

Sage, Pan-Axiom and Aldor

<http://www.sagemath.org>

Michael Abshoff¹

¹Department of Mathematics
Technical University of Dortmund, Germany

RISC, Hagenberg, 2008-07-24



- 1** Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom
- 3 Sage and Common Lisp
- 4 Connecting to Sage
- 5 Lisp Related Short Term Plans
- 6 Achieving Optimum Results

- There is a limit in size that the core version of Sage can grow to.
- Inclusion procedure at <http://wiki.sagemath.org/spkg/InclusionProcedure>.
- The “bus factor” of code is significant in the long term.
- History of the Sage project has taught us that supporting software is a non-trivial task.
- But: We are happy to accommodate everybody interested and provide infrastructure for optional SPKGs.

- 1 Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom**
- 3 Sage and Common Lisp
- 4 Connecting to Sage
- 5 Lisp Related Short Term Plans
- 6 Achieving Optimum Results

- Sage would prefer if there was only one happy Axiom family. In light of the forks would prefer to deal with the strongest and most active one.
- An optional spkg for Axiom (Bill Page) existed
- An optional spkg for FriCAS 1.0.2 (Burcin Erocal) and now 1.0.3 (Bill Page, Waldek Hebisch) is available.
- A more or less working pexpect interface exists. It was written for Axiom, i.e. pre-fork.

- 1 Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom
- 3 Sage and Common Lisp**
- 4 Connecting to Sage
- 5 Lisp Related Short Term Plans
- 6 Achieving Optimum Results

A short history of common Lisp in Sage:

- Candidates: sbcl, cmucl, gcl, clisp, ecl
- Criterion One: self hosted \Rightarrow sbcl and cmucl are out
- Criterion Two: Supported by Maxima \Rightarrow ecl is out
- Criterion Three: Actually works and builds on all of Sage's supported platforms \Rightarrow gcl and clisp is out
- We ship clisp since it is the the best of the above for our purpose.
- But: Maxima 5.16.0 will add ecl support, so Sage will dump clisp and switch to ecl.
- Ecl is great! Easy to compile, fast, great maintainer, embeddable!

- 1 Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom
- 3 Sage and Common Lisp
- 4 Connecting to Sage**
- 5 Lisp Related Short Term Plans
- 6 Achieving Optimum Results

- Python's shared library extension mechanism
- pexpect, i.e. pseudo ttys, ported to native Windows by the Sage team thanks to funding by MSR.
- speed vs. performance
- simple vs. complicated
- Both approaches have their place and none is inherently superior to the other approach.

- 1 Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom
- 3 Sage and Common Lisp
- 4 Connecting to Sage
- 5 Lisp Related Short Term Plans**
- 6 Achieving Optimum Results

- Help the Maxima people finish the ecl integration, i.e. sort out autoconf issues.
- Upgrade Maxima to the 5.16.0 release at the start of August.
- Switch over to ecl at the same time.
- ecl's dependency, i.e. boehm_gc is already packaged.
- The switch to ecl is absolutely essential for Sage's native Windows and Solaris port.

- 1 Optional vs. Core SPKGs
- 2 Sage and Pan-Axiom
- 3 Sage and Common Lisp
- 4 Connecting to Sage
- 5 Lisp Related Short Term Plans
- 6 Achieving Optimum Results**

- Connecting Aldor and Python: One possible route is
[Sage \leftrightarrow] Python \leftrightarrow Cython \leftrightarrow Aldor
- There are problems about types here, see the talk by Bill Page:
Python (class/metaclass) vs. Aldor (domain/category)
- Connecting Pan-Axiom and Python: Is it realistic to embed
Pan-Axiom into ecl and then use it as a shared library from
Sage?
- Aldor must be truly free IMHO to have a chance to prosper.
The competition from open tool sets is fierce and at least for
Sage it is about using a mainstream language. There are many,
many magnitudes more Python users than Aldor, which is a
tremendous factor in Sage's success since we can recruit from
a huge pool of developers.