

CoCoA-5 - Bug #1527

Inadequate error message on incorrect list access

30 Oct 2020 10:19 - Julian Danner

Status:	Closed	Start date:	30 Oct 2020
Priority:	Low	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	2.33 hours
Target version:	CoCoA-5.4.0	Spent time:	2.35 hours
Description When trying to access a list using another list, the error message seems to be incorrect. Here is an example: <pre>L:=1..10; L[[1]];</pre> which gives the error ERROR: Expecting type INT or STRING, but found type LIST. AFAIK a list cannot be accessed by a string, at least I could find nothing in the manual about it.			

History

#1 - 30 Oct 2020 12:02 - John Abbott

- Category set to enhancing/improving
- Status changed from New to In Progress
- Target version set to CoCoA-5.4.0
- % Done changed from 0 to 10

The relevant function seems to be **IndexedAccessExpression::implEval** in Interpreter.C around lines 3677--3695.

Now sure how to fix it. Need to find out whether the indexed object is a RECORD; and if not, restrict indexes to being just integers.

#2 - 02 Nov 2020 21:42 - John Abbott

- Assignee set to John Abbott

I have a partial fix: the error mesg now says "Expecting type STRING" if the value being indexed is a RECORD otherwise it says "Expecting type INT".

This is not entirely satisfactory, because there are other "weird" cases where the error mesg is still unhelpful:

```
I := ideal(x);  
I[I]; --> ERROR: expecting type INT
```

Really it should check if the value being indexed is indexable, and if not then say "ERROR: type IDEAL cannot be indexed"

#3 - 02 Nov 2020 21:45 - John Abbott

There is a fn `canBeIndexedByBigInt`, but there does not seem to be a `canBeIndexedByString*`.

Maybe I should make the string version (it is a mem fn), and then test if both give false then complain that the object is not indexable.

That is probably more helpful to the user... do I want to spend more time deciphering the "self-documenting" interpreter? :-/

#4 - 02 Nov 2020 22:17 - John Abbott

Na toll:

```
--> ERROR: The type LIST cannot be indexed, but there is an index
```

Time to go to sleep...

#5 - 02 Nov 2020 22:20 - John Abbott

I do not understand lines 146 and 147 in AST.H.

They seem to say that everything can be indexed by both INT and STRING... seems like dodgy code to me!

#6 - 03 Nov 2020 21:12 - John Abbott

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

- % Done changed from 10 to 70

I have made a change (by adding a call to mem fn `asRightValue`).

Now the code seems to be doing more or less what I had hoped/expected.

Perhaps I'll wait until tomorrow before checking in.

I make no claims about correctness nor efficiency... it compiles and the CoCoA-5 tests passed!

#7 - 03 Nov 2020 22:17 - John Abbott

This is what happens now:

```
--> ERROR: Expecting type INT, but found type LIST
--> L[[1]];
-->    ^^
```

Also we get with an ideal:

```
/**/ I := ideal(x);
/**/ I[1];
--> ERROR: The type IDEAL cannot be indexed, but there is an index
--> I[1];
-->    ^
```

#8 - 04 Nov 2020 10:48 - John Abbott

- *Status changed from Resolved to Feedback*
- *% Done changed from 70 to 90*

I have checked in the code.

#9 - 08 Jan 2021 11:19 - John Abbott

- *Status changed from Feedback to Closed*
- *% Done changed from 90 to 100*
- *Estimated time set to 2.33 h*