CoCoA-5 - Feature #606

Evaluate in ring operator (was called :: in CoCoA-4)

01 Aug 2014 10:53 - Anna Maria Bigatti

Status:	Closed	**Start date:**	19 Mar 2014
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	CoCoA-4 function to be added	**Estimated time:**	1.00 hour
Target version:	CoCoA-5.4.0	**Spent time:**	0.70 hour

Description

CoCoA-4 has the **::** operator which evaluated expressions in a given ring:

```
R::x+y
```

It would be nice to have something similar in CoCoA-5; this would make it easier to specify poly ring homomorphisms, perhaps like this

```
P1 ::= QQ[x,y];
P2 ::= QQ[a,b];
PolyAlgHom(P1, P2, P2::[a^2,b^3]);
```

Currently you must Use P2 to create the vector of indet images, and then pass that to PolyAlgHom. This is relevant only at top level, I think.
or ReadExpr:

```
PolyAlgHom(P1, P2, [ReadExpr(P2,"a^2"), ReadExpr(P2,"b^3")])
```

Related issues:

Related to CoCoA-5 - Feature #484: Evaluate in other ring (was called :: in C...	**Closed**	**19 Mar 2014**
Related to CoCoA-5 - Bug #878: RingElem applied to a symbol (repr as a string)	**Closed**	**09 May 2016**

History

#1 - 11 May 2015 14:36 - John Abbott

- Target version changed from CoCoA-5.1.2 summer 2015 to CoCoA-5.1.3/4 Jan 2016

#2 - 17 Feb 2016 13:15 - John Abbott

- Target version changed from CoCoA-5.1.3/4 Jan 2016 to CoCoA-5.?.?

It might not be too painful if ReadExpr could read lists as well. The example in the issue description could then become:

```
PolyAlgHom(P1, P2, ReadExpr(P2,"[a^2,b^3]"));
```

Maybe there could be a fn called ReadExprList if we do not like extending ReadExpr to return different types of result.

I admit that it does "feel funny/awkward" having to put the value inside a string which is parsed only at run-time... it could be nice to have read-time parsing (but that would presumably need a new syntax).

Anyway, delaying this issue to some future version of CoCoA-5.

#3 - 09 May 2016 17:01 - John Abbott

- Related to Bug #878: RingElem applied to a symbol (repr as a string) added

#4 - 08 Oct 2016 22:04 - John Abbott

The suggestion of another function (perhaps called ReadExprList) still seems good to me.

Hmmm, but then again a list of lists would still be problematic :-/
In C++ the fn needs to have a well-defined return type, so we really do need different fns names for different return types (assuming the args continue to be a ring and a string).

#5 - 08 Oct 2020 13:57 - John Abbott

- Status changed from New to In Progress

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0

- % Done changed from 0 to 50

I think this has largely been resolved by RingElem or RingElems or RingElemList.
Right?

#6 - 03 Feb 2022 19:10 - John Abbott

- % Done changed from 50 to 70

Is this resolved to a satisfactory degree? I think BringIn and RingElem(P, string) cover most use cases.
Can we close?

#7 - 16 Feb 2022 20:02 - John Abbott

- Status changed from In Progress to Closed

- % Done changed from 70 to 100

- Estimated time changed from 4.00 h to 1.00 h

No one objected, so I am closing this issue.
I think a combination of CanonicalHom, BringIn and RingElem(R, str) should cover pratically all cases.