

## CoCoA-5 - Bug #321

### interpreter proposes no near misses for "bin"

25 Feb 2013 22:13 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	25 Feb 2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Parser/Interpreter	<b>Estimated time:</b>	4.10 hours
<b>Target version:</b>	CoCoA-5.1.1 Seoul14	<b>Spent time:</b>	4.00 hours
<b>Description</b>			
In C5, if you slightly mistype a name, the interpreter offers a choice of similar names.  But if I type <b>bin()</b> then the suggestions are <b>Min</b> and <b>min</b> ; even though <b>Bin</b> exists it is not offered!  Why?			
<b>Related issues:</b>			
Related to CoCoA-5 - Bug #104: Misprints suggestions: Z --> ZZ		<b>Closed</b>	<b>15 Mar 2012</b>

### History

#### #1 - 27 Feb 2013 12:51 - John Abbott

- Subject changed from *interpreter proposes no near misses for "Ord"* to *interpreter proposes no near misses for "bin"*

#### #2 - 27 Feb 2013 12:58 - John Abbott

- Category set to *Parser/Interpreter*
- Status changed from *New* to *In Progress*
- Target version set to *CoCoA-5.0.9*

The problem is caused by the fact the suggestions offered depend on the arity in the context in which the name appears.

Since **Bin** has fixed arity (=2), but the arity was 0 in the input, the interpreter does not propose it. **Min** and **min** have variable arity, so they are offered.

If **bin** is input as a variable then all suggestions appear because any name can be used in that context.

What to do? Arity filtering seems like a good idea, but in this case it produces quite a misleading result.

Here are two possible courses of action:

1. we could disable arity filtering (might produce lots of useless suggestions?)
2. we could tell the interpreter to offer two sets of suggestions: those with suitable arity; and any others that match.

#### #3 - 01 Aug 2013 07:20 - Anna Maria Bigatti

- Target version changed from *CoCoA-5.0.9* to *CoCoA-5.1.0 Easter14*

#### #4 - 09 Apr 2014 17:31 - John Abbott

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

## #5 - 28 Aug 2014 12:54 - John Abbott

- % Done changed from 0 to 10

Now we get slightly different output:

```
>>> bin();
--> ERROR: Cannot find a variable named "bin" in scope.
Similarly named variables (that you might need to import) are: "Gin", "Min", "min"
--> bin();
--> ^^^

>>> bin;
--> ERROR: Cannot find a variable named "bin" in scope.
Similarly named variables (that you might need to import) are: "Bin", "Gin", "Min", "gin", "min"
--> bin;
--> ^^^
```

Notice that Gin appears in both lists of suggestions, while gin appears in only the second list. This is because of how Gin is defined in BackwardCompatible.cpkg5.

## #6 - 02 Sep 2014 11:00 - John Abbott

- Target version changed from CoCoA-5.1.1 Seoul14 to CoCoA-5.?.?

## #7 - 02 Sep 2014 14:06 - John Abbott

The place to intervene appears to be Interpreter.C:2775 in the function collectSimilarlyNamedIdentifiers. The procedure returns its result in a single list, so to obtain two lists it will be necessary to change the signature of the procedure.

## #8 - 02 Sep 2014 23:46 - John Abbott

- Assignee set to John Abbott

I propose the following variation on current behaviour:

- the suggested names should be in order of increasing *levenshtein* distance
- if the name appears as a fn name in a call then name with "inappropriate" arity should appear but only after those with appropriate arity

This means that the set of suggested names does not vary with context, but the order in which they appear may vary (hopefully the "better matches" being earlier in the list).

What do you think? It should not be too hard to implement :-)

**NOTE** (20140903) damn, it won't be as easy as I'd hoped because Giovanni used std::set rather than list/vector.

**#9 - 03 Sep 2014 07:47 - Anna Maria Bigatti**

John Abbott wrote:

I propose the following variation on current behaviour:

- the suggested names should be in order of increasing *levenshtein* distance
- if the name appears as a fn name in a call then name with "inappropriate" arity should appear but only after those with appropriate arity

yes! I like that!

**#10 - 03 Sep 2014 12:19 - John Abbott**

- % Done changed from 10 to 40

I now have a first impl.

- changed the use of `std::set` into `set::vector`
- implemented the idea of printing the names in increasing order of "remoteness"

I note that there are two distinct fns called `collectSimilarlyNamedIdentifiers`; I have changed only one of them. The other one appears to be recursive... not sure when it is called.

I'll talk with Anna; if she approves, I'll check in.

**#11 - 04 Sep 2014 12:54 - John Abbott**

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

- Target version changed from *CoCoA-5.?.?* to *CoCoA-5.1.1 Seoul14*

- % Done changed from 40 to 100

- Estimated time set to 4.10 h

After some testing Anna and John approve the idea in comment 8 (already implemented).

We have also increased the max distance allowed when comparing names to  $(len+3)/4$  where `len` is the length of the name typed by the user.