# CoCoA-5 - Feature #909

### ReadExpr: decimal point

19 Jul 2016 10:12 - Anna Maria Bigatti

Status:	Closed	Start date:	14 Jul 2016	
Priority:	Normal	Due date:		
Assignee:	John Abbott	% Done:	100%	
Category:	enhancing/improving	Estimated time:	2.00 hours	
Target version:	CoCoA-5.2.0 spring 2017	Spent time:	1.90 hour	
Description				
The function ReadExpr should also read the decimal point				
/**/ ReadExpr(QQ, "0.1");				
Related issues:				
Related to CoCoALib - Feature #910: BigRat: read from a string in "decimal" f			Closed	19 Jul 2016
Related to CoCoALib - Bug #938: ReadExpr: misreads 2/3^4			Closed	06 Oct 2016
Related to CoCoALib - Bug #936: ReadExpr: cannot handle exponent which is not			Closed	05 Oct 2016
Related to CoCoALib - Feature #1052: ReadExpr(P, string) and RingElem(P, stri			Closed	26 Apr 2017

#### History

#### #1 - 19 Jul 2016 13:21 - John Abbott

The current code simply uses operator>> for a BigInt.

We could replace it with operator>> for BigRat, but then that raises the question of whether it is OK to read a decimal expression as a BigRat.

Comments? Opinions? Ideas?

#### #2 - 19 Jul 2016 14:22 - John Abbott

- Related to Feature #910: BigRat: read from a string in "decimal" format? added

#### #3 - 21 Jul 2016 10:58 - John Abbott

Anna pointed out that CoCoA-4 has always converted "decimal" numbers into exact rationals; CoCoA-5 does so too. So it makes sense that ReadExpr does as well.

#### #4 - 21 Jul 2016 17:19 - John Abbott

- Status changed from New to Resolved

- % Done changed from 0 to 70

I have made the necessary (small) changes to RingElemInput.C. Currently there is no specific test for ReadExpr, but it seems to be used in other tests, and they all pass. :-)

#### #5 - 22 Jul 2016 12:47 - John Abbott

A simple quick test passed: ReadExpr(P, "1.2\*x"); produced the expected result.

JAA notes that there is a minor change in semantics compared to the old version of ReadExpr.

The difference becomes evident if you have a non-reduced rational and want to evaluate over a finite field. Previously ReadExpr(k, "6/3") would give error if k has characteristic 3 because numerator and denominator were separately converted to elements of k, and then divided; now it always succeeds because "6/3" is recognized as a rational, and so is automatically reduced (as a BigRat) before being converted to an element of k.

### #6 - 07 Oct 2016 15:11 - John Abbott

- Related to Bug #938: ReadExpr: misreads 2/3^4 added

### #7 - 08 Oct 2016 21:59 - John Abbott

- Status changed from Resolved to Feedback
- Assignee set to John Abbott
- % Done changed from 70 to 90

I think I have fixed it all now.

The comment about changed semantics (comment 5) no longer holds!

Note that 1.25 works like (5/4) where the surrounding brackets are important (e.g. so that powers works as expected)

### #8 - 08 Oct 2016 22:10 - John Abbott

- Related to Bug #936: ReadExpr: cannot handle exponent which is not integer literal added

## #9 - 27 Apr 2017 11:42 - John Abbott

- Related to Feature #1052: ReadExpr(P, string) and RingElem(P, string) in CoCoALib added

## #10 - 27 Apr 2017 11:43 - John Abbott

- Status changed from Feedback to Closed
- % Done changed from 90 to 100

Closing after 7 months in feedback without any problems (but probably not much stress testing either).