Software Cult

Ivan Monroy Lopez

Metamathematics

The use of the axiomatic method goes back to the time of greek math. Basically it's the idea of developing theories from a small number of axioms via logical deduction. The classical example of this is Euclid's geometry. Euclid starts his 'Elements' treatise by stating five postulates and five common notions.[1] From this seed, he develops his theory of geometry by logically accumulating result upon result.

[1] In the first two sections, italics are used when the phrase or word has a special meaning for the math in question.

In the 1920s, David Hilbert proposed a metamathematical plan of action that now goes by the name of 'Hilbert's program'. It is called metamathematical because it is not really concerned with say, the properties of triangles, but rather with the properties of the properties of triangles.

Were it possible to carry out, Hilbert's program would be like the natural conclusion to the use of the axiomatic method. At some level of naïve intuition, it even feels plausible and doable. One thinks: 'if the axiomatic method has so far proved very effective, why should it not be able to fulfill Hilbert's wishes?' In a nutshell, he proposed to (a) axiomatize all of math, and (b) derive the *soundness* of the axiomatization from the axiomatization itself.

In books like 'Principia Mathematica' people like Russell and Whitehead put years of work trying to advance the Hilbert program. Now we know – thanks to Gödel's *incompleteness* theorems – about the recursive shortcircuitry that kept all those efforts from making much progress.

One could roughly say that Gödel showed that no axiom system that contains arithmetic can (a) account for its (the system's) *soundness*, and (b) account for all arithmetic truth. This means that – in a way – all axiom

systems that contain arithmetic are incomplete – hence the name. What's interesting about these theorems is that they reduce an involved math/philosophy issue to a simple yes/no question, and then they answer this question with a no.

[2] http://plato.stanford.edu/entries/logic-ai/#jmc

2. Logic, Computer Science, Artificial Intelligence

Gödel's negative results brought new energy to the field of mathematical logic. It is out of this period of intense activity that Computer Science springs. In this respect, the classical example is that of the Turing machine – a theoretical construct for today's computers that was employed by Alan Turing when tackling the metamathematical *halting problem.* Another lesser known and more recent result in this direction is the so-called 'Curry-Howard' isomorphism.

This result expresses rigorously the intuition that a lot of computing is already implicit in abstract math. It asserts the equality of the apparently disparate concepts of computer programs and mathematical proofs. One informal way of stating it is: programs are proofs.

With regards to this theorem Berline cites an avenue of experimentation that looks interesting. If programs and proofs are equivalent, then we can ask ourselves: 'what programs correspond to the proofs that we already know?' In that there's a long history of mathematical proofs that precedes the invention of physical computers, this is a rather arbitrary way of coding – math becomes like a clunky mindless interface that churns out potentially useless software.

At least historically, there are also connections between mathematical logic and Artificial Intelligence. Alongside chess playing, theorem deduction is one of the classic types of reasoning that AI research tried to emulate. Apparently there's even a whole subfield called logical AI. According to some website, 'the most influential figure'[2] in logical AI is John McCarthy, the creator of LISP. He also makes an appearance in the retelling of early hacker history that occupies the first chapter of Steven Levy's *Hackers* – he's the professor that taught the first course on programming that the Signals and Power members of the Tech Model Railroad Club enrolled in:

> *That spring of 1959, a new course was offered at MIT. It was the first course in programming a computer that freshmen could take. The teacher was [...] John McCarthy.*[3]

Later in that story, McCarthy – who was also trained as a mathematician – gets the hackers to work in the computer chess problem.

[3] S. Levy, Hackers, New York, Delta, 1985. Chapter 1

[4] http://www.oreilly.com/catalog/opensources/book/stallman.html

So far, I've tried to give an overview of some of the ideas at the intersection of math and computing that I find interesting. I've done this in an effort to sketch one of the traditions that Richard Stallman belongs to. I think that he belongs to this tradition because he was a good mathematician who got into computers, and who ended up working at the MIT AI Lab.

In general, I think that hacking not only finds its roots in things like the Tech Model Railroad Club, but also in the sort of stuff that I've been talking about. There's – or there could be – both engineering and pure science in hacking. If anything, this heritage may help in understanding the more utopian strains of software development.

Take for example, Stallman's classic: 'With my community gone, not continuing as before was impossible.'[4] If we take into account Stallman's roots, his words could be interpreted as nostalgia for a time when the hacker community still fell under the official protection of the academic institutions of science; a time before it was discovered that software could be sold as a commodity (unlike other more abstract mathematical products.)

3. TV-Edit

I think that the story that I've been sketching is also about the materialization of abstract concepts. Computers are in the flesh realizations of metamathematical concepts. I think that this could be one way of enriching the discussions about computing. We could, for example, search for traces of platonic contemplation in 'staring into monitors' phenomena – all the way from monastic laptop music performance to computer addiction anxiety.

If anything, mathematical abstraction can be difficult. It's hard to get your head around objects whose only distinguishing features at times are like fleeting shadows. This is probably the reason why abstraction is also one of the things that some people strive for in their work style. In this respect one of many possible references is 'abstract nonsense' an idea that finds its origins in Category Theory. Though it may sound derogatory, it is actually used 'as an indication of mathematical sophistication or coolness.'[5]

A book that also talks about the intersections of computing and math, but from a slightly different angle is Douglas Hofstadter's *Gödel, Escher Bach*. One of its weaknesses is that it is the type of book with an authoritative tone. When reading it, one imagines big men tackling big questions: Hofstadter is at the cutting edge of science (in 1979), working in the novel field of AI. Not only is he a very intelligent human being, he's also investigating human intelligence – in an intelligent manner, one would gather.

There are also good things about *GEB*. It is a book that exudes energy. Its gross generalizations[6] make me think of Hofstadter as a high-speed idea-spewing machine that has little time to double-check every single fact. His book may even belong to a self-publishing tradition – he was afforded the luxury (at the time) of typesetting his own book. In terms of book design, it even has style:

> *Equally important to me, however, is Pentti's rare quality: his sense of style. If my book looks good, to Pentti Kanerva is due most of the credit.* [7]

In this quote from the 'Word of Thanks' Hofstadter is referring to TVEdit, one of the alternatives to Emacs that were available at the time. It was built by Kanerva, and it's the text-editor that Hofstadter used to write *GEB*. It's remarkable that he decided to thank someone who was involved in the making of his book at such a fundamental level. Nowadays, this extended credit sequence could painlessly be skipped.

I think that this quote gives us a glimpse into the world that hackers like Stallman inhabited. It would seem that the boundaries between hackers and users were blurry back then. At least the two social groups were on better terms than they are today. It's hard to generalize, but these days the profusion of abuse on the user's part is almost like genuine class exploitation. The sys-admin hell that this gives rise to is obliquely hinted at in things like the dice.com 'success stories.'[8]

4. Free as in O'Reilly

Let's talk about the O'Reilly book *Free as in Freedom*. In Chapter 11, it chronicles a meeting organized by Tim O'Reilly during which important work towards cementing the open source definition was done. This was a meeting that snubbed[9] Stallman, as he was not invited. After reading

[5] http://en.wikipedia.org/wiki/Abstract_nonsense

[6] 'Every mathematician has the sense that there's a kind of metric between ideas in mathematics - that all of mathematics is a network of results between which there are enormously many connections.' D. Hofstadter, Gödel, Escher, Bach, New York, Vintage Books, 1980, p. 614

[7] D. Hofstadter, Gödel, Escher, Bach, New York, Vintage Books, 1980

[8] http://www.dice.com/stories/success_stories.html

this, I could not help but wonder about O'Reilly's intentions in publishing a book that deals with something that he was involved in undermining. Although the book ends with the requisite appendices containing an 'introduction' into hacker culture, and the GNU Free Documentation License (GFDL), its real end is a rosy epilogue. In this epilogue we're afforded a look into the red tape that went into writing the book:

> *The drama in front of the curtain often pales in comparison to the drama backstage [...] The story behind this story starts in an Oakland apartment [...] Ultimately, however, it is the tale of two cities [...] The story starts in April, 2000 [...]* [10]

Williams does go to great lengths to explain how Stallman was the intellectual best man of his wedding. But – aside from a good account of the implications of the GDFL for his own book – all we get in the pages where he talks about O'Reilly is praise.[11] No indications as to why O'Reilly chose to publish it are given.

Parts of the book read like welcome elaborations on parts of the Stallman essay from 'Open Sources.' Take for example, the chapter 'For Want of a Printer' – it's a retelling of the story about how someone refused to give Stallman the source code for the MIT AI Lab's printer. Also interesting, is the way in which Stallman's launch of GNU is portrayed, especially in relation to anger and angst as creative drives.

I'm not entirely at odds with the portrayal of hackers as angry or disaffected teenagers – some hackers may actually fulfill the cliché. The allusions to dice.com in the previous section are indications of how teenage alienation can turn into worker exploitation beyond the realm of the abstract. Stallman may have even fallen into this hacker category at some point. Maybe there were even elements of solitude[12] and anger[13] in Stallman's decision to start GNU development. However, to put so much weight on such sentimentality is to turn a blind eye to the politics of Stallman. *Free as in Freedom* didn't have to be a GNU propaganda pamphlet, but it didn't have to be an depoliticized book, either. Its psychological absolutions of radical postures are condescending, at best.

An actual discussion of GNU politics could have been more interesting. I for one sometimes feel that operating system development is too abstract a field for activism. In my view, there are more urgent issues to

[9] I'm using Williams' exact wording. He leaves the question on whether Stallman was or was not snubbed unresolved.

[10] S. Williams, Free as in Freedom, Sebastopol,O'Reilly, 2002. p. 185-6

[11] 'Of all the publishing houses in the world, O'Reilly […] seemed the most sensitive […] As a reporter, I had relied heavily on the O'Reilly book 'Open Sources' […] During the O'Reilly Open Source Conference in 1999, I […] had been wowed.' S. Williams, Free as in Freedom, Sebastopol, O'Reilly, 2002 p. 194-6

[12] 'Richard built up an entire political movement to address an issue of profound personal concern […] [c]rushing loneliness.' S. Williams, Free as in Freedom, Sebastopol, O'Reilly, 2002 p. 198

[13] 'righteous anger […] has propelled Stallman's career as surely as any political ideology or political belief.' S. Williams, Free as in Freedom, Sebastopol, O'Reilly, 2002. p. 12

attend to. Still, I'm sympathetic to GNU and would like to be convinced otherwise.

The next section is devoted to painting a portrait of GNU as a cult. In the face of the Magic Pixie Dust that something like O'Reilly Media seems to embody, I've chosen to play with the idea of a cult as a way of showing support for a more radical current of software development.[14] In this I'm inspired as much by the Church of the SubGenius, as by the Church of Emacs. I should also mention that the focus on GNU is largely rhetorical. Beyond GNU, there are other mock cults in FLOSS.

5. Software Cult:

5.1. Isolation

Cult membership sometimes entails changes that could lead members to isolate from their social circle. Extreme cults demand that their members entirely break any contact with even their closest relatives and friends if they have not been already converted. Likewise, the use of GNU could translate into an implosion of someone's social life. If you use Skype on a power pc, switching to GNU would mean closing an avenue of communication with your acquaintances. There are other communication technologies, but these may be populated by people who are already GNU users. Any emotional attachment to utilities like Powerpoint will have to be severed. For some people, this could be painful.

5.2. Conversion

In general, computing-related activities involve some degree of coercion. In the end, computing is a surplus. To paraphrase Hakim Bey, we got along fine without the ability to sustain six phone conversations at once.[15] In this context, usability could be seen as a business strategy aimed at maximizing consumption. It's strange that at its basest level, this approach leads to things like idiotic instant messaging emoticons.

At the other extreme lies terminal-based work. Hackers have had the privilege of going through a 'learning period' in one way or another. However, for many people who come later in life to free software, especially if they have had previous experience with GUIs, the experience can very easily lead to frustration. The paradigm-shift can be so daunting, that it can be likened to thought reform or brainwashing.

[14] More recently, Tim O'Reilly coined another term, namely Web 2.0. Magic Pixie Dust . This comes from an early survey in which Register readers were asked to 'redefine the paradigm': http://www.theregister.co.uk/2005/11/11/web_two_point_naught_answers/

[15] 'OK, I can now carry on six phone conversations at once. But what difference has this made in my ordinary life? […] Frankly, I already had plenty of data to enrich my perceptions', H. Bey, Temporary Autonomous Zone, New York, Autonomedia, 1985, 1991 , http://hermetic.com/bey/taz_cont.html

5.3. CSS Asceticism

The design of the gnu.org webpage changed a few months ago.[16] It now boasts an enormous grey/blue header. It has a navigating bar that makes heavy use of the a.hover pseudo-element. Its ubiquitous unordered lists have also been styled – bullets are now bluish squares.

It's strange that even with that re-hash, its designers did not decide to make much use of the link variety of pseudo-elements. Most unvisited links are still blue, and they turn magenta once they've been clicked. The a.hover pseudo-element is used, but only to turn links red – the default color they take on when active. Overall, the GNU webpage gives the impression of being on the verge of losing its styling.

Before its redesign, the homepage of GNU was for many years a practically unformatted page. Sometime in early 2004 it acquired those sans-serif fonts, and its present right column / horizontal bar layout. Its earliest recorded version at the internet archive[17] dates from 1998, when it was apparently updated by rms himself. The design is as sober as it gets – a few h2 tags, one h3 tag, and one image. The contact information is inside a pre tag.

5.4. Iconography

There's also asceticism in some of the GNU imagery. I'm thinking particularly of the many simple black and white anthropomorphic gnu illustrations.[18]

There's a connection between this imagery and a certain para-mathematical visual tradition – the type of images that are sometimes used as book adornments. Like the GNU drawings, these images tend to also be naïve, and a bit outer-worldly, even when they're not lacking in conceptual depth. Some of them veer closely to the mnemonic research map genre. The best of them have a sense of humor, and play on a sort of 'less is more' strategy – like Guy Steele Jr.'s 'Crunchly Saga'.

5.5. Guru

There are few interest groups which have such a clear and visible leader as Stallman. This is probably due to a strong personality that sometimes veers on the aggressive. In his St. IGNUcius persona, Stallman himself plays with the idea of being a religious leader. He's probably the

[16] This section makes reference to the design of gnu.org at the end of 2006. There are some approximations at the internet archive: http://web.archive.org/web/20061219051658/http://www.gnu.org/

[17] http://web.archive.org/web/19990125085924/http://gnu.org/

[18] Examples of these may be found in http://www.gnu.org/graphics/graphics.html This tradition is fleeting, and almost non-existent.

principal free software evangelist, a term that already has cultish overtones.

All free software is open source software, and vice versa. It could be argued that the open source definition came into being just to get Stallman out of the picture. Conversely, he has a reputation for not budging an inch, thus alienating a lot of potential allies. This has the effect of increasing the isolation of his followers that was alluded to previously. In this, GNU is like a political cadre.

Even though Stallman is an atheist, he speaks about the beginnings of GNU in almost religious terms. In his own narrative, ethical choices were involved; a community was dispersed; and Stallman played a role akin to that of a messiah:

> *I had just the right skills to be able to do it [...] and nobody was there but me [...] I felt: 'I'm elected, I have to work on this. If not me, who?'* [19]

[19] 'Free Software: Freedom and Cooperation' speech, http://www.gnu.org/philosophy/audio/audio.html#NYU2001

References

C. Berline, 'A Presentation of the Curry-Howard Correspondence', http://www.pps.jussieu.fr/~berline/Cur-How.ps

H. Bey Temporary Autonomous Zone, New York, Autonomedia, 1985, 1991, http://hermetic.com/bey/taz cont.html

C. Dibona, S. Ockam, M. Stone, eds, Open Sources, Sebastopol, O'Reilly, 1999.

D. Hofstadter, Gödel, Escher, Bach, New York, Vintage Books, 1980.

S. Levy, Hackers New York, Delta, 1985.

S. Williams, Free as in Freedom Sebastopol, O'Reilly, 2002.