

CoCoALib

GPL C++ Library



<http://cocoa.dima.unige.it/>

John Abbott & Anna Bigatti

Università di Genova, Italy

(1) **Where to start from?**

download and compile!

(2) **[compilation in real time]**

(3) **Philosophy**

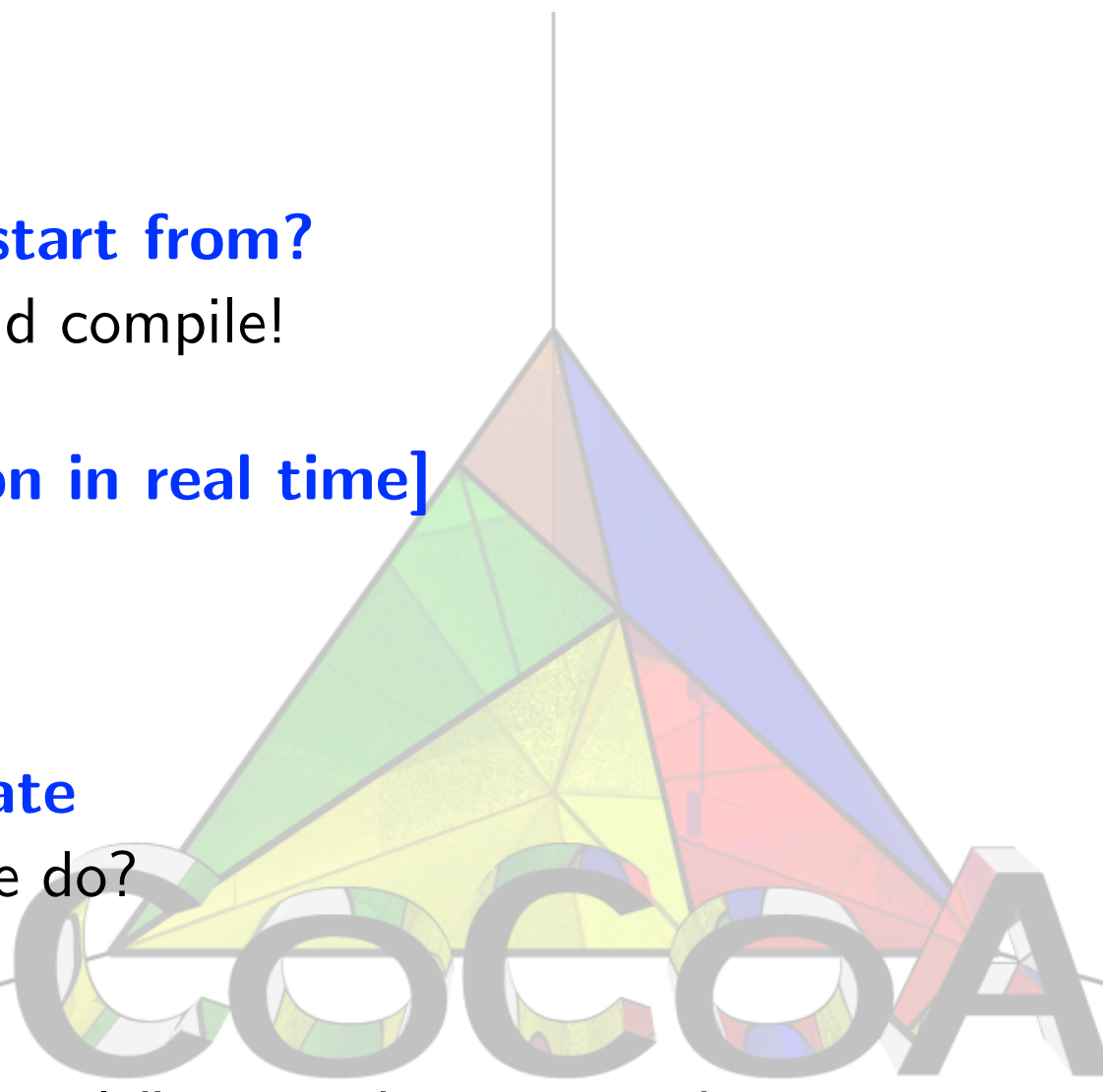
(4) **Current state**

what can we do?

(5) **Examples**

a good “example” is worth 1000 words

(6) **Design**





(1) Where to start from?

- **Write an email to cocoa@dima.unige.it**
“I want to be an alpha tester” → secret page
- **Download CoCoALib**
current version: `cocoalib-0.94`
- **Configure and compile**
`./configure; make`



(2) Let's start!

```
./configure; make
```



(3) Philosophy

Inline functions and inheritance \implies
simplicity, flexibility, and speed.

- **For all users** natural syntax with extensive checking.
- **For experienced users** alternative syntax for faster unchecked operations.
- **For developers** there are several debugging aids.



(4) Current state

- Accessible via prototype **server** from **CoCoA 4**.
- Types for representing **polynomial rings, ideals and submodules**.
- Computes **Gröbner bases** for ideals and modules.
- Offers the **coefficient rings** \mathbb{Q} , \mathbb{F}_p , \mathbb{R} , and $k(a)$; algebraic extensions, and multivariate rational functions due shortly.
- Uses **ring homomorphisms** for transporting values from one ring to another.



(5) Examples

Some examples focus on a class and show a long list of functions.

E.g. from `ex-RingElem1.C`:

```
GlobalOutput() << "a == b gives      " << (a == b) << endl;
GlobalOutput() << "a * b gives      " << a*b << endl;
GlobalOutput() << "-a          gives      " << -a << endl;
GlobalOutput() << "a += b gives a = " << (a += b) << endl;
GlobalOutput() << "IsZero(one)      gives " << IsZero(one) << endl;
GlobalOutput() << "IsOne(one)       gives " << IsOne(one) << endl;
GlobalOutput() << "IsMinusOne(one) gives " << IsMinusOne(one) << endl;
GlobalOutput() << "power(a, 10)    gives " << power(a, 10) << endl;
```

Other examples give “pieces of code” explaining particular functions, e.g. `ex-PolyRing1.C` or workarounds for missing or uncomplete aspects, e.g. `ex-NF.C`