CoCoA-5 - Bug #1717

CRASH: equality test to a FUNCTION

01 Dec 2022 11:43 - John Abbott

Status:	Closed	**Start date:**	01 Dec 2022
Priority:	Urgent	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	bug	**Estimated time:**	1.11 hour
Target version:	CoCoA-5.4.2	**Spent time:**	1.15 hour

Description

Julian Danner reported the following crash via email:

```
/**/  use R::=QQ[x]; phi := CoeffEmbeddingHom(R); phi = func() endfunc;
CoCoAInterpreter: Interpreter.C:1581: boost::intrusive_ptr<CoCoA::InterpreterNS::RightValue> CoCoA
::InterpreterNS::RuntimeEnvironment::binaryOperatorDispatch(boost::intrusive_ptr<const CoCoA::Inte
rpreterNS::RightValue>, boost::intrusive_ptr<const CoCoA::InterpreterNS::RightValue>, boost::intru
sive_ptr<CoCoA::InterpreterNS::RightValue> (* (*)[24])(CoCoA::InterpreterNS::RuntimeEnvironment*,
boost::intrusive_ptr<const CoCoA::InterpreterNS::RightValue>, boost::intrusive_ptr<const CoCoA::In
terpreterNS::RightValue>, const CoCoA::LexerNS::CharPointer&, const CoCoA::LexerNS::CharPointer&),
 const CoCoA::LexerNS::CharPointer&, const CoCoA::LexerNS::CharPointer&): Assertion `leftOp->getTy
pe()->dispatchIndex == rightIndex' failed.

Process cocoa5 aborted (core dumped)
```

Ouch! It was originally a typo **:=** mistyped as **=**

History

#1 - 01 Dec 2022 11:45 - John Abbott

Most likely the bug is in **intrusive_ptr<RightValue> RuntimeEnvironment::binaryOperatorDispatch**
around line 1520 in Interpreter.C

#2 - 01 Dec 2022 11:51 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The relevant assert is at Interpreter.C:1581

A slightly shorter failing case is:

```
func() endfunc = IdentityHom(ZZ);
```

In many cases CoCoA detects that FUNCTION cannot be compared to the other type, but RINGHOM is not so reported.
Odd!

#3 - 01 Dec 2022 12:17 - John Abbott

- Status changed from In Progress to Resolved

- Assignee set to John Abbott

- % Done changed from 10 to 70

Here is what I found.
The decision about which types may be compared with **=** is determined by **opEqualMap**.
This had a fn pointer to **opEqual_Function_Function** which just threw an exception ("Functions cannot be compared").

However if there is no entry in OpEqualMap for (FUNCTION,FUNCTION) then essentially the same error mesg is produced ("I don't know how to ...").

Note that RINGHOM is regarded as a subtype of FUNCTION via function **bool TYPE::isProperSubtypeOf** around Interpreter.C:1467. I am loath to change that function for fear that it might break something *e.g.* when using a RINGHOM in a "function call context".

Instead I have removed the (FUNCTION,FUNCTION) entry from opEqualMap and from opNotEqualMap.

All CoCoA-5 tests pass.

Now CoCoA-5 reacts as follows:

```
/**/ func() endfunc = IdentityHom(ZZ);
--> ERROR: I don't know how to evaluate operator = between FUNCTION and RINGHOM
--> func() endfunc = IdentityHom(ZZ);
-->                  ^
```

I think this resolves the issue. Comments? Criticisms?

#4 - 14 Dec 2022 16:23 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 70 to 100

- Estimated time set to 1.11 h

We are satisfied that this is a good resolution.