

CoCoA-5 - Design #403

Misuse of dot-dot operator

16 Sep 2013 10:46 - John Abbott

Status:	Rejected	Start date:	16 Sep 2013
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Parser/Interpreter	Estimated time:	2.20 hours
Target version:	CoCoA-5.1.1 Seoul14	Spent time:	2.15 hours

Description

Giulia Bianco produced an **interesting misuse** of the .. operator, namely $v[1]..v[r]$ thinking (not unreasonably) that it would produce the list $[v[1], \dots, v[r]]$.

In her case the result was a mystifying error message about something not being an indeterminate (because the values in the list v were RINGELEMs).

Reconsider what arg types operator .. will accept. We added a "hack" to allow .. to work between polynomials provided the values were actually "indeterminates" -- this allows CoCoA-5 to simulate CoCoA-4 when specifying ranges of indeterminates (when doing elimination).

The correct way to produce the result Giulia wanted is $[v[j] \mid j \text{ in } 1..r]$ which is somewhat cumbersome; is it worth attempting to do better?

History

#1 - 16 Sep 2013 10:58 - John Abbott

Making operator .. work "as expected" for indeterminates is not entirely simple.

Consider the expression $x[1]..x[2]$

In CoCoA-4 there was no doubt what the expression means: it will give the list $[x[1], x[2]]$ or an error either indet does not exist.

As currently implemented in CoCoA-5 the result is not uniquely defined!! Consider the following two examples:

(A)

```
Use QQ[x[1],x[2],x[3]];
x[1]..x[2]; --> same result as in CoCoA-4
```

(B)

```
Use QQ[x[1],y,z,x[3],x[2]];
x[1]..x[2]; --> list of all 5 indets!!
```

because internally it identifies $x[1]$ as the 1-st indet, and $x[2]$ as the 5-th indet, so it produces the list of all indets from 1-st to 5-th.

The behaviour in case (B) is unlikely to be what most people would expect. **This is bad**

#2 - 16 Sep 2013 11:06 - John Abbott

In CoCoA-4 the range of indets produced was simply from the i-th to the j-th where i is the index of the first indet, and j the index of the second. No heed was taken of the names of the indets. Here are two examples:

Example (AA)

```
Use QQ[x, y, z, t];  
x..z; --> [x, y, z]
```

Example (BB)

```
Use QQ[x[1..3], y[1..3]];  
x[1]..y[2]; --> [x[1], x[2], x[3], y[1], y[2]]
```

Note that the range produced is determined solely by the order of the indets appearing the ring definition, and specifically does not depend on the PP ordering.

JAA does not like (BB) much; it could easily have been a typo for x[1]..x[2]

#3 - 16 Sep 2013 11:08 - John Abbott

Is the indet version of operator .. ever used outside a call to elim?

#4 - 02 Apr 2014 09:53 - Anna Maria Bigatti

- Subject changed from dot-dot operator to Misuse of dot-dot operator
- Category set to Parser/Interpreter
- Status changed from New to In Progress
- Target version set to CoCoA-5.1.0 Easter14
- % Done changed from 0 to 10

#5 - 09 Apr 2014 17:29 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.1.1 Seoul14

#6 - 29 Jul 2014 07:28 - Anna Maria Bigatti

I suggest to reject this issue.

A..B works for A and B integers or indeterminates in the order of their definition (?Range Operator).
I think what it does non need to do anything more than what it already does.
(too much effort to invent new meanings which are more powerful and safe at the same time)

#7 - 29 Jul 2014 11:54 - John Abbott

- % Done changed from 10 to 20
- Estimated time set to 2.20 h

The main action taken was to clarify the manual entry for ".."; it is now quite clear, and even mentions Giulia's incorrect use, and explains what one should do instead.

After re-reading this, it seems that my implicit suggestion was to consider forbidding ".." between indets. Is it ever used outside the context of elimination? Do we really want to retain it for elimination, or is there a better alternative?
[think about backward compatibility]

#8 - 29 Jul 2014 12:04 - John Abbott

- Status changed from In Progress to Rejected
- Assignee set to John Abbott
- % Done changed from 20 to 100

My main concern is the (possibly) unexpected behaviour illustrated in example **(B)** in comment 1. However, for a long time CoCoA-4 has had the behaviour illustrated in example **(BB)** in comment 2, and no complaints have been received about it.

Also, while the behaviour shown in **(B)** is "surprising", I am also inclined to think that anyone who defines a "strange" polynomial ring like the one in that example cannot justifiably complain if careless use of operator .. produces a strange result.

I suppose the principle of not needlessly breaking backward compatibility will win in this case (*i.e.* the improved manual is sufficient response).

#9 - 29 Jul 2014 12:31 - John Abbott

Giulia's problem could be neatly resolved by allowing the following syntax **L[2..4]** to mean $[L[k] \mid k \text{ in } 2..4]$. This could be achieved by creating a new type (*e.g.* IntRange) which can be used after the in keyword, and in other contexts we want.

Big question: is it worth it? Selecting a contiguous sublist is not very common.

#10 - 29 Jul 2014 16:26 - Anna Maria Bigatti

the error message was indeed a bit puzzling in some cases now it is:

```
x..(y-1);  
--> ERROR: dot-dot: only between INTs, or between indets of the same ring  
--> x..(y-1);  
-->      ^^^
```