

CoCoALib - Design #1467

Change syntax `apply(phi,M)` into `phi(M)`?

22 Jun 2020 10:39 - John Abbott

Status:	Closed	Start date:	22 Jun 2020
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Improving	Estimated time:	2.01 hours
Target version:	CoCoALib-0.99800	Spent time:	1.95 hour
Description			
Currently to apply a RingHom to the entries of a matrix or a C++ vector we must use the fn <code>apply</code> .			
Is it better to use simpler notation where the RingHom can be applied directly: e.g. <code>phi(M)</code> or <code>phi(v)</code> ?			
Pros and cons? Opinions?			
Relevant source file is <code>apply.H</code>			
Related issues:			
Related to CoCoALib - Bug #1601: Compilation ambiguity		Closed	16 Jun 2021
Related to CoCoALib - Feature #1598: RingHom: implement <code>phi(X)</code> as <code>apply(phi, ...)</code>		Closed	10 Jun 2021
Related to CoCoA-5 - Design #1615: <code>apply</code> : remove for RingHom		Rejected	04 Oct 2021

History

#1 - 22 Jun 2020 10:44 - John Abbott

I noticed this while making a prototype impl for automatic ringelem promotion. In particular for ringelem times matrix the code ended up like this (**!!!note!!! the code has changed slightly due to issue [#635](#)**)

```
const RingHom promote = AutomaticConversionHom(Rx,R, "RingElem*Mat");
if (codomain(promote) == Rx)
    return x * apply(promote,M);
return promote(x) * M;
```

It would be slightly neater if I could write `promote(M)` instead of `apply(promote,M)`.

A feature of writing `apply(...)` is that it is obvious to the reader that `M` is not a plain ringelem (but that ought to be clear anyway).

At the moment, it seems to me to be "useless clutter". Some might argue that applying a ringhom directly to a matrix is an "abuse of notation" (but it is also clear, unambiguous and compact...)

NOTE aha! I see that CoCoA-5 wants to use `apply` when applying a ringhom to `MAT`, `LIST` or `RINGELEM` (why this last one???)

#2 - 22 Jun 2020 10:54 - John Abbott

- Status changed from *New* to *In Progress*

- % Done changed from 0 to 10

We could permit both syntaxes, perhaps making **apply(...)** obsolescent?
[JAA does not much like having two different but semantically equivalent syntaxes]

Or we could change CoCoA-5 too? Making **apply(...)** there obsolescent?

Opinions?

#3 - 22 Jun 2020 11:00 - John Abbott

- Description updated

#4 - 29 Oct 2020 13:35 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#5 - 02 Aug 2021 09:53 - John Abbott

- Related to Bug #1601: Compilation ambiguity added

#6 - 04 Oct 2021 11:50 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 90

In the end my hand was forced.

Some future version of C++ (maybe C++17?) defines a template fn **apply** which matches better than the CoCoA fns; there really seemed to be no way to make C++ use the CoCoA fns, so compilation failed. Perhaps the problem could be resolved using C++20 (with restrictions on when templates will match)? Anyway, not practicable at the moment.

So I have removed **apply** from CoCoALib, and changed all code which used it.
Everything compiles, and all tests pass.

Moving to "feedback".

#7 - 17 Feb 2022 19:33 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99800

#8 - 18 Feb 2022 15:02 - Anna Maria Bigatti

- Related to Feature #1598: RingHom: implement $\text{phi}(X)$ as $\text{apply}(\text{phi}, X)$ also for X vector and matrix added

#9 - 18 Feb 2022 15:03 - Anna Maria Bigatti

This issue overlaps with Feature [#1598](#).

#10 - 18 Feb 2022 15:04 - Anna Maria Bigatti

- Related to Design #1615: **apply**: remove for RingHom added

#11 - 18 Feb 2022 15:24 - Anna Maria Bigatti

- Status changed from In Progress to Closed

- % Done changed from 90 to 100

- Estimated time set to 2.01 h