How to Parallelize SINGULAR–Code for Multiple Core Machines

Stefan Steidel

Technische Universität Kaiserslautern

ITWM Kaiserslautern, July 15, 2010

First Steps for Parallelization in SINGULAR

Technical Facts

- Parallelization just possible on SINGULAR library level (interpreter language) – not yet in the kernel
- via MP-links

First Steps for Parallelization in SINGULAR

Technical Facts

- Parallelization just possible on SINGULAR library level (interpreter language) – not yet in the kernel
- via MP-links

Applications – Modular Algorithms

- modular computation of Gröbner bases over $\mathbb{Q}$ (cf. `modstd.lib`)
- modular computation of associated primes of a zero–dimensional ideal over $\mathbb{Q}$ (cf. `assprime.lib`)
- primary decomposition over $\mathbb{Z}$ (to appear)

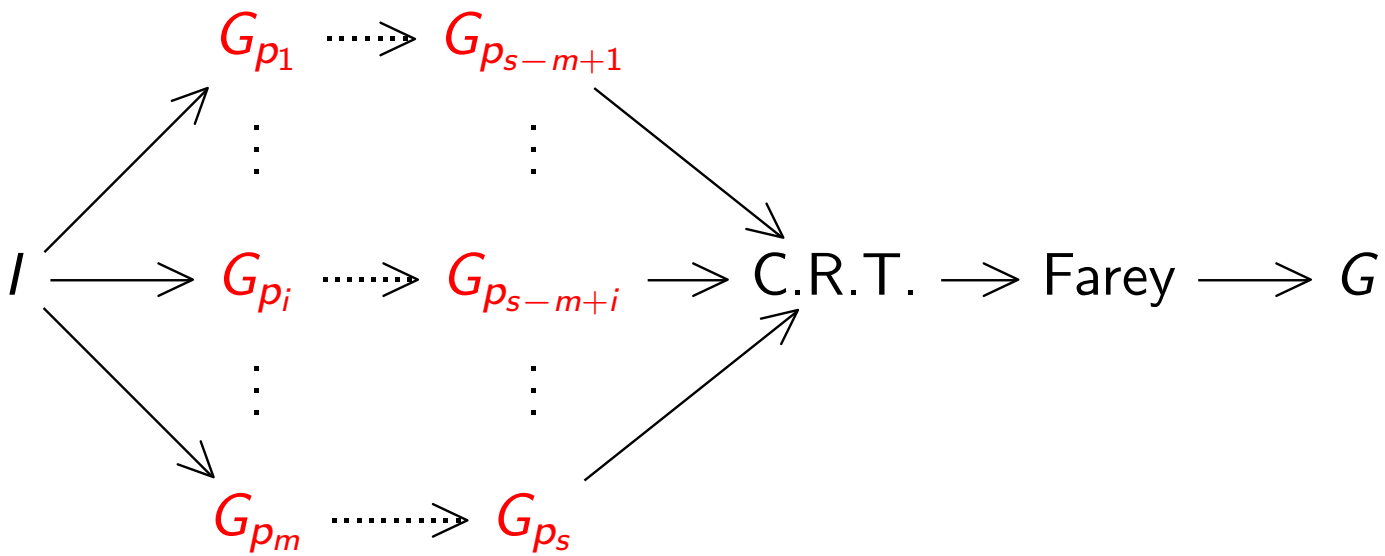
Parallelized Modular Gröbner Basis Computation

Idea of `modStd`:

$$I \longrightarrow G_{p_1} \dashrightarrow G_{p_i} \dashrightarrow G_{p_s} \longrightarrow \text{C.R.T.} \longrightarrow \text{Farey} \longrightarrow G$$

Parallelized Modular Gröbner Basis Computation

Idea of parallelized `modStd`:



Parallelized Modular Gröbner Basis Computation

Details about the implementation of the procedure `modStd` can be found on the handout resp. in the SINGULAR library `modstd.lib`.

Parallelized Modular Gröbner Basis Computation

Example:

Compute the Gröbner basis of $\texttt{cyclic}(8) \subseteq \mathbb{Q}[x_1, \ldots, x_8]$ with monomial ordering $>_{\texttt{dp}}$.

Parallelized Modular Gröbner Basis Computation

Example:

Compute the Gröbner basis of $\texttt{cyclic}(8) \subseteq \mathbb{Q}[x_1, \ldots, x_8]$ with monomial ordering $>_{\texttt{dp}}$.

Timings are conducted by using the 32-bit version of SINGULAR 3-1-1 on an Intel® Xeon® X5460 with 4 CPUs, 3.16 GHz each, 64 GB RAM under the Gentoo Linux operating system.

Parallelized Modular Gröbner Basis Computation

Example:

Compute the Gröbner basis of $\texttt{cyclic}(8) \subseteq \mathbb{Q}[x_1, \ldots, x_8]$ with monomial ordering $>_{\texttt{dp}}$.

Timings are conducted by using the 32-bit version of SINGULAR 3-1-1 on an Intel® Xeon® X5460 with 4 CPUs, 3.16 GHz each, 64 GB RAM under the Gentoo Linux operating system.

- using `std` (basic Gröbner basis algorithm in SINGULAR): *killed after more than 2 days because of memory*

Parallelized Modular Gröbner Basis Computation

Example:

Compute the Gröbner basis of $\texttt{cyclic}(8) \subseteq \mathbb{Q}[x_1, \ldots, x_8]$ with monomial ordering $>_{\texttt{dp}}$.

Timings are conducted by using the 32-bit version of SINGULAR 3-1-1 on an Intel® Xeon® X5460 with 4 CPUs, 3.16 GHz each, 64 GB RAM under the Gentoo Linux operating system.

- using `std` (basic Gröbner basis algorithm in SINGULAR):
  killed after more than 2 days because of memory
- using `modStd`:
  4652 sec

Parallelized Modular Gröbner Basis Computation

Example:

Compute the Gröbner basis of $\texttt{cyclic}(8) \subseteq \mathbb{Q}[x_1, \ldots, x_8]$ with monomial ordering $>_{\texttt{dp}}$.

Timings are conducted by using the 32-bit version of SINGULAR 3-1-1 on an Intel® Xeon® X5460 with 4 CPUs, 3.16 GHz each, 64 GB RAM under the Gentoo Linux operating system.

- using `std` (basic Gröbner basis algorithm in SINGULAR):
  killed after more than 2 days because of memory
- using `modStd`:
  4652 sec
- using parallelized `modStd`:
  2692 sec

Problems and Outlook

- Parallelization is just applicable using 32–bit SINGULAR version since MP–links are not compatible with 64–bit SINGULAR version.

Problems and Outlook

- Parallelization is just applicable using 32–bit SINGULAR version since MP–links are not compatible with 64–bit SINGULAR version.
- MP–links are not compatible with polynomial ring over the integers.

Problems and Outlook

- Parallelization is just applicable using 32–bit SINGULAR version since MP–links are not compatible with 64–bit SINGULAR version.
- MP–links are not compatible with polynomial ring over the integers.
- Enable parallelization in the kernel.