CoCoA-5 - Bug #1424

'UNCAUGHT UNKNOWN EXCEPTION' when aborting 'isin' computation

25 Feb 2020 17:29 - Julian Danner

Status:	Closed	**Start date:**	25 Feb 2020
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	bug	**Estimated time:**	2.90 hours
Target version:	CoCoA-5.3.0	**Spent time:**	2.80 hours

Description

If you call the 'isin' command and want to abort the computation using <ctrl+c>, CoCoA-5 crashes and outputs: **ERROR** UNCAUGHT UNKNOWN EXCEPTION.

For example use the following:

```
Use R::=ZZ/(101)[x,y,z];
I:=ideal([RandomLinearForm(R)^10*(RandomLinearForm(R)+x^2) | i in 1..10]);
f:=RandomLinearForm(R)^2;
f isin I;
```

If one aborts the computation with <ctrl+c> before it finishes, CoCoA-5 crashes. (Don't be too slow, this example does not give you much time...)

History

#1 - 25 Feb 2020 21:36 - John Abbott

I confirm the bug in 5.2.9.

#2 - 25 Feb 2020 21:58 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I expect the problem isin Interpreter.C at line 3316 where a CoCoA::ErrorInfo is caught, but InterruptReceived is of type CoCoA::exception.

#3 - 25 Feb 2020 22:04 - John Abbott

- Assignee set to John Abbott

- Target version set to CoCoA-5.3.0

- % Done changed from 10 to 50

I have fixed the problem, but want to discuss with Anna before checking in.

The problem was indeed where I said, but there is a deeper issue...

#4 - 26 Feb 2020 12:02 - John Abbott

After looking at the code, I think the same problem exists for modules.

#5 - 26 Feb 2020 12:12 - John Abbott

Here is a test for both ideals and modules....

```
Use R::=ZZ/(101)[x,y,z];
g := [RandomLinearForm(R)^13*(RandomLinearForm(R)+x^2) | i in 1..10];
I := ideal(g);
x isin I;   -- interrupt quickly!   RINGELEM isin IDEAL
I := ideal(g);
1 isin I;   -- interrupt quickly!   INT isin IDEAL
I := ideal(g);
1/2 isin I; -- interrupt quickly!  RAT isin IDEAL

// Same but for modules...
F := NewFreeModule(R,1);
S := submodule(F, [ModuleElem(F,[f]) | f in g]);
J := ModuleElem(F, [1]);
J isin S; -- interrupt quickly!

1/101 isin I; --> also caused a crash!
```

#6 - 26 Feb 2020 13:28 - John Abbott

The impl in Interpreter.C is not as clear as it could be... The relevant function is in lines 3285--3338
intrusive_ptr<Value> IsInExpression::implEval(RuntimeEnvironment *runtimeEnv) const

It needs to be tidied considerably. :-/

#7 - 26 Feb 2020 14:35 - John Abbott

- % Done changed from 50 to 70

I have tidied (incl. indented comprehensibly) the code. I have tested it with the example in comment 5, and all seems well.

NOTE: not sure how we could make an automatic test for this...

#8 - 26 Feb 2020 15:25 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 70 to 80

- Estimated time set to 2.50 h

I have checked all uses of ErrorInfo in Interpreter.C (and changed some indentation). It seems OK now.
Added another failing case to the test in comment 5.

#9 - 26 Feb 2020 16:24 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

- Estimated time changed from 2.50 h to 2.90 h

Anna has checked everything on her computer. Works fine.
Closing.