GSoC 2015 Application

Travis Scholl
tscholl2@uw.edu
Department of Mathematics, University of Washington, Seattle WA 98195

March 27, 2015

1 Background

1.1 Math

I am a second year graduate student in mathematics at the University of Washington in Seattle. I am working towards a PhD. Academically, I am primarily interested in algebraic number theory, cryptography, and arithmetic of elliptic curves. I have taken several courses and independent studies in these fields as well as algebraic and arithmetic geometry.

1.2 Sage

I enjoy using Sage in my work where applicable. For example, computing primes of good reduction for a given variety or finding torsion points on an elliptic curve. While sometimes I believe in working through an explicit computation by hand, I often feel it's much better to verify the solution with Sage. The Python syntax and built in Sage packages make this very straightforward.

I am also constantly recommending Sage to my colleagues for their computations. I believe Sage is a great tool for a wide variety of mathematicians because many problems have computational aspects which can be efficiently solved using some package in Sage. Because of this, I would love to help contribute to Sage. Not only would it give me the opportunity to give back to the software that has helped me, but also add and refine functionality in order to help other researchers in similar situations.

1.3 Programming

Coding is a fun hobby of mine that I have enjoyed since college. I graduated from University of Oregon with a minor in computer science where I took standard courses on data structures and algorithms. I am regularly working on some type of pet project in my spare time. I am most interested in web based applications and database design. I have also worked with others on their projects including an indoor positioning system, a reddit bot, and educational websites for math courses.

My main platform is Windows but I regularly use linux based virtual machines and remote servers to work on various projects. My most comfortable programming languages currently are Python, Go, and CoffeeScript/Javascript. I have been using Sage since I came to University of Washington and would love to learn more about the contribution system and the organizational structure of the code.

---

2 Project Outline

2.1 Title

Adding and Optimizing Functionality in Algebraic Number Theory

2.2 Description

The point of this project will be to improve the Algebraic Number Theory (ANT) package in Sage. Specifically, Sage does not implement the Chinese Remainder Theorem (CRT) with ideals, only with specific elements. However, Magma's implementation allows for the use of ideals.[1] Also Sage's current Hermite Normal Form (HNF) algorithm only works over PIDs but PARI's algorithm works over orders in number fields.[2]

2.3 Details

1. Implement CRT for ideals as general as possible.
2. Compare implementations between PARI's and Magma's HNF algorithm in order to decide whether to wrap one or write a basic implementation into Sage directly.
3. Implement all necessary prerequisites for the HNF algorithm such as module theory over rings of integers for arbitrary number fields.
4. Write (or wrap) an implementation of the HNF algorithm and submit to Sage.

2.4 Schedule

- May 25th - June 5th

  I will be in school which will take take most of my time. However, I plan on using part of this period to review Sages internal organization and to get used to its established development process by meeting with my mentor and reading through the documentation.

- June 6th - June 10th

  Find sources which describe a CRT algorithm for use with general ideals. Also figure out if all available tools needed are already in Sage. Open a ticket for generalizing the current CRT.

- June 11th - June 25th

  Work on writing a working example of CRT with ideals. Try to generalize as much as possible. This will probably require working through a few sections of Cohen's book "Advanced topics in computational number theory".

- June 26th - July 3rd

  Review the basic CRT implementation with a mentor.

- July 5th - July 11th

  Start reading about HNF in Cohen's book and any other good resources to get an overview of the algorithm.

[1] See `http://magma.maths.usyd.edu.au/magma/handbook/text/174#1378`.

[2] See `http://pari.math.u-bordeaux.fr/pub/pari/manuals/2.7.0/users.pdf`, page 176.

- July 12th - July 19th

  Begin researching into HNF algorithm implementations in PARI and Magma. The goal will be to compare speed/quality between the two and decide whether to try and wrap existing function or writing a new one.

  A few days will be spent learning enough PARI/Magma to run and evaluate the algorithms. At least one day will be compiling all the data and reviewing it with a mentor to decide on the best course of action.

- July 20th - July 27th

  Map out any foundational requirements required for using HNF in Sage. For example, there may be missing parts of module theory over rings of integers of number fields that are necessary before writing the HNF algorithm.

- July 28th - August 16th

  Most likely this time will be devoted to wrapping the HNF from another library into Sage. Other options include possibly writing a basic, but likely slower, algorithm directly in Sage.

  The reason for the expanded time for wrapping is due to the age and unfamiliarity of PARI code.

- August 17th - August 24th

  Adjustment time for when previous steps take longer than expected.

2.5 Risk Management

The most likely problem will be time management. Implementing and optimizing CRT or HNF could take more time than expected. Also there is the worry that I could spend time implementing basic CRT or HNF algorithms for Sage only to find it will be too slow to use. In which case I could switch to wrapping the existing functions in PARI or Magma.
