

CoCoA-5 - Feature #674

Ring constructor: allow empty range for indices?

13 Mar 2015 11:08 - John Abbott

Status:	Rejected	Start date:	13 Mar 2015
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	1.21 hour
Target version:	CoCoA-5.4.0	Spent time:	1.20 hour
Description			
<p>I am writing a CoCoA-5 program which may need some "dummy" variables. My code looks like this:</p>			
<pre>NumDummies := ...; P ::= QQ[x[1..9], dummy[1..NumDummies]];</pre>			
<p>This code works fine if there is at least 1 dummy variable, but fails if there are none. The dot-dot operator complains that the end of the range is smaller than the start.</p>			
<p>Should we allow empty ranges?</p>			

History

#1 - 13 Mar 2015 13:21 - John Abbott

There would be no technical problem allowing a range such as 1..0; perhaps the main question is whether the gain in extra flexibility is "greater" than the reduction in safety (protecting a user from carelessly inputting a daft range).

If we do allow empty ranges, must they be of the form $n..(n-1)$ or would we also allow "very empty" ranges such as 2..0 or in general $n..(n-k)$ for some $k > 1$?

#2 - 13 Mar 2015 13:56 - Anna Maria Bigatti

John Abbott wrote:

There would be no technical problem allowing a range such as 1..0; perhaps the main question is whether the gain in extra flexibility is "greater" than the reduction in safety (protecting a user from carelessly inputting a daft range).

I think we should not allow empty ranges.

I find it more likely that a user writes $x[4..1]$ thinking to get $[x[4], x[3], x[2], x[1]]$

#3 - 13 Mar 2015 14:35 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I understand your point. Actually, I suppose the correct solution is to use anonymous indets for the dummies, but we do not have those in CoCoA-5.

So how do I make my code work? I could create at least 1 dummy variable (even if I will never use it), or I could put in an if statement and use two different commands to build the ring I want to work in. Neither solution appeals to me much.

You would not be willing to compromise:

- `n..(n-1)` is OK and gives the empty range
- `n..(n-k)` for $k > 1$ gives an error

#4 - 13 Mar 2015 14:50 - Anna Maria Bigatti

John Abbott wrote:

I understand your point.

Actually, I suppose the correct solution is to use anonymous indets for the dummies, but we do not have those in CoCoA-5.

That's another problem, but we decided it is not worth having anonymous symbols in cocoa-5 (and I think it was a sensible decision)

So how do I make my code work? I could create at least 1 dummy variable (even if I will never use use), or I could put in a if statement and use two different commands to build the ring I want to work in. Neither solution appeals to me much.

I would go for the second. The code might in principle be different if you have no dummies.

You would not be willing to compromise:

- `n..(n-1)` is OK and gives the empty range

I'd rather not: `x[2..1]` in my head is closer to `[x_2, x_1]` than to `[]`

- `n..(n0k)` for $k > 1$ gives an error

?

#5 - 13 Oct 2020 15:00 - John Abbott

- Assignee set to John Abbott
- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0
- % Done changed from 10 to 30

This was last considered 5 years ago. The problem has not arisen in this time (in my experience), so I think that it is a rare occurrence. The logical conclusion would be that there is no real demand for empty ranges -- in those rare cases where they might be needed, simple (if tedious) workarounds exist.

As a consequence I suggest rejecting this proposal. Anna?

#6 - 14 Oct 2020 14:43 - Anna Maria Bigatti

- Status changed from In Progress to Rejected
- % Done changed from 30 to 100
- Estimated time set to 1.21 h