CoCoA-5 - Slug #1629

RingElem slow with many indets

08 Nov 2021 12:39 - John Abbott

Status:	Closed	**Start date:**	08 Nov 2021
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	6.33 hours
Target version:	CoCoA-5.4.2	**Spent time:**	6.35 hours

Description

Bernhard Andraschko reports the following (via email):

```
S ::= QQ[x[1..435,1..4],y[1..865,1..4]]; -- just some arbitrary high numbers
NumIndets(S);
indet(S,3456); -- very fast
RingElem(S,"y[429,4]"); -- extremely slow
```

Improve performance!

Related issues:

Related to CoCoALib - Slug #881: ReadExpr is too slow on large polys	**Closed**	**09 May 2016**
Related to CoCoALib - Slug #1238: ReadExpr is too slow on long lists of mono...	**Closed**	**21 Jan 2019**
Related to CoCoALib - Feature #1654: New function IsInSymbols	**New**	**28 Jan 2022**

History

#1 - 26 Nov 2021 14:58 - John Abbott

- Related to Slug #881: ReadExpr is too slow on large polys added

#2 - 26 Nov 2021 14:59 - John Abbott

- Related to Slug #1238: ReadExpr is too slow on long lists of monomial with many indets: ---> use RingElems instead added

#3 - 26 Nov 2021 15:03 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Simpler test:

```
/**/ P ::= QQ[x[1..10000]]; --> a bit slow
/**/ NumIndets(P); --> instant
10000
/**/ indet(P,9876); --> instant
x[9876]
/**/ RingElem(P, "x[9876]"); --> very slow > 70s
```

#4 - 26 Nov 2021 15:29 - Anna Maria Bigatti

- Category deleted (enhancing/improving)

- Assignee set to John Abbott

- Target version deleted (CoCoA-5.4.2)

After a common investigation via skype, we noticed that ReadExpr (called by RingElem) creates map<symbol, RingElem>& SymTable, in order to make the search faster (see #881).
In this case, with a ring with many indets, every RingElem in the map is quite big, so the creation of this map takes time and memory.

Try converting the code in RingElemInput so that it uses map<symbol, long>& SymTable, and then compare timing also for long polys with few indets (#881).

#5 - 26 Nov 2021 18:44 - John Abbott

- Category set to enhancing/improving

- Assignee deleted (John Abbott)

- Target version set to CoCoA-5.4.2

Anna implemented a quick fix: it was much faster, but would also sometimes give a wrong answer (we understood why).

The current idea is to re-design the interface regarding symbols:

- a dedicated function which says whether a symbol is in a ring (the result might be bool or it might be an index, with -1 meaning "not found").
- Using AreDistinct to detect if a symbol is in a ring is not efficient.
- how to produce the ringelem corresponding to a symbol without incurring a large time and memory overhead?
- one possible idea is to have inside the ring a std::map which returns the index of a symbol (if it corresponds to an indet); it seems that there is already a std::vector of ringelems, one for each symbol.

We need to think about the design, and perhaps make a prototype implementation.

#6 - 28 Jan 2022 08:52 - Anna Maria Bigatti

- Related to Feature #1654: New function IsInSymbols added

#7 - 07 Mar 2022 10:26 - Anna Maria Bigatti

Discussion:

John Abbott wrote:

> Anna implemented a quick fix: it was much faster, but would also sometimes give a wrong answer (we understood why).

Probably this cryptic comment is related to the fact that Anna's current trick (map<symbol, long>& SymTable) would not work for symbols in the coefficient ring, but only for indets:
a in QQ[a][x]

#8 - 11 May 2023 08:01 - Anna Maria Bigatti

I have my own prototype

```
/**/ P ::= QQ[x[1..10000]]; --> a bit slow
/**/ indet(P,9876); --> instant
/**/ RingElem(P, "x[9876]"); --> was > 70s, now instant
```

I cannot find a wrong output. I tried

```
/**/ P := NewPolyRing(NewFractionField(NewPolyRing(QQ, "a,b")), "x,y");
/**/ RingElem(P, "a*x -x +y/(a-b)");  // works fine -->
(a -1)*x +(1/(a -b))*y
```

I cannot remember what was going wrong and if it is fixed now, so (notes for me)

1. investigate
2. check in

#9 - 12 May 2023 07:52 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- % Done changed from 10 to 70

I checked my personal code.
My guess at the old history is that I made a map<symbol, long> (long instead of RingElem) with the indices of symbols(P), which, at first, would then build the i-th indet in P, giving the wrong answer mentioned in https://cocoa.dima.unige.it/redmine/issues/1629#note-5.

Now it calls RingElem(P, s), with s a symbol (in function ReadAtom, ch=='a', reading the symbol s) so it works correctly and I (at last!) check it in.

I think that now we should just make sure that RingElem(P, s) in each ring implementation is as fast as it can be.

#10 - 12 May 2023 07:59 - Anna Maria Bigatti

Thinking all this again, I noticed that we are now using the map just to detect if the symbol found is in the symbols of P.
That's why I had started the new issue #1654 (IsInSymbols).

#11 - 15 May 2023 20:36 - John Abbott

I must do a check-out, or Anna must do a check-in? The example from comment 8 is still slow on my computer

#12 - 16 May 2023 09:26 - John Abbott

I have just checked out Anna's update, and confirm that the first example in comment 8 is now fast. Thanks to Anna.

#13 - 19 May 2023 22:07 - John Abbott

Should we advance this issue to "feedback"?

#14 - 23 May 2023 14:50 - Anna Maria Bigatti

- Status changed from In Progress to Feedback

- % Done changed from 70 to 90

#15 - 22 Dec 2023 20:06 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 6.33 h

Closing after 7 months in feedback.