

Sage Days 23.5: Sage & Singular

William Stein

14. Juli 2010

Goal for this meeting

Improve interaction between the Sage and Singular groups

- Motivation: by working together we can accomplish much, much more than by competing with each other.
Cocoa vs Macaulay vs Singular vs Giac, and the winner is... commercial software!
- Anything the Sage project can do to strengthen Singular is a win for us.

Why is Singular in Sage instead of Macaulay2 or Cocoa? Answer: *In 2005, Singular was much easier to build from source than Macaulay2, and Cocoa released their first open source version way too late.*

Goals for Sage developers:

- Learn in detail how Singular is really developed (a social question)
- Understand the goals and roadmap of the Singular project (recent grant proposal?)
- Understand the actual Singular codebase: **I will get a patch into Singular!**

Goal for Singular developers:

- Learn in detail how Sage is really developed (again, a social question)
- Understand the goals and roadmap of the Sage project.
- Better understand the actual Sage codebase: **You will get a patch into Sage!**

How Sage is developed

- Very distributed. Large number of people involved. Most developers have never met each other. Over 400 trac accounts.
- Technology: trac + mailing lists + irc + rotating release manager + testing
- Low tech still: Plain text patches get posted to trac, refereed, then merged into sage manually, though better approaches are implemented.
- spkg: Sage packages.
- See Robert Miller's talk for more details

The Goal of the Sage project

Create a viable free open source alternative to Magma, Maple, Mathematica, and Matlab

(Their expensive proprietary approach is evil, and I'm sick of it.)

A viable alternative will have:

- fast Groebner basis, polynomial factorization, function fields
- flashy 2d/3d graphics
- a notebook interface; an IDE
- sophisticated symbolic integration, special functions, etc.
- commercial support (including customized notebook servers)
- full spectrum of undergrad curriculum (see Zimmerman et al.!).

Roadmap of Sage project

- Sage-5.0 (August 31, 2010)
- Sage-6.0 (2011)
- Sage-7.0 (2012)

Roadmap of Sage project

- Sage-5.0 (August 31, 2010)
 - Windows port via Cygwin
 - Upgrade PARI to the latest SVN version
 - Upgrade MPIR to version 2.x
 - Get doctest coverage to 90% (currently at 84.6%; we need 1327 more functions)
- Sage-6.0 (2011)
- Sage-7.0 (2012)

Roadmap of Sage project

- Sage-5.0 (August 31, 2010).
- Sage-6.0 (2011), some potential goals:
 - Function fields (Hess's algorithms, 2-descent, L-functions)
 - Special functions (mpmath, symbolic)
 - Much improved Sage notebook (scalability, customizability)
 - Textbooks, interacts, etc., integrated with the sage distribution
 - Commercial support (custom notebook servers)
 - Fully switch to using C library interfaces for GAP and Maxima.
 - Get doctest coverage to 100%
- Sage-7.0 (2012)

Roadmap of Sage project

- Sage-5.0 (August 31, 2010).
- Sage-6.0 (2011):
- Sage-7.0 (2012), some potential goals:
 - Fast Groebner basis computation that is competitive with Magma/Maple
 - Vast improvements in Sage for Science and Engineering (documentation, diff'eq, data workflows, reproducible research, instrument support, data formats like HD5)
 - Statistics: something Pythonic/Cythonic and built on top of R + scipy.stats + GSL, which competes with SAS, etc.
 - Switch to Python 3.x
 - Much randomized and unit testing that goes beyond doctesting

... **World Domination**

Any questions?