[g_, l_, m_] => γa[f g, l, m, j, k],
  B[x, a[f, j, k]] /. {a[g_, l_, m_] => ca[g, i, l, m], _γ → 0, _γa → 0}
];*)
(*δaa[f_, i_, j_, k_, l_] /; OrderedQ[{l, j}]∧DQ[i, j, k, l] := δaa[f, k, l, i, j];
δaa[f_, i_, j_, i_, l_] /; OrderedQ[{l, j}]∧DQ[j, l] := δaa[f, i, l, i, j];*)
B[x_, δa[f_, i_, j_]] := B[x, a[f, i, j]] /. {a → δa, _γ => 0};
Eγ2 = {
  γ[f_, j_, k_] => δa[f, j, k] + βc[Expand[-f b_j], k]
};
Check2[y_γ] := {t1 = B[a[f, j, k], y] /. Eγ2,
  t2 = B[a[f, j, k], y /. Eγ2] /. Eγ2, t1 == t2 // Simplify}

tt check

γ[g, j, l] // Check2
{0, 0, True}

th check

γ[g, i, j] // Check2 // ColumnForm
βc[-f g b_i b_j, k] + δa[f g b_j, i, k]
βc[-f g b_i b_j, k] + δa[f g b_j, i, k]
True

ht check

γ[g, k, l] // Check2 // ColumnForm
δa[f g b_j, k, l] + δa[-f g b_k, j, l]
δa[f g b_j, k, l] + δa[-f g b_k, j, l]
True

hh checks not done

```

```
 $\Upsilon$ [g, n, i, k] // Check3 // ColumnForm
```

```
ca[-f g bj, k, n, i] + ca[f g bj, i, n, k] + ca[-f g bn, i, j, k] +  $\delta$ aa[f g, n, i, j, k]  
ca[-f g bj, k, n, i] + ca[f g bj, i, n, k] + ca[-f g bn, i, j, k] +  $\delta$ aa[f g, n, i, j, k]  
True
```

```
 $\Upsilon$ [g, n, k, i] // Check3 // ColumnForm
```

```
ca[-f g bj, i, n, k] + ca[f g bj, k, n, i] + ca[f g bn, i, j, k] +  $\delta$ aa[-f g, n, i, j, k]  
ca[-f g bj, i, n, k] + ca[f g bj, k, n, i] + ca[f g bn, i, j, k] +  $\delta$ aa[-f g, n, i, j, k]  
True
```