Philosophizing from the bench

Eric Blair

4 April 2007

I have written elsewhere about how there is no economic justification for patents on software or business methods, and how the legal basis of such patents is based on a very specific (and in my opinion, false) reading of prior case law.

But there is one final question regarding such patents: are they *ethical*? To answer this question, we must answer another entirely unanswerable question: do people *invent* mathematical results or *discover* them? Are the symbols mathematicians write down a reflection of some innate structure of the universe, or just human symbols manipulated using human rules?

One could find people on all levels of the spectrum between math as pure invention and math as pure discovery, but this has not always been true: before about two hundred years ago, mathematical results were firmly a part of nature that humans stumbled upon. In such a context, the ideal of granting a patent—an ownership right—in a mathematical algorithm would have been taken as simply absurd, an unethical and unenforceable handing over of a piece of nature to one person. The monopolies now granted for mathematical algorithms are thus the product of a few centuries' worth of development in mathematics and our attitude toward the subject.

Unfortunately, software patents do not represent the cutting edge in modern sensibilites regarding the nature of mathematical algorithms. Instead, they make sense only via a school of thought that was prevalent from the late 1800s until it was discredited in 1931. Thus, the courts have tried to keep the law up-to-date by revising the scope of patent law from where it stood in 1790, but they remain behind the times nonetheless.

The realists The *realist* view originated with Pythagoras (about 582-507 BCE). Pythagoras observed various regularities, like how the sound of a plucked chord made the most harmonious sound when played in concert with a chord exactly half its length (what we now call an octave apart), and then with a chord a third its length (a fifth apart), et cetera.[1] He concluded from these pleasing regularities that all of the world is a reflection of a set of harmonious mathematical relationships—a music of the spheres.

Plato (born about 75 years after Pythagoras's death) picked up on the Pythagorean's geometrical obsession. If you've ever taken a philosophy class, you are familiar with Plato's view that the forms we see are vague, secondary reflections of a perfect ideal—nature is a reflection of mathematics. Plato said that people *remembered* mathematical

---

[1] See Donald Duck in Mathemagic Land.

results, because they are imprinted in our minds and we need only get the right signal to remind ourselves of the mathematical truth inside ourselves.

Around this time, it was a popular trick to try to try to write down as many theorems as possible from the basic axioms of geometry. The most famous such attempt is Euclid's Elements. This is an oft-told story, but here are the first five basic assumptions Euclid needed to derive all of geometry:

1. A straight line segment can be drawn by joining any two points.
2. A straight line segment can be extended indefinitely in a straight line.
3. Given a straight line segment, a circle can be drawn using the segment as radius and one endpoint as center.
4. All right angles are congruent (i.e., equal).
5. If two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough.

If you're like most people, you were nodding your head up until you got to that last one, which is something of an eyesore in its lack of simplicity; we'll get back to it shortly.

2000 years pass. The history books typically characterize these as periods of religious fervor, but that doesn't mean that they're periods of scientific inactivity. However, the realist viewpoint took a small twist: it's not that nature haphazardly reflects mathematical ideals, but that the Divine Creator used math to design everything. But people like Copernicus and Newton still saw themselves as just marvelling at how neatly the Divine Watchmaker designed the mathematical world around us, and still placed themselves in the role of observer rather than inventor.

The formalists appear Back to that fifth assumption. Some tried to derive it from Euclid's other axioms. They would begin by assuming that the fifth assumption is false, and then search for a contradiction to the other axioms. But a funny thing happened: under some means of constructing a system where the fifth axiom is false, no contradictions turn up. One could construct a whole world that in many ways looks absolutely nothing like Euclid's. Gauss, the inventor/discoverer of Gaussian elimination, the Gaussian distribution, Gaussian quadrature, and et cetera, was one of many in the mid-1700s to question that fifth postulate. What if you could have lines point toward each other but still never meet? We thus get non-Euclidian geometry, which caused something of an explosion.

This was the first chance for the *formalist* viewpoint to take hold. Many of these non-Euclidian geometries didn't describe anything we have seen here on Earth. Some got the last laugh a few centuries later when Einstein showed that non-Euclidian geometry sometimes did a better job of describing reality than Euclidian, but at the time there was the gnawing question that maybe these axioms and their derived results were just a set of amusing inventions—human-made symbols that reflect nature only by sheer luck, if that.

The project of mathematics became the problem of designing systems of symbols and their manipulation that are interesting in and of themselves. Of course, such a system should not self-contradict, and reflects at least some of our intuitive beliefs, like how if $A = B$ and $A = C$, then $B = C$.

Here in the present day, mathematical geometry courses build the subject in a series of steps. They start with defining sets, then establishing the characteristics of open and closed sets, then describing networks of those (aka topologies), and then adding in neighborhoods (aka manifolds), and only then can the concept of distances (metric spaces) come in. So what Euclid took to just be space, we take to rely on definitions of sets, algebras, topologies, manifolds, and metrics.

The larger project is to have a unifying set of symbols, beginning with sets, that would allow one to trace the most advanced mathematical ideas all the way back to basic manipulations of sets. Like the Euclidian craze of the Greek era, the late 1800s–early 1900s brought about a flurry of people writing derivations of as much as possible from basic axioms of sets. The stand-out attempt was Whitehead and Russel's *Principia Mathematica*, which went pretty darn far in starting with very simple symbols and building up basically everything.

It will be relevant below that Bertrand Russel, the paragon of the set theoretic formalization of mathematics, was not very happy with his symbolic designs:

> I wanted certainty in the kind of way in which people want religious faith. I thought that certainty is more likely to be found in mathematics than elsewhere. [. . .] But as the work proceeded, I was continually reminded of the fable about the elephant and the tortoise. Having constructed an elephant upon which the mathematical world could rest, I found the elephant totterling, and proceeded to construct a tortoise to keep the elephant from falling. But the tortoise was no more secure than the elephant, and after some twenty years of very arduous toil, I came to the conclusion that there was nothing more that I could do in the way of making mathematical knowledge indubitable. [Russel, 1956, pp 54–55]

Russel thus raises a natural question: how far can all this go? Kurt Gödel famously showed that it's not as far as one would hope, using the formalization of the following simple declaration. *This sentence is false*. If that sentence actually is false, then the sentence is proclaiming a true statement—which means that the sentence is actually false—which means that it's true. . . .

Gödel's version was the statement "This statement is not provable using the logical system L." Name this statement S. If S were provable using the system L, then the statement would be false, meaning that L has proven a contradiction. If S is not provable using L, then L is incomplete, in the sense that S is a provable statement (it is not provable using L, as promised), but L can not prove it.

This was a horrible blow to the formalists. However powerful their system, there would still be simple logical chains like the last paragraph that prove things that the logical system can not handle. The formalist movement basically lost credibility. The proof that there is something to mathematics that our symbolic systems can not handle clearly advocates for the realist side of the spectrum.

The derivation of computing At this point, the symbol-manipulators did not entirely give up, but instead rephrased the question: having accepted that for any logical system some expressions are not evaluable, what mathematical expressions *are* evaluable? Two people simultaneously provided suggestions of determining what is evaluable, in 1937–8. The first, Alonzo Church, invented a means of writing expressions, claiming that his writing scheme covered all possible evaluable operations. Alan Turing took it in a slightly more imaginative turn: he described a machine with a tape (memory) and a head that moves along the tape and modifies the data written thereon; if Turing's machine can evaluate the expression in finite time, then it is evaluable.

That is, Turing described a computer, and said that if something is evaluable via computer, then it is evaluable via the systems of set theory as well. In fact, every modern computer out there is equivalent to Turing's machine and Church's lambda calculus (which are themselves equivalent). Barring highly specialized languages, virtually every modern programming language is *Turing equivalent*, meaning that it is equivalent to a Turing machine, the lambda calculus, and all of the other Turing equivalent languages. That means that programs written in a modern programming language are equivalent to mathematical expressions using traditional mathematical notation.

Thus, modern computing has its roots in the set theoretic attempts to write a language that describes everything, which in turn has its roots in the formalist perspective that mathematical symbols need not reflect any inherent logic of the universe.

In computing, the bias toward formalism is still heavier, because programs *look* like human designs. Further, they are often describing systems built by humans. *Geometry* is Greek for "measuring the Earth," because it was first used (by the Egyptians) for surveying land, but as the mathematical and computational edifice grows taller, it becomes increasingly difficult to see the ground below.

The final step in formalist philosophy The birth of formalism laid the foundations for the software patent.

There is an understanding that laws of nature may not be patented, which persists to this day. The law of gravity, or a newly discovered element, are not human inventions, but discoveries regarding nature. Within the law of nature exception lay a sub-exception: mathematical algorithms may not be patented. In the terminology above, setting mathematics as a subset of nature is clearly and firmly a realist view. This is appropriate, because Thomas Jefferson wrote the first patent law in 1790, while Gauss had written his development of non-Euclidian geometry around 1820–1830. The realist school was thus the prevalent (and only) understanding of mathematics when the patent law was written. Legal scholars often ask what the "congressional intent" was behind a bill, and it is effectively impossible for the congressional intent to have been that mathematical results are not laws of nature.

Now let us skip forward to 1980, at the founding of the Court of Appeals for the Federal Circuit (CAFC), to consolidate patent hearings (and some other issues) into one specialist court. Several of the judges on the CAFC bench are assigned to hear only patent cases—cases about human inventions. Many of them are former prominent patent attorneys. Therefore, it is no surprise at all that with regards to mathematics, they are formalists.

Let us open with a law review article from 1986, five years after the Supreme Court ruled for the third time that mathematical algorithms may not be patented: "A mathematical or other algorithm is neither a phenomenon of nature nor an abstract concept. [A mathematical] algorithm is very much a construction of the human mind. One cannot perceive an algorithm in nature. The algorithm does not describe natural phenomena (or natural relationships)."[Chisum, 1986] This passage is clearly a product of the towers of elephants and tortoises above. Russel's *Principia Mathematica* was published in 1913, and this law review passage arrived 73 years later. Given the speed at which attitudes toward mathematics move, this perspective is downright trendy.

In the courts, the origins of the software patent are typically traced to the ruling written by Judge Giles Rich in *In re Alappat* (33 F.3d 1526, 31 USPQ2d 1545, 1994), which split the mathematical algorithm exception off from the law of nature exception—and then denied the existence of the mathematical algorithm exception:

> [T]he Supreme Court explained that there are three categories of subject matter for which one may not obtain patent protection, namely "laws of nature, natural phenomena, and abstract ideas." . . . the Supreme Court also has held that certain mathematical subject matter is not, standing alone, entitled to patent protection. . . . A close analysis . . . reveals that the Supreme Court never intended to create an overly broad, fourth category of subject matter excluded from Section 101.

Clearly, this discussion makes no sense if a mathematical algorithm falls into the categories of "law of nature, natural phenomena, and abstract ideas."

Having split off mathematical algorithms as a separate category from things existing in nature, the Federal Circuit killed it off by 1999. *AT & T v Excel* (172 F.3d 1352, 50 USPQ2d 1447, 1999), cited earlier CAFC rulings to determine that: "the judicially-defined proscription against patenting of a 'mathematical algorithm,' to the extent such a proscription still exists, is narrowly limited to mathematical algorithms in the abstract." Such a narrow limitation is no limitation at all, because it is trivial to state "I claim a machine on which is loaded an algorithm to. . . " before any purely abstract algorithm. Indeed, patents granted based on such wording abound.

But the world is not formalist As you can see, the CAFC has positioned itself at the formalist extreme of the formalist-realist spectrum. However, since Gödel, few practitioners of math and computer science placed themselves at such an extreme.

Dedekind was a mathematician instrumental in the development of set theory, and was thus essential to the formalist camp. In the opening to his notes on differential calculus [Dedekind, 1901, pp 1–2], he cast himself as a formalist, complaining that resorting to intuition ". . . can make no claim to being scientific, no one will deny. For myself this feeling of dissatisfaction was so overpowering that I made the fixed resolve to keep meditating on the question till I should find a . . . perfectly rigorous foundation for the principles of infinitesimal analysis [differential calculus]." But there his formalism toward differential calculus gives way to his realist belief, that he was working to ". . . discover its true origin in the elements of arithmetic and thus at the same time to secure a real definition of the essence of continuity."

I have not yet mentioned the *intuitionist* movement, which Kline [1980] traces back to the early 1900s. The position of the intuitionist is closer to the realist: we all know what zero and one are, an we all have an idea of what addition and multiplication mean, so we should build from there. Causality is something about which we all have an intuitive grasp, but which is simply impossible to pin down using statistical tools. Judea Pearl, the author of the standard reference on causality [Pearl, 2000], is entirely unfazed by the fact that his chosen subject is completely ungrounded: "For me, the adequacy of a definition lies not in abstract argumentation but in whether the definition leads to useful ways of solving concrete problems. The definitions of causal concepts that I have used in my book have led to useful ways of doing things. . . ."[2]

So while patent law has followed the single thread of formalism to the exclusion of all other threads, the typical person having ordinary skill in the art of computing and mathematics believes a mix of the realist, the intuitionist, the formalist, and perhaps even the theological. Therein lies the conflict: Judge Rich was philosophizing from the bench, and mandated that patent law shall take the formalist viewpoint that mathematics is the human manipulation of human symbols—but mathematicians themselves have prevalently had the view that strict formalism is an inaccurate description of mathematics and computing since the 1930s. Practitioners thus see the patentability of software and mathematical results as based on a false—and even condescending—view of their chosen field.

References

Donald S Chisum. The future of software protection: The patentability of algorithms. *U Pitt Law Rev*, 47:959–961, 1986.

Richard Dedekind. *Essays on the Theory of Numbers*. Open Court Publishing, 1901.

Morris Kline. *Mathematics: the Loss of Certainty*. Oxford University Press, 1980.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Bertrand Russel. *Portraits from Memory, and Other Essays*. Simon and Schuster, 1956.

---

[2] `http://www.mii.ucla.edu/causality/?p=33`