Progress Report 1 for Report on the Risch Algorithm for Symbolic Integration and Implementation in the SymPy Computer Algebra System

Aaron Meurer

November 12, 2010

1 Research Conducted

1.1 Primary Research

For all intents and purposes, the primary research for my project has already been conducted. As I said in the proposal, I spent last summer learning and implementing the Transcendental Risch Algorithm for Symbolic Integration. Because I was implementing the algorithm, I had to fully understand it, both from a mathematical and a implementational perspective. Therefore, I have read quite closely almost the entirety of Bronstein's book [3]. In other words, this was more than just a cursory reading. In order to implement an algorithm, you have to understand exactly what is happening mathematically, so that you can fix problems when they arrise. Also, sometimes the algorithms given were ambiguous, requiring me to understand what should be happening enough to implement it, because actual implementations in programming languages cannot be ambiguous in any way, by their nature.

1.2 Secondary Research

Secondary research is actually the thing that I need to do more of. I still need to read through most of the citations that I go through in Appendix A. I think that some of these may end up changing for the final report, depending on exactly what direction I want to take with it. For example, the sources I have now mostly relate to things that I would want to do with the algorithm in the future that go beyond Bronstein's book. For now, that is the plan, because I don't see many other things that I could do with secondary sources. But, for example, one thing that I might choose to do instead would be to look at what other people in the field have done. That would require a whole different secondary source set. This would be more difficult, though, as it would require me reading source code for other programs, rather than reading articles from journals.

2 Modifications

Nothing has changed with respect to the outline, though some parts have been refined, as per the suggestions at the conference. The timeline may need to be modified a little bit, as I have not written as much of the paper itself at this point as I originally had planned. However, I do not see it as an issue, as I am already ahead with the actual research part.

3 Questions

I have one question. Should I cite the source code for the algorithm that I implemented in the references? The source code is freely available (it is open source, under the BSD license) at `http://github.com/asmeurer/sympy/tree/integration3`. If so, I'm not exactly sure how it will look. Citing source code is not a standard format in BibTeX.

Another issue is that SymPy is a collaborative project, written by many people, so the majority of the source code was actually written by someone else. You can see the changes that are actually by me at `http://github.com/asmeurer/sympy/commits/integration3` (any change by `asmeurer` is by me).

A Annotated Bibliography

- These sources all have a common thread: they all are about things that I could use to extend the algorithm or are about other algorithms related to symbolic integration. The plan is that someday, when I finish implementing the algorithms from Bronstein's book [3], that I will work on these.

  - **Bronstein** [2]: This is by the same author as and is referenced by Bronstein's book. The paper contains an algorithm used in integrating tangents that was not included in the textbook. Among the things that I am likely to implement in the future going beyond Bronstein's book, this is the most likely, as it is actually requirement for completing the algorithm (I believe Bronstein couldn't fit the material in his book, so he just referenced a paper that had written).
  - **Davenport** [4]: This source, like the next one, deals with the algebraic part of the Risch Algorithm. The Risch Algorithm, being the complicated algorithm that it is, is actually broken up into three parts. Bronstein's book only covers the first[1] and easiest part, the transcendental part. The other two parts are the algebraic part, which Davenport details here, and the mixed transcendental-algebraic part (the transcendental part deals with purely transcendental functions, the algebraic part deals with purely algebraic functions, and the transcendental-algebraic part deals with functions that are combinations of both types).
  - **Kauers** [5]: This paper describes a simple heuristic algorithm for the part of the algebraic part. The algebraic part of the algorithm is very difficult to implement—even more so than the transcendental part. Therefore, sometimes implementors will use a heuristic instead of the full algorithm. The downside to this is that a heuristic will be unable to prove when no antiderivative exists—it will only return with a failure. This doesn't constitute a proof of nonexistence because it will sometimes also fail even when an antiderivative does exist, which is another downside of using a heuristic. However, heuristics are easier to implement and often can act as fast preparsers to the full algorithm.

- These sources are about the history of the development of the Risch algorithm. They will be good sources for the background section of the report.

[1] First in the sense that the transcendental algorithm must be implemented before the algebraic algorithm can be.

– **Moses** [6]: This paper was written in 1971, whereas Risch described his algorithm in 1969 [7]. It compares the deterministic Risch algorithms with the heuristics that were employed in computer algebra systems prior to its discovery. The basic idea of the paper is that the heuristics are weak compared to the full algorithm, which integrates functions directly instead of trying pattern matching, algebraic manipulation, and other such heuristic methods.

– **Risch** [7]: This is the seminal paper in which Risch described his algorithm, filling in all the pieces and proving that the problem of integration in finite terms was decidable. Actually, the algorithm as originally described by Risch was difficult, if not impossible, to actually implement in any computer algebra system. It required the work of others, such as Bronstein and Davenport [4] to develop the algorithm in ways so that it could be implemented. Therefore, this paper if anything will provide an interesting comparison to the latest development of the algorithm given in Bronstein's book [3].

References

[1] V. S. Adamchik and O. I. Marichev. The algorithm for calculating integrals of hypergeometric type functions and its realization in reduce system. In *ISSAC '90: Proceedings of the international symposium on Symbolic and algebraic computation*, pages 212–224, New York, NY, USA, 1990. ACM.

[2] M. Bronstein. Simplification of real elementary functions. In *Proceedings of the ACM-SIGSAM 1989 international symposium on Symbolic and algebraic computation*, page 211. ACM, 1989.

[3] M. Bronstein. *Symbolic integration I: transcendental functions.* Springer Verlag, 2005.

[4] James Davenport. Integration in finite terms. *SIGSAM Bull.*, 18(2):20–21, 1984.

[5] M. Kauers. Integration of algebraic functions: a simple heuristic for finding the logarithmic part. In *Proceedings of the twenty-first international*

symposium on Symbolic and algebraic computation, pages 133–140. ACM, 2008.

[6] J. Moses. Symbolic integration: the stormy decade. *Communications of the ACM*, 14(8):548–560, 1971.

[7] R.H. Risch. The Problem of Integration in Finite Terms Trans. In *Amer. Math. Soc*, volume 139, pages 167–189, 1969.

[8] Kelly Roach. Meijer g function representations. In *ISSAC '97: Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pages 205–211, New York, NY, USA, 1997. ACM.

5/5 -- Very good.
And, you wouldn't need to cite the code, as it is easily accessible. But you might provide the URL (perhaps in a footnote).