`LocalizeRingForHomalg`: Localize Commutative Rings at Maximal Ideals

Mohamed Barakat & Markus Lange-Hegermann

TU-Kaiserslautern & RWTH-Aachen

ISSAC, 26.07.2010

Software for *homological* computations in the category of finitely presented modules over a *localized ring* $R_\mathfrak{m}$.

Computability of ABELian Categories

Definition

We call an ABELian categoriy **computable**, if all existential quantifiers appearing in the axioms can be turned into *constructive* ones.

Example axiom: For any morphism there *exists* a kernel.

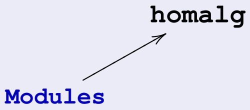
The homalg Project

The idea: An abstract homological algebra meta-package for computable ABELian categories

homalg

The homalg Project

The category of finitely presented modules as the basic example of a computable ABELian category



Computable Rings

- A ring R is called **computable**, if there exist algorithms for *solving linear systems*[1] over R.

[1] $\{x \in R^{n\times 1} \,|\, Ax = b\}$ and $\{x \in R^{1\times m} \,|\, xA = c\}$ for $A \in R^{m\times n}, b \in R^{m\times 1}, c \in R^{1\times n}$

Computable Rings

- A ring R is called **computable**, if there exist algorithms for *solving linear systems*[1] over R.
- $K[x_1, \ldots, x_n]$ is computable (BUCHBERGER's Algorithm).
  $K[x_1, \ldots, x_n]_{\langle x_1, \ldots, x_n \rangle}$ is computable (MORA's Algorithm).
  And probably *your* favorite ring is computable.

[1] $\{x \in R^{n \times 1} \mid Ax = b\}$ and $\{x \in R^{1 \times m} \mid xA = c\}$ for $A \in R^{m \times n}, b \in R^{m \times 1}, c \in R^{1 \times n}$

Computable Rings

- A ring R is called **computable**, if there exist algorithms for *solving linear systems*[1] over R.
- $K[x_1, \dots, x_n]$ is computable (BUCHBERGER's Algorithm). $K[x_1, \dots, x_n]_{\langle x_1,\dots,x_n \rangle}$ is computable (MORA's Algorithm). And probably *your* favorite ring is computable.

[BLH, Theorem 3.4]

Let R be computable. Then the ABELian category of finitely presented R-modules is computable.

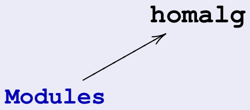
"Proof": Use Matrices

($\Box$)

[1] $\{x \in R^{n\times 1} \mid Ax = b\}$ and $\{x \in R^{1\times m} \mid xA = c\}$ for $A \in R^{m\times n}, b \in R^{m\times 1}, c \in R^{1\times n}$

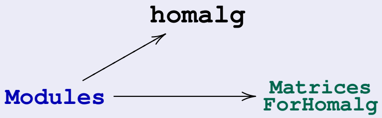
The homalg Project

The category of finitely presented modules as the basic example of a computable ABELian category



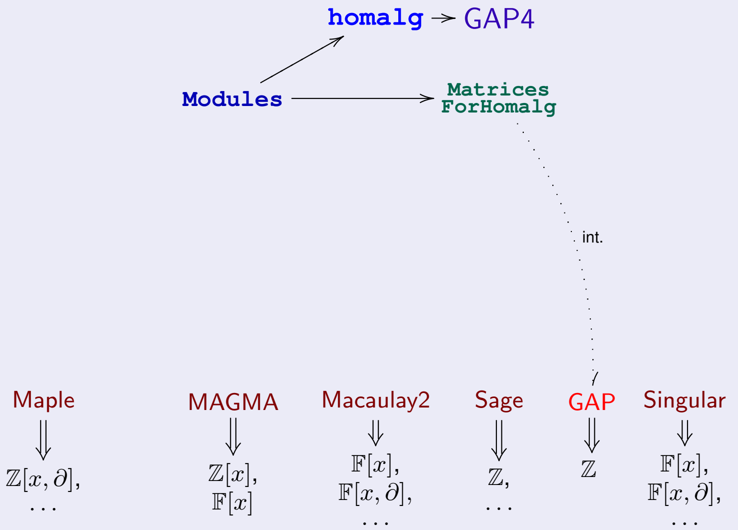
The homalg Project

Matrices provide the needed data structure for finitely presented modules and their morphisms

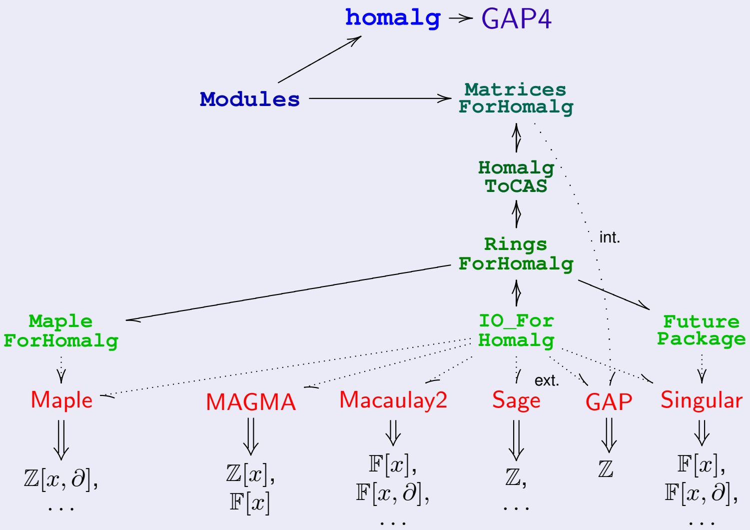


The homalg Project

homalg: GAP has a superb language for mathmatical programming but "sufficiently supports" only the ring of integers

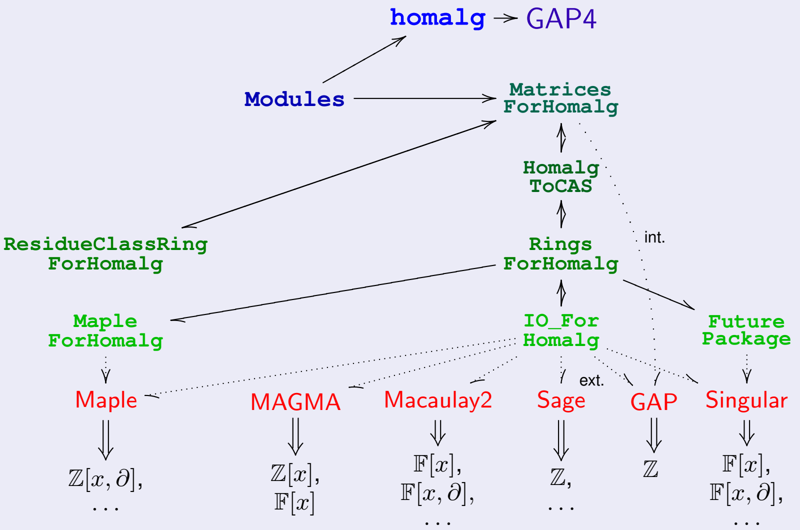


External CASs host the matrices and GAP4 contains the higher logic → **Principle of least communication**



The homalg Project

`ResidueClassRingForHomalg`: Create a residue class ring of a ring modulo an ideal.



Localized Computable Rings

[BLH, Theorem 4.1]

Let R be a computable, commutative ring and $\mathfrak{m}$ a finitely generated maximal ideal in R. Then $R_{\mathfrak{m}} := (R \backslash \mathfrak{m})^{-1} R$ is **computable**.

Localized Computable Rings

[BLH, Theorem 4.1]

Let R be a computable, commutative ring and $\mathfrak{m}$ a finitely generated maximal ideal in R. Then $R_{\mathfrak{m}} := (R \setminus \mathfrak{m})^{-1} R$ is **computable**.

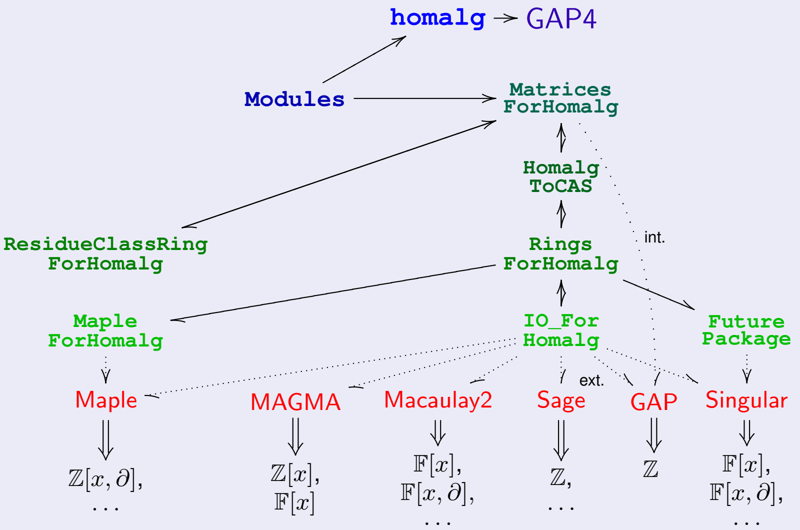
"Proof":

Homogeneous system: $\text{Syzygies}_{R_{\mathfrak{m}}}(\,\cdot\,) = R_{\mathfrak{m}} \otimes_R \text{Syzygies}_R(\,\cdot\,)$

Particular solution: The submodule membership problem of $R_{\mathfrak{m}}$ reduces to the submodule membership problem of R.

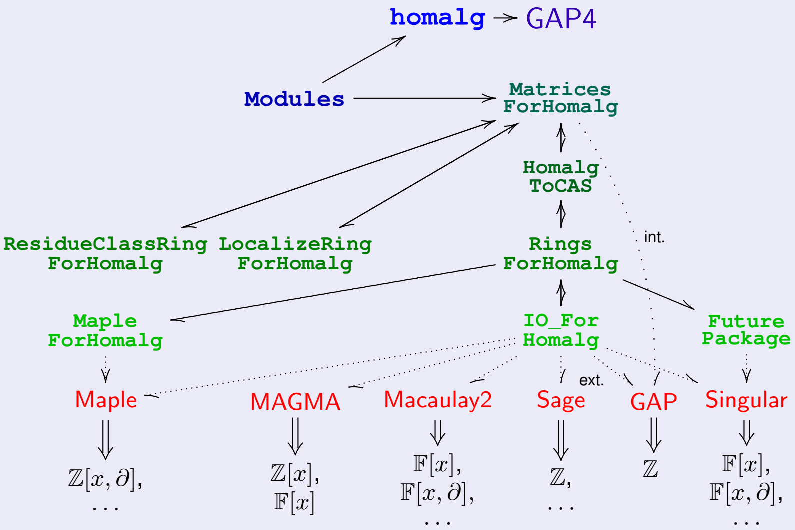
($\square$)

ResidueClassRingForHomalg: Create a residue class ring of a ring modulo an ideal.



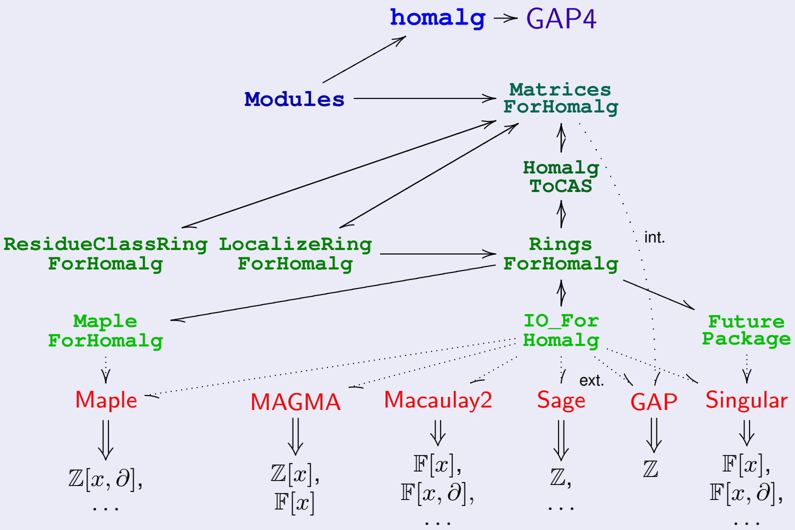
The homalg Project

LocalizeRingForHomalg: Localizations of commutative rings in homalg at maximal ideals.



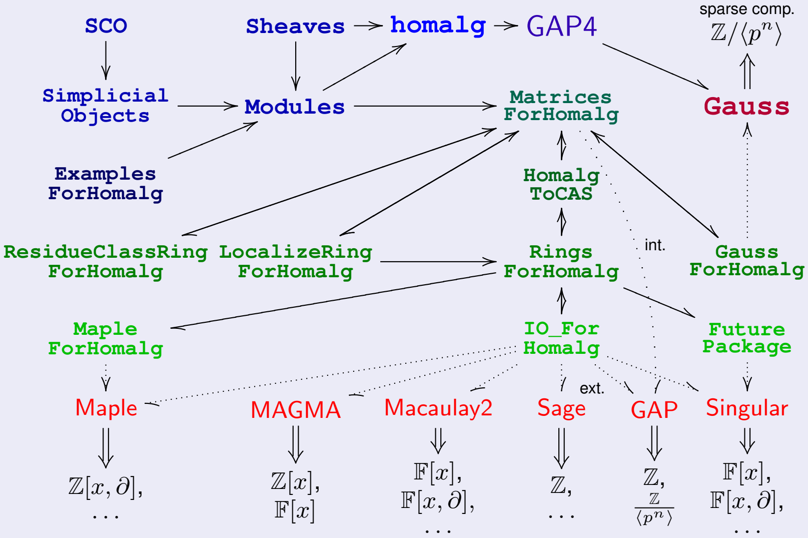
The homalg Project

LocalizeRingForHomalg: Use MORA's algorithm in Singular to localize polynomial rings at maximal ideals.



The homalg Project

Advanced applications building upon homalg



Example: SERRE's Intersection Formula

SERRE's formula of intersection multiplicity for two ideals $I, J \lhd R$ at a prime ideal $\mathfrak{p} \lhd R$:

$$i(I, J; \mathfrak{p}) = \sum_j (-1)^j \operatorname{length}\left(\operatorname{Tor}_j^{R_\mathfrak{p}}(R_\mathfrak{p}/I_\mathfrak{p}, R_\mathfrak{p}/J_\mathfrak{p}) \right)$$

Let $R := \mathbb{F}_5[x, y, z, v, w]$, $\mathfrak{p} = \mathfrak{m} = \langle x, y, z, v, w \rangle \lhd R$ maximal ideal and $R_0 = S_0 := R_\mathfrak{m}$.

Example: SERRE's Intersection Formula

SERRE's formula of intersection multiplicity for two ideals $I, J \lhd R$ at a prime ideal $\mathfrak{p} \lhd R$:

$$i(I, J; \mathfrak{p}) = \sum_j (-1)^j \underbrace{\text{length}}_{\text{MORA}} \left(\underbrace{\text{Tor}_j^{R_\mathfrak{p}}(R_\mathfrak{p}/I_\mathfrak{p}, R_\mathfrak{p}/J_\mathfrak{p})}_{\text{MORA}} \right)$$

Let $R := \mathbb{F}_5[x, y, z, v, w]$, $\mathfrak{p} = \mathfrak{m} = \langle x, y, z, v, w \rangle \lhd R$ maximal ideal and $R_0 = S_0 := R_\mathfrak{m}$.

Example: SERRE's Intersection Formula

SERRE's formula of intersection multiplicity for two ideals $I, J \lhd R$ at a prime ideal $\mathfrak{p} \lhd R$:

$$i(I,J;\mathfrak{p}) = \sum_j (-1)^j \underbrace{\mathrm{length}}_{\text{MORA}} \left(\underbrace{\mathrm{Tor}_j^{R_\mathfrak{p}}(R_\mathfrak{p}/I_\mathfrak{p}, R_\mathfrak{p}/J_\mathfrak{p})}_{\text{~~MORA~~}} \right)$$

Let $R := \mathbb{F}_5[x,y,z,v,w]$, $\mathfrak{p} = \mathfrak{m} = \langle x,y,z,v,w \rangle \lhd R$ maximal ideal and $R_0 = S_0 := R_\mathfrak{m}$.

Example: SERRE's Intersection Formula

SERRE's formula of intersection multiplicity for two ideals $I, J \lhd R$ at a prime ideal $\mathfrak{p} \lhd R$:

$$i(I,J;\mathfrak{p}) = \sum_j (-1)^j \underbrace{\mathrm{length}}_{\text{MORA}} \left(\underbrace{\mathrm{Tor}_j^{R_\mathfrak{p}}(R_\mathfrak{p}/I_\mathfrak{p}, R_\mathfrak{p}/J_\mathfrak{p})}_{\text{BUCHBERGER}} \right)$$

Let $R := \mathbb{F}_5[x,y,z,v,w]$, $\mathfrak{p} = \mathfrak{m} = \langle x,y,z,v,w \rangle \lhd R$ maximal ideal and $R_0 = S_0 := R_\mathfrak{m}$.

Main advantages...

... of the `homalg` project

One implementation

for all categories.

Main advantages...

... of the `homalg` project

- Abstract homological algebra allows **one implementation** for (*derivation* and *composition* of) *functors*, *filtrations*, *spectral sequences*, etc. being applicable **for all categories**.

Main advantages...

... of the homalg project

- Abstract homological algebra allows **one implementation** for (*derivation* and *composition* of) *functors*, *filtrations*, *spectral sequences*, etc. being applicable **for all categories**.
- Performance strongly benefits from logic in GAP.

Main advantages...

... of the `homalg` project

- Abstract homological algebra allows **one implementation** for (*derivation* and *composition* of) *functors*, *filtrations*, *spectral sequences*, etc. being applicable **for all categories**.
- Performance strongly benefits from logic in GAP.

... of `LocalizeRingForHomalg`

- Universal: Compatible with *any* commutative computable ring R.

Main advantages...

... of the `homalg` project

- Abstract homological algebra allows **one implementation** for (*derivation* and *composition* of) *functors*, *filtrations*, *spectral sequences*, etc. being applicable **for all categories**.
- Performance strongly benefits from logic in GAP.

... of `LocalizeRingForHomalg`

- Universal: Compatible with *any* commutative computable ring R.
- No (expensive) local computations in $R_{\mathfrak{m}}$.
  Compute in R, which usually has highly optimized algorithms.

References

`LocalizeRingForHomalg` is deposited in GAP.

- Mohamed Barakat and Markus Lange-Hegermann, *An Axiomatic Setup for Algorithmic Homological Algebra and an Alternative Approach to Localization*, to appear in Journal of Algebra and its Applications (`arXiv:1003.1943`).
- The homalg project authors, *The* `homalg` *project*, 2003-2010, (`http://homalg.math.rwth-aachen.de/`).

Thank you for your attention.