CoCoA-5 - Feature #245

Meaningful error for functions defined in "missing" external library

01 Oct 2012 10:19 - Anna Maria Bigatti

Status:	Closed	**Start date:**	01 Oct 2012
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	Portability	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.1.1 Seoul14	**Spent time:**	6.00 hours

Description

Right now if an external library, say, Normaliz is not linked in, all its cocoa-5 functions are not defined.
This means that a user would get the error message "cannot find ... in scope".
A message like "External Library Normaliz is not linked in" would be much better!

History

#1 - 01 Oct 2012 11:13 - John Abbott

Initially I was unconvinced, but then realised that an error message is necessary, and naturally the error message should be as helpful as possible. ***Cannot find ... in scope*** is surely less clear than ***Function unavailable because CoCoA was not built with Normaliz*** (exact wording still to be decided)

We do not make arity checks for unavailable functions: it is surely more useful to the user (and simpler for us) to report *unavailable* rather than *wrong arity*.

#2 - 01 Oct 2012 11:28 - John Abbott

Here is a reasonable proposal for how we could implement the ideas above.
We define a CPP macro called DECLARE_MISSING_EXTLIB which registers the name as a function with any arity, and which always produces a *Missing external library* error.

Here is how it might look in the file BuiltinFunctions-Normaliz.C

```
#ifndef CoCoA_WITH_NORMALIZ

  DECLARE_MISSING_EXTLIB(NormalizHilbertBasis,    "NORMALIZ");
  DECLARE_MISSING_EXTLIB(NormalizNormalToricRing, "NORMALIZ");
  DECLARE_MISSING_EXTLIB(NormalizHilbertBasis,    "NORMALIZ");
  DECLARE_MISSING_EXTLIB(NormalizHilbertBasis,    "NORMALIZ");

#else

DECLARE_STD_BUILTIN_FUNCTION(NormalizHilbertBasis, 1) {...}
```

The main defect is that the name of each function has to be written twice: once when using the DECLARE_MISSING_EXTLIB macro, and once when giving the proper definition. However, I think the block of calls to DECLARE_MISSING_EXTLIB could also function as a handy index of the functions defined in the file.

#3 - 02 Apr 2014 17:34 - Anna Maria Bigatti

- Target version set to CoCoA-5.1.0 Easter14

#4 - 04 Apr 2014 15:04 - John Abbott

- Category set to Portability

- Status changed from New to In Progress

- Assignee set to Anna Maria Bigatti

- % Done changed from 0 to 10

Here's a possible defn for DECLARE_MISSING_EXTLIB

```
#define DECLARE_MISSING_EXTLIB(FnName, ExtLibName)
DECLARE_BUILTIN_FUNCTION(FnName)
{
  throw MissingExtLibException(#FnName " not defined because CoCoA was built without external library " #ExtLibName);
}
```

#5 - 09 Apr 2014 17:37 - John Abbott

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

#6 - 02 Sep 2014 17:46 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

- % Done changed from 10 to 80

I applied John's suggestion.
I'd rather find an "automatic" solution (so that we don't risk forgetting or misspelling a function) but it works fine for the time being: we don't add/rename functions so frequently.

#7 - 11 Sep 2014 17:40 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100