CoCoALib - Design #1523

Input fns: action when when istream is in bad state?

26 Oct 2020 14:31 - John Abbott

Status:	Closed	**Start date:**	26 Oct 2020
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Safety	**Estimated time:**	2.22 hours
Target version:	CoCoALib-0.99800	**Spent time:**	2.25 hours

Description

Output fns in CoCoALib check whether the ostream is in a bad state, and if so, return without doing anything more.

What should input fns do? Should they check if the istream is good and then simply return?

Related issues:

Related to CoCoA-5 - Feature #1509: RingElems with empty input	**Closed**	**14 Oct 2020**
Related to CoCoALib - Design #1529: INPUT questions	**Closed**	**31 Oct 2020**

History

#1 - 26 Oct 2020 14:32 - John Abbott

- Related to Feature #1509: RingElems with empty input added

#2 - 26 Oct 2020 14:35 - John Abbott

At the moment my preference is to recommend (strongly) that input fns immediately check whether the istream is not good, and if so, simply return (leaving the istream in its original state).
An advantage is that this may be a bit faster than doing any preprocessing, then attempting to read something and discovering after the attempt that it failed. It also "feels cleaner" to me to make the check right at the start.

Note that an istream where the last char has been read but is not yet EOF, will not be in a bad state even though any further reads will fail (and put it into an EOF state).

#3 - 26 Oct 2020 19:41 - John Abbott

If we do follow this recommendation, which files must be considered?

- RingElemInput
- BigInt and BigRat?
- symbol

#4 - 30 Oct 2020 19:24 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 30

After discussion with Anna, we have decided that it is more helpful if an exception is thrown if the istream is not in a good state when an input fn is called. This applies only to publicly callable fns.

#5 - 30 Oct 2020 19:46 - John Abbott

- Status changed from In Progress to Resolved

- Assignee set to John Abbott

- % Done changed from 30 to 70

I have added checks to all input fns so that an exception is thrown if the istream is not in a good state when called.

I have tried to clean some fns in RingElemInput.C so that WhatsNext is not called when EOF has already been encountered (not certain I've covered all possibilities).

#6 - 30 Oct 2020 19:54 - John Abbott

The function symbol::myInput reports an error by setting the channel state (failbit) rather than throwing an exception.
So it more or less follows the C++ way; however it leaves the "read cursor" at the position where it found a problem even if it would be possible to rewind to a state where reading had been successful *e.g.* with input x[1 the entire input is read and an error signalled, even though one could say that x is a valid symbol, and [1 is just following junk.

Should it throw an exception?

#7 - 30 Oct 2020 20:20 - John Abbott

Checked in, but not symbol.C.

#8 - 31 Oct 2020 09:05 - John Abbott

- Related to Design #1529: INPUT questions added

#9 - 04 Dec 2020 23:19 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 70 to 90

- Estimated time set to 2.22 h

I have also modified and checked in symbol.C

Conclusion: input fns throw if istream is not in good state

Also **input fns throw if input is not valid**. Throwing gives more info than simply putting the stream into a bad state.

If input succeeds then istream is left in a good state (even if EOF was hit -- the EOF flag is cleared before returning).

#10 - 17 Feb 2021 11:01 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

Closing after 2 months in feedback.

One last change: input fns throw if istream is not in decimal mode