“My Little Project with CoCoALib”

Anna M. Bigatti, Genova, Italy

Kassel, October 2021

Do you have a little project to do with CoCoA/CoCoALib?

- an algorithm in a paper
- a mathematical conjecture
- a new idea
- .. or just your homework ;-)

Example 1

Today “My little project” is this:

Implement (my own) Buchberger algorithm in CoCoALib

- Already implemented it in CoCoA-5
- My work today it to translate it into CoCoALib.

First steps on the project

1. Start with a sketch on **PAPER**!
2. Compute easy examples by hand, following the algorithm
3. Create examples by “reverse engineering” (see *SystemSolving*)
4. Identify the important objects and steps

Example 2

Design phase for my project already done😀

- **CoCoA-5**: input `IDEAL`, output `LIST of RINGELEM`.
  CoCoALib: input `ideal`, output `vector<RingElem>`.
- Functions:
  - ReducerIndex
  - NormalRemainderLPP
  - SPoly
  - MyBGasis

First steps on the computer

1. Start by a meaningful example “step-by-step”
2. **Understand** which **types** you need:
   e.g. BigInt, BigRat, ring, RingElem, ideal, list/vector, **..**
   (You probably will not need to make a new class)
3. **Understand** which **functions** you need
4. Make a **simple prototype** (in CoCoA-5 or CoCoALib)

Example 3

1. – Done in CoCoA-5 –
2. ring, RingElem, ideal, `vector<RingElem>`, `pair`, **...**
3.
   - ReducerIndex: LPP, IsDivisible, ...
   - NormalRemainderLPP: LM, - * / for polynomials, ...
4. we already have an excellent prototype in CoCoA-5!

On paper work “**top-down**”:

- start with the big picture
- and identify the “logical blocks”

On computer work “**bottom-up**”:

- First the basic functions (“logical blocks”) – *and test each of them*
- then implement the functions calling them – *and test each of them*

Print, print print!! (Use “verbosity”)

Example 4

- And now we see `ex-MyGBasis.C` coming alive! (bottom-up)