

## CoCoA-5 - Design #668

### Remove keyword "weights"?

05 Mar 2015 18:26 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	05 Mar 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	Parser/Interpreter	<b>Estimated time:</b>	4.01 hours
<b>Target version:</b>	CoCoA-5.1.2 summer 2015	<b>Spent time:</b>	3.80 hours
<b>Description</b> In CoCoA-4 "Weights" was used to set weights in a polynomial ring construction. Now weights are dealt with in a completely different matter (see manual for NewPolyRing), and it is unlikely that the keyword will be used again. Should we remove it?			
<b>Related issues:</b> Related to CoCoA-5 - Feature #673: Error message: I was expecting... <b>Closed</b> <b>11 Mar 2015</b>			

### History

#### #1 - 05 Mar 2015 18:40 - Anna Maria Bigatti

- Status changed from New to In Progress
- % Done changed from 0 to 20

The problem arose when Claudia Fassino worked with PreprocessPoint, and had trouble getting the value of P.weights, this happened because "weight" is a protected keyword (even though unused).

Such record field cannot be created in CoCoA-5, but can be created in BuiltinFunctions.C. In any case it cannot be accessed in any way, not even P["weights"]. (well, I did find an ugly workaround, so ugly I'm not writing it here ;-)

#### #2 - 05 Mar 2015 18:43 - Anna Maria Bigatti

The involved lines are:

```
Lexer.C:391:      Keyword("Weights", TT_WEIGHTS),
Parser.C:178:    case TT_WEIGHTS: return "\"Weights\"";
Parser.C:1465:   case TT_WEIGHTS:
```

#### #3 - 06 Mar 2015 11:10 - John Abbott

This is trickier than I originally thought.

The way the CoCoA-5 interpreter is designed a keyword is always seen as that keyword even it would make no sense in that context. Many computer languages work this way (FORTRAN is the only exception I know of), so it is probably a good idea.

In principle we could allow records to contain fields whose names are CoCoA-5 keywords; indeed PreprocessPoints does exactly that. Modifying the indexing function would not be hard: we could make rec["weights"] work as expected; it would be harder to make rec.weights work as expected since that would break the rule of a keyword always being seen as a keyword. However, if we were to do that then a printed record containing a field with name weights could not be cut-and-pasted as input -- for that to work we would have to modify parsing of a record which looks to be rather tricky (and would further break the rule about a keyword always being seen as a keyword).

Nevertheless I do find it unnatural/inconvenient that fields in a record are "arbitrarily" forbidden from having certain names.

I suppose the cleanest solution is to stick with the rule that keywords can never be used as identifiers in any context. It has the advantage that it is fairly easy to remember (assuming you have a list of keywords available), and very easy to implement (no need to change the existing code).

I note that the same limitation applies to tags in tagged values; this is probably unnecessary as a tag only ever appears as a string (right?).

I shall try to improve the error message when executing `rec["weights"]` which is unhelpful in its current form.

#### **#4 - 06 Mar 2015 11:21 - John Abbott**

I should also modify `RECORD::setField` so that it gives an error if the field name is not a valid identifier for CoCoA-5. This ought to be easy.

Hint: `Interpreter.C:2276--2288`

Fixing the error messages, as suggested in my previous post, looks to be rather trickier than I had hoped :-)

#### **#5 - 06 Mar 2015 20:31 - John Abbott**

- % Done changed from 20 to 30

After speaking to Anna by Skype we have decided:

- Anna will remove the keyword weights (from interpreter and UIs)
- John will attempt to complete the revision of error messages

#### **#6 - 10 Mar 2015 20:13 - John Abbott**

In `Interpreter.H:718` the mem fn `setField` is declared as virtual; why??

#### **#7 - 10 Mar 2015 21:09 - John Abbott**

- % Done changed from 30 to 50

At the moment I have implemented `setField` and `setFieldNoCheck`; the latter is just the old fn renamed. The new fn needs source code location info in case an error occurs.

I have changed `Interpreter.C`, `BuiltinFunctions.C` and `BuiltinFunctions-Normaliz.C` but found I was undecided whether to call `setField` or `setFieldNoCheck`. In the end I used the simple rule: if the field name was a string literal then I called `setFieldNoCheck`, otherwise I called `setField` (so the field name would be checked at run time).

I'm not sure this is the most sensible strategy -- it was quick to effect :-)

#### **#8 - 11 Mar 2015 13:07 - John Abbott**

It works... more or less. `A["weights"]` gives the error I wanted, but `A.weights` still gives a less helpful error. To rectify this requires changing `Parser.H:158`; perhaps I'll add a new issue for this as it will probably take some time to do.

**#9 - 11 May 2015 14:21 - John Abbott**

- *Target version changed from CoCoA-5.?.? to CoCoA-5.1.2 summer 2015*

**#10 - 26 Jun 2015 12:16 - Anna Maria Bigatti**

- *Status changed from In Progress to Feedback*

- *% Done changed from 50 to 90*

seems effectively done: close this issue?

**#11 - 27 Jul 2015 15:16 - Anna Maria Bigatti**

- *Status changed from Feedback to Closed*

- *% Done changed from 90 to 100*

- *Estimated time set to 4.01 h*