

CoCoA-5 - Slug #31

theValue makes copy

15 Nov 2011 11:21 - Anna Maria Bigatti

Status:	In Progress	Start date:	15 Nov 2011
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	60%
Category:	Parser/Interpreter	Estimated time:	10.00 hours
Target version:	CoCoA-5.4.2	Spent time:	6.05 hours
Description			
in Interpreter.H			
<pre>template <typename T> typename CoCoALibType<T>::type theValue(boost::intrusive_ptr<T> x);</pre>			
<pre>template <> BigInt theValue<BigIntValue>(boost::intrusive_ptr<BigIntValue> x);</pre>			
<pre>template <> const BigInt theValue<const BigIntValue>(boost::intrusive_ptr<const BigIntValue> x);</pre>			
(...)			
couldn't it be			
<pre>template <> const BigInt& theValue<const BigIntValue>(boost::intrusive_ptr<const BigIntValue> x);</pre>			
Related issues:			
Related to CoCoA-5 - Slug #687: Builtin fn makes unnecessary copy of arg		Closed	19 Apr 2015
Related to CoCoA-5 - Slug #1597: GetRow/GetRows are extraordinarily slow		Closed	27 May 2021
Related to CoCoA-5 - Slug #1228: SLUG: filling an array		In Progress	30 Sep 2018
Related to CoCoA-5 - Feature #1112: New function: IsEmpty		Closed	27 Oct 2017
Related to CoCoA-5 - Bug #946: Function "ideal" evaluates the argument twice		Resolved	17 Oct 2016

History

#1 - 15 Nov 2011 13:59 - Anna Maria Bigatti

- Subject changed from theValue make copy to theValue makes copy

#2 - 15 Nov 2011 14:00 - Anna Maria Bigatti

- Category set to Parser/Interpreter

#3 - 09 Feb 2012 16:56 - Giovanni Lagorio

- % Done changed from 0 to 50

The solution is to implement a move constructor in BigInt (and every "huge" data structure ;-)

#4 - 17 Jul 2012 14:37 - Anna Maria Bigatti

- Status changed from New to In Progress

open until we make experiments with C++11

#5 - 21 Sep 2012 14:32 - Anna Maria Bigatti

Giovanni Lagorio wrote:

The solution is to implement a move constructor in BigInt (and every "huge" data structure ;-)

Giovanni: do you have a easy reference/example about implementing and using a move constructor?

Do I need to do it **only** for the BigInt class (say) or also for the cocoa-5 wrapper class (I guess not, since that is a pointer....)

#6 - 02 Apr 2014 17:33 - Anna Maria Bigatti

- Target version set to CoCoA-5.1.0 Easter14

#7 - 17 Apr 2014 19:03 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.?.?

#8 - 22 Jan 2021 09:35 - Anna Maria Bigatti

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.2

#9 - 28 May 2021 15:24 - John Abbott

- Related to Slug #1597: GetRow/GetRows are extraordinarily slow added

#10 - 28 May 2021 17:59 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- Target version changed from CoCoA-5.4.2 to CoCoA-5.4.0

- % Done changed from 50 to 60

The (template) function theValue makes a copy of

```
typename CoCoALibType<T>::type
```

so I changed it so that it returns

```
const typename CoCoALibType<T>::type&
```

and modified accordingly all its implementations... relying on CVS to recover the original code ;-)

Well, it works, all tests pass. Too good to be true?

#11 - 28 May 2021 18:01 - Anna Maria Bigatti

Now investigate why the code is split in Interpreter.H and BuiltInOneLiners.H

#12 - 28 May 2021 20:35 - John Abbott

- Assignee deleted (*Anna Maria Bigatti*)
- Target version changed from *CoCoA-5.4.0* to *CoCoA-5.4.2*
- % Done changed from 60 to 50

Current plan of action:

- Anna will check in her changes (hinted at above)
- Anna will make an interim release of the source (so that "alpha testers" can help) -- maybe wait about a month so that problems are likely to surface
- Find out why the template fns are defined in *BuiltinOneLiners.H* but also declared in *Interpreter.H*; hopefully rectify
- probably rename the *Value* to something indicating that it returns a reference.
- investigate whether it makes sense to have almost duplicate template fns *RefTo* (if so, document the difference)

#13 - 28 May 2021 20:37 - John Abbott

- Related to *Slug #1228: SLUG: filling an array* added

#14 - 28 May 2021 20:42 - John Abbott

- Related to *Feature #1112: New function: isEmpty* added

#15 - 29 May 2021 15:37 - John Abbott

- Assignee set to *Anna Maria Bigatti*
- Target version changed from *CoCoA-5.4.2* to *CoCoA-5.4.0*
- % Done changed from 50 to 60

Correcting earlier mistake (when I forgot to reload the page)

#16 - 01 Jun 2021 11:46 - Anna Maria Bigatti

John Abbott wrote:

Current plan of action:

- Find out why the template fns are defined in *BuiltinOneLiners.H* but also declared in *Interpreter.H*; hopefully rectify

I think this was because I implemented *RefTo* in *BuiltinOneLiners.H*, and they are very similar. Now I moved the definition of the *Value* back to *Interpreter.H* and it works fine.

- Anna will check in her changes (hinted at above)

done

- Anna will make an interim release of the source (so that "alpha testers" can help) -- maybe wait about a month so that problems are likely to surface

done with "final" 5.3.3c

#17 - 18 Feb 2022 16:42 - Anna Maria Bigatti

- Target version changed from CoCoA-5.4.0 to CoCoA-5.4.2

#18 - 14 Mar 2023 22:12 - John Abbott

- Related to Bug #946: Function "ideal" evaluates the argument twice added