

CoCoA-5 - Design #917

CallOnGroebnerFanIdeals: better to use an iterator (or a "stream")

31 Aug 2016 14:25 - John Abbott

Status:	New	Start date:	31 Aug 2016
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	enhancing/improving	Estimated time:	0.00 hour
Target version:	CoCoA-5.?.?	Spent time:	0.45 hour
Description			
The fn CallOnGroebnerFanIdeals is not a good way of achieving its end. Design and implement a better way.			
Related issues:			
Related to CoCoA-5 - Feature #903: New function CallOnGroebnerFanIdeals: call...		Closed	04 Jul 2016

History

#1 - 31 Aug 2016 14:28 - John Abbott

I have in mind a "stream", so the user could write CoCoA code like the following:

```
I := some_ideal;
foreach RGB in AllGFanIdealsSTREAM(I) do
  if not (has_nice_property(RGB)) then continue; endif;
  // store or process the interesting case
endforeach
```

I think such a notion of "streams" could be useful in many contexts (*e.g.* including cartesian products, integer ranges)

#2 - 31 Aug 2016 14:28 - Anna Maria Bigatti

but there is no such thing in CoCoA-5....

#3 - 31 Aug 2016 14:36 - Anna Maria Bigatti

John Abbott wrote:

I have in mind a "stream", so the user could write CoCoA code like the following:
[...]

Ok, that would be great indeed, but would require adding a type "STREAM" (and the semantic of foreach on it), correct?

#4 - 31 Aug 2016 16:06 - John Abbott

Yes, it would require some modification to the interpreter.

It would anyway be interesting to have such "stream" objects" also available in CoCoALib; perhaps they should not be called "stream" since that word is already used for I/O streams.

#5 - 10 Oct 2016 13:40 - Anna Maria Bigatti

- *Related to Feature #903: New function CallOnGroebnerFanIdeals: call function on GFan ideals added*

#6 - 26 Nov 2016 17:46 - John Abbott

- *Subject changed from CallOnGroebnerFanIdeals: better to use an iterator (or a "stream") to CallOnGroebnerFanIdeals: better to use an iterator (or a "stream")*

After recently delving into the source code for the interpreter to handle interrupts as I'd originally intended, I am now more cautious about proposing changes to the interpreter... :-/

In any case, making it an iterator in C++ surely seems like a good idea; but the difference in behaviour would mean that translating "well" CoCoA-5 code to CoCoALib requires some effort if the code uses CallOnGroebnerFanIdeals. This is probably not so important unless we expect there to be a lot of such code...