Issue partial fractioning with GiNaC

July 2020

1 Introduction

Let us investigate here, when the function of GiNaC, *sqrfree parfrac* gives wrong results.

I first noticed this error, for terms of the form

$$\frac{1}{a}\frac{x^b}{(x-1)^c}\,. \tag{1}$$

Applying this to some particular example, using partial fractioning this should give

$$\frac{1}{4}\frac{x^4}{x-1} = \frac{1}{4}\left(x^3+x^2+x+\frac{1}{x-1}+1\right)\,. \tag{2}$$

The result outputed by GiNaC was

$$\frac{1}{4}\left(x^3+x^2+x+1\right)+\frac{1}{x-1}\,. \tag{3}$$

As you can see, the prefactor missing in front of the $\frac{1}{x-1}$ term is missing. That this is not simply an output error can be seen from plugging a value for the x term, here I used $x = 4$.

I investigated this and the following terms all give wrong result.

$$\frac{1}{4}\frac{x^4}{(x-1)^2}\,, \tag{4}$$

$$\frac{1}{4}\frac{x^3}{(x-1)^3}\,, \tag{5}$$

$$\frac{1}{3}\frac{x^3}{(x-1)^2}\,, \tag{6}$$

$$\frac{1}{5}\frac{x^2}{(x-1)^3}\,. \tag{7}$$

Notice however, that terms of the form (no denominator present for the numerical prefactor) give a correct result:

$$3\frac{x^3}{(x-1)^2}\,. \tag{8}$$

As you can see, there is an emerging pattern. We always see that the term $1/(...)$ initially present ín the expression is missing the (numerical) prefactor, if the prefactor is a ratio.

The code I used was

```
#include <iostream>
#include <complex>
#include <math.h>
#include <ginac/ginac.h>

using std::cout;
using std::endl;

int main()
{
    GiNaC::symbol x("x");

    GiNaC::ex ex1 = pow(x,4)/(x-1)/4;

    GiNaC::ex ex2 = GiNaC::sqrfree_parfrac(ex1,x);

    cout << "the input is ex1 = " << ex1 << endl;

    cout << "the result after partial fractioning: "<< ex2 << endl;

    cout << "this is what you get after x=4, in ex1 = "<< ex1.subs(x==4)<< endl;

    cout << "insert number:, say x = 4, which gives wrong result "
    << ex2.subs(x==4) << endl;

}
```

And I modified the term ex1 accordingly.