gfun[`diffeq+diffeq`] - determine the differential equation satisfied by the sum of two holonomic functions

gfun[`diffeq*diffeq`] - determine the differential equation satisfied by the Cauchy product of two holonomic functions

gfun[hadamardproduct] - determine the differential equation satisfied by the Hadamard product of two holonomic functions

Calling Sequence

`diffeq+diffeq`(**eq1**, **eq2**, **y(z)**)

`diffeq*diffeq`(**eq1**, **eq2**, **y(z)**)

hadamardproduct(**eq1**, **eq2**, **y(z)**)

Parameters

`eq1, eq2` – two linear differential equations with polynomial coefficients

`y,z` – name of the holonomic function and itsXS generic variable

Description

- If **f** (resp. **g**) is a holonomic function solution of **eq1** (resp. **eq2**), **gfun[`diffeq+diffeq`]** outputs a linear differential equation verified by **f+g**, **gfun[`diffeq*diffeq`]** outputs a linear differential equation verified by **f*g**, and **gfun[hadamardproduct]** outputs a linear differential equation verified by the Hadamard product of **f** and **g** (the function whose coefficient of **z^n** in the Taylor expansion around 0 is the product of the corresponding coefficients of **f** and **g**).
- The differential order of the output equation is at most the sum of the input equations differential orders for **gfun[`diffeq+diffeq`]**, and their product for **gfun[`diffeq*diffeq`]**.

Examples

```
> with(gfun):
  eq1 := D(y)(x)-y(x):
  eq2 := (1+x)*(D@@2)(y)(x)+D(y)(x):
  `diffeq+diffeq`(eq1,eq2,y(x));
```

$$(-3-x)\left(\frac{d}{dx}y(x)\right)+\left(1-2x-x^2\right)\left(\frac{d^2}{dx^2}y(x)\right)+\left(2+3x+x^2\right)\left(\frac{d^3}{dx^3}y(x)\right) \quad \textbf{(2.1)}$$

```
> `diffeq*diffeq`(eq1,eq2,y(x));
```

$$x\,y(x)+(-1-2x)\left(\frac{d}{dx}y(x)\right)+(1+x)\left(\frac{d^2}{dx^2}y(x)\right) \quad \textbf{(2.2)}$$

```
> hadamardproduct(eq1,eq2,y(x));
```

$$\left\{(1+x)\left(\frac{d}{dx}y(x)\right)+x\left(\frac{d^2}{dx^2}y(x)\right)-\_C_1, y(0)=\_C_0, D(y)(0)=\_C_1\right\} \tag{2.3}$$

See Also

gfun, gfun[poltodiffeq], gfun[rec+rec], gfun[rec*rec], gfun[cauchyproduct],