

Fugue No. 3

C-Sharp Major

Well-Tempered Clavier Book II

Johann Sebastian Bach

© 2003 Timothy A. Smith (the author)¹

To read this essay in its hypermedia format, go to the Shockwave movie at <http://bach.nau.edu/clavier/nature/fugues/Fugue27.html>.



Subject: Fugue No. 3, *Well-Tempered Clavier*, Book II

The human mind has first to construct forms, independently, before we can find them in things.

Albert Einstein

In 1980 Douglas Hofstadter forever associated the names of Gödel, Escher, and Bach. Here we relate the fugue to:

- Gödel's theorem of incompleteness
- Turing's machines
- a symbolic system
- self-referential ways of knowing
- versus tacit and truthful knowing

We are about to brew a tempest in a teapot. If you are the type who likes to find shelter on the approach of a storm, you might rather just listen to the fugue and study the timeline. If you don't care about Gödel but would like to know how this fugue transforms its subject, skip to a symbolic system. But if you like to stretch your mind, continue reading from here.

¹ You may print, copy, link to, or cite this document, for non-profit educational purposes, so long as credit is given to the author as per fair use. You may not reproduce this document electronically, enfold it into a web site, or incorporate it into a saleable product without written permission from the author.

Gödel's Theorem of Incompleteness

Let us suppose that you must decide if the following statement is true or false. Here's the statement:

This statement is false.

If you decide that the statement is true, then it is false. But if you decide that it is false, then it is true. If you think hard enough on it your mind will begin to melt.

These types of problems are known as *undecidable propositions*, and they have been around for a long time. Epimenides of Crete posited one of the more famous: "All Cretans are liars," he said. This one, known as the *Cretan paradox*, will also turn your mind to mush if you let it. The Cretan paradox is (in a teapot) the type of problem that prompted Gödel's theorem of incompleteness.

In 1931 Kurt Gödel (1906-1978) published the most important mathematical work of the twentieth century: "On formally undecidable propositions of *Principia Mathematica* and related systems." Gödel was responding to the landmark *Principia Mathematica* published by Bertrand Russell and Alfred Whitehead two decades earlier. Russell and Whitehead had attempted to reduce mathematics to a system of logic. *Principia* is still regarded as the greatest book on logic since Aristotle's *Organon*.

But Gödel proved that logic ultimately leads to propositions that cannot, by the same logic, be proved or disproved. Gödel's work has created doubt about the extent to which we "know" in the Euclidian sense. In Hofstadter's words: "Gödel showed that provability is a weaker notion than truth, no matter what axiomatic system is involved."

Now that we have elevated this subject to cosmic heights, it will be helpful to unpack Gödel's theorem, which may be applied to this fugue. But in order to apply it, we'll need to understand it. Don't worry; the basic ideas are quite simple.

Gödel's first theorem of incompleteness states that any formal system that is complex enough to do basic arithmetic will have at least one proposition that cannot be proved or disproved by the system. This means that we can know that something is true, but the system cannot provide the logic to demonstrate it. The larger implication is that the domain of provable things is always incomplete — smaller than what is true. Inversely, the domain of truth is always larger than what can be proven true.

As part of his proof, Gödel demonstrated something else. His second theorem replies: it is impossible to prove, by using the rules of any complex system, that the system is self-consistent. This means that the more one tries to erect a consistent system, the more complex it becomes, therefore more prone to reveal its inconsistencies.

Turing's Machines

Alan Turing (1912-1954) showed that what Gödel applied to static systems (things that never change) also applies to dynamic systems (things that change by learning from their experiences). Specifically, he showed that Gödel's

theorem applies to the computer.

Now you may object that the computer as we know it didn't exist in Turing's lifetime. In Turing's day a "computer" was a person who performed computations. So at their first mention, Turing's computing machines were imaginary. They were capable of doing more than ours; they could read, write, recognize their mistakes, erase and correct them.²

In his modeling of mathematical problems Turing employed these fantasy computers that he stipulated could infallibly do all that a person could do plus what our computers do: lightning-speed arithmetic, Boolean values and logical operations like, if this is true then that is false. Today we call these imaginary things *Turing machines*.³

Feeding a question to a Turing machine is called a *Turing test*.⁴ A Turing test reads like a conversation: G asks TM (Turing Machine) if X is a true statement and TM replies that X is false. This hardly seems like mathematics! Because it doesn't use numbers (yet), it is called *modeling the problem*. Once they've gotten the model to work, mathematicians will express it arithmetically and "prove" something.

In a moment we'll ask our own Turing machine some questions about this fuge. But first it will be helpful to know more about the problem that Turing solved.

Turing's paper, "On computable numbers, with an application to the *Entscheidungsproblem*," was written in response to Hilbert's 1928 assertion that there is a method whereby any mathematical "decision problem" (*Entscheidungsproblem*) can be solved. Turing showed that Hilbert's thesis was false; there are relatively few problems that can be solved, but many more that cannot.

Turing's proof required the application of what we have described as Turing tests. Building on Gödel's work of nine years earlier, Turing restricted his machines to "computable numbers," meaning real numbers whose expression as

² Bach's contemporary, Gottfried Wilhelm Leibniz, was the first to use a binary number system, which is the mathematical foundation for today's computer. While programmable devices had existed since the late 18th century, the modern computer was first proposed by Alan Turing in his 1936 paper, "On Computable Numbers." Turing was a key player in the effort to decipher encrypted communications by the Germans in WWII. In 1939 that project resulted in the creation of the "Bombe," the world's first electronic *mechanical* computer, and four years later, the "Colossus," the first *digital* programmable device. But for want of electronic memory and algorithmic execution, neither machine was "Turing complete," as envisioned in his paper. This required invention of the bipolar transistor (1947) and integrated circuit (1952). The first *Turing complete* computer was made by Konrad Zuse in 1953, the year before Turing died.

³ All modern computers can be properly described as "Turing machines."

⁴ A Turing test evaluates the computer's response to a question to determine if it can be distinguished from that of a human. Have you ever been asked to supply a CAPTCHA? Guess what! That is the acronym of **C**ompletely **A**utomated **P**ublic **T**uring **T**est to tell **C**omputers and **H**umans **A**part. The backstory of the Turing test is the question, "Do computers think?" Since his colleagues couldn't agree on what it means, "to think," Turing asked a different question, "Can a human tell the difference between a computer's response, and that of another person?" It matters not if the machine can *think*, but only that it can *perform* at a human level. And why do we ask such questions; it is the dream of artificial intelligence!

a decimal is calculable by finite means. To his wonderment he discovered that the proportion of computable numbers is infinitesimally small as compared with the total class of real numbers. In plain English, no matter how large you make your computer (how much it learns), there will always be more problems that it cannot solve, than those it can.

Now if all of this has confused you, just think of a Turing machine as an imaginary lie detector, or Truth Machine (TM). It will tell you, based upon information you provide, if what you say is true or false. If you ask the TM enough questions, you may get it to melt down by trying to answer a Cretan paradox: *If I say this is true then it is false, but if I say it is false then it is true.*

Before feeding this fugue to the TM it will be helpful to know that Turing had two kinds of Truth Machines: one with fixed intelligence, and another that could learn new things as it went along. Turing called the first an *automatic machine*. It feels like it never needs help, so it never asks for it. This type of machine automatically gives an answer based upon what it knows and believes to be all that it needs to know (like some people we know).

Turing called the other machine, the one that could learn new things, an *oracle machine*. He considered it to be more powerful because it combined its own knowledge with that of its oracle. While the oracle machine was more powerful, it was also more prone, by virtue of its power, to reveal its inconsistencies.

When an oracle machine encounters an undecidable proposition it says, "I feel so stupid because I can't answer that question; please teach me how to answer it." You, being a generous oracle, supply the machine with new information to which it responds, "Ah, now I understand," then answers the question.

In supplying the machine with more information, you have just made it more complex. According to Gödel and Turing, the more complex you make it, the more the machine will respond with, "I feel so stupid." In other words, an oracle machine is like a wise man: the more it knows, the slower it is to answer, and the quicker it is to ask thoughtful questions.

A Symbolic System

In the following exercise we'll illustrate how difficult it is to teach a computer to recognize the transformations of a fugue's subject. This will help us to realize the richness of fugue as a symbolic system and how the hearing of everything in it requires a complex skill set. In showing how difficult it is to create an artificial intelligence, we'll renew our appreciation for the endowments of perception we enjoy.

Here are the rules. We'll call the computer UFM for Universal Fugue Machine. Our programming routine will have two kinds of statements: those that test UFM and those that teach it. All testing statements will begin with "UFM," to which UFM must respond with *true* or *false*. These are the only answers that UFM may give to a testing statement.

Statements that do not begin with "UFM" are teaching statements. They make UFM more complex. Because UFM is an oracle machine we'll allow it to

respond to seemingly contradictory teaching statements with, *teach me*. In these cases we'll reconcile the apparent contradiction with additional information. We'll stipulate that UFM can hear sounds and has a base knowledge of music theory: pitch names, keys, triads, etc.

The last rule is that we must at all cost avoid testing UFM with an undecidable proposition. To answer such a proposition truthfully, UFM would be required to tell a lie. Since no input from the oracle can avert this dilemma, such a statement will cause UFM to melt down. Are you ready? Here we go.

This is the subject.

Click the statement so that UFM can listen to it. By hearing it, UFM will remember that ~~c#-e#-c#-g#~~ in the octave below middle-C, in eighth notes, in the order of root-third-root-fifth of a Major triad, and an up-down-up contour of precisely these intervals, is the subject. We are now ready to test UFM with...

UFM will say that this is the subject.

...to which UFM replies, *true*. So far so good! Now let's verify that UFM has integrated its knowledge of this fugue with what it already knows about music theory.

UFM will say that the subject is a major triad.

UFM will say that the subject's intervals are
two Major thirds followed by a Perfect fifth.

UFM indicates that both of the above are *true* statements. Since we know that a fugue continuously restates the subject, we're curious to know if UFM now can hear other statements that may exist in this fugue. Fully expecting UFM to reply that it is *true*, we posit:

UFM will say that this is NOT
the only statement of the subject.

UFM listens to the fugue and replies, *false!* Somewhat startled, we perform our own scan and discover that UFM has told the truth. All other statements of the subject involve different pitches, octaves, triadic (or non-triadic) qualities, durations, contours, or intervals. Realizing that UFM interpreted our first statement too literally, we hint that it should loosen up by telling it:

This is NOT the only statement of the subject.

Sensing that it has more to learn, UFM replies, *teach me*. We respond by

letting UFM hear two instances of the subject that differ from the first by only one parameter.

The soprano of m. 4 states the subject.
The bass of m. 17 states the subject.

Upon hearing these, UFM infers that register is an invalid parameter and that the subject's last duration may be longer than an eighth note. Realizing that nearly every statement of the subject employs variation, and that we must teach UFM each one (all the timeline represents in copper brown), we begin introducing them one by one. With each introduction UFM tries to unravel the essential connection between them.

The soprano of m. 1 states the subject.

UFM immediately recognizes that this is not a major triad — a quality it had thought to be essential. Accordingly it responds with, *teach me*. We tell UFM that some subjects are triads, but others are not. UFM abandons the triad and briefly considers intervals. Since the last interval in the soprano of m. 1 (a P4) is not analogous to the first statement (a P5), it jettisons the interval as the defining trait. UFM concludes that the essential subject must be found in what remains: contour or rhythm.

First let us refine UFM's understanding of the subject's rhythm. UFM already knows that the last duration may be altered, but it does not realize that there is another technique involving proportionality. We tell UFM that:

The bass at the end of m. 19 states the subject.

The bass of m. 27 states the subject.

With this information UFM concludes that the subject may employ any durational set that is proportional to the first. It is now prepared to recognize subject diminutions and augmentations. We're confident that UFM will recognize the tenor of m. 25 as an augmentation, so we don't bother to test it.

Having discarded pitch classes, intervals, triads, and durational identity as comprising the subject's essence, UFM is left with contour. But this too must be refined. We expand the complexity of UFM's program by telling it that:

The soprano of m. 15 states the
subject in two consecutive iterations.

Here UFM learns that the subject's contour may be inverted. Whereas the original went up-down-up, the soprano of m. 15 went down-up-down. UFM is now smart enough to evaluate the following:

UFM will say that these are subject inversions
with diminution.

...to which UFM responds, *true*. But this fugue contains yet another trick that we must teach UFM. We tell it that:

The bass of m. 22 states the subject.

Here UFM pauses to think. It immediately perceives a diminution. It also recognizes that the interval order in the bass of m. 22 is the retrograde of the original. Whereas the original employed two intervals of the third followed by a fifth, this one is a fifth followed by two thirds. But UFM is stymied by the identical contour (up-down-up) of the two. This is because UFM knows that retrogradation also inverts the contour. Accordingly it cannot understand how the bass of m. 22 having the same contour as the original can be related. It begs, *teach me*. Our reply confirms that: inversion after retrogradation cancels the inversion effected by the retrogradation, thereby restoring the original contour.

UFM is now equipped to recognize a retrograde-inversion of the subject. The final piece in the puzzle is to teach UFM that a *false subject* states the first three pitches but deviates from any known aspect of the subject on its fourth.

The tenor of m. 19 states a false subject.

Now we're ready to put UFM to the acid test. Whereas we know that it can identify every subject, does UFM know that it can do this? We'll test UFM's self-awareness by feeding it a statement that, unlike the others, requires evaluation of its logic by means of the same. If UFM can answer this question then it will be aware of the limits of its own knowledge. So we require UFM to answer the following: true or false.

UFM knows that it doesn't know if it can
identify every statement of the subject.

UFM begins to churn and whirl as it ponders whether the statement is true or false. After a minute UFM begins to emit an acrid odor and we observe a wisp of smoke. After two minutes billows of particulates have filled the room and UFM has melted into a puddle of imaginary plastic. Why? Here is what UFM thought.

How else can I know that I don't know every instance except by knowing every instance? By answering true, I will have implied that I know every instance, and the statement is therefore false. On the other hand, by answering false, I will have implied that I don't know if I don't know, and the statement is therefore true. Either way I'm stuck!

So UFM self-destructs. Any requirement that it evaluate its logic by means of that logic is recursive, therefore undecidable. While UFM can evaluate the fugue based on parameters that the oracle provides, it cannot know, unless the oracle tells it, if said parameters are complete.

Self-Referential Ways of Knowing

Gödel's proof has metaphysical and cosmological implications. If the universe is a complex system (which it obviously is), aspects of which are expressible in arithmetic (which they obviously are), then any attempt to explain it by reference to what exists in the universe will contain undecidable propositions. This is called the *self-referential paradox*. All self-referential knowledge is incomplete — like a dictionary that defines a word by using that word in its definition.

The grander implication of Gödel's proof is that we can know something to be true, but not be able to prove it. This is because every proof employs what exists in the universe. All empirical methods are models of the universe they seek to explain, therefore prone to the self-referential paradox.

Because science, math, and logic are part of the universe, therefore smaller than it, the truths they provide will be incomplete. Making them more complete requires that they be made more complex, which makes them more prone to revealing their inconsistencies.

But we are here to discuss music, not the grand scheme of the universe. Here too Gödel's theorem applies. Like math and logic, music is a symbolic system. Musical forms (fugue, sonata-allegro, etc.) represent facets of that *system*, any expression of which is a *model* of the form. Analytical methods like Schenkerian or pitch-class set analysis also symbolize the system.

Beethoven wrote 32 piano sonatas, many movements of which are in sonata-allegro form. Each of these movements (the music itself) is a model of sonata-allegro. Each one symbolizes sonata-allegro. (Note the synonymous use of *model* and *symbol*.) Every Schenkerian graph revealing the fundamental structure of a sonata-allegro is therefore a symbol of a symbol of the system.

The fugue is one of music's most complex of forms. Bach wrote a cycle of them in each key in order to model that form, no single fugue of which could capture every possibility. With each new fugue the form was revealed to be more complex, therefore (applying Gödel) more incomplete. As Bach continued to compose, the floodgates opened to him and his conception of fugal models evolved. The more complex they became, the more incomplete the system was perceived to be — a measure of the fugue's depth as a symbolic system.

It is revealing that Bach himself did not call the repeated cycle "Book II" (as we do today). The first cycle he titled, *The Well-Tempered Clavier, or preludes and fugues in every key....* The second he called *Twenty-four Additional Preludes and Fugues*. Technically this *addition* (Book II) is not the *Well-Tempered Clavier*, but in addition to it. This is like the maraschino cherry on top of your banana split. It is not the banana, but in addition to it. This suggests that, having completed the *Well-Tempered Clavier* (Book I), Bach believed that he had not adequately modeled the system, hence the addition.

What facet of fugue might Bach have missed? Did he himself even know? If he knew, why didn't he fill the gap? If you ask me, I would say that he probably had an intuition of something more, but couldn't pin it down (or ran out of time). Maybe he conceived of fugue as an inexhaustible system that no amount of

modeling could express — i.e. perpetually incomplete.

Bach anchored his fugues to the diatonic system — major and minor keys. The chromatic layout of the *WTC* is so essential to it that keyality itself appears to have been integral to his notion of fugue. While the layout is correctly interpreted to be the solution to a problem of tuning, it is more than that. It represents Bach's apparent belief that tonality is part of the fugal essence.

Like Bach's *Additional Preludes and Fugues* (Book II), similar cycles by Hindemith and Shostakovich represent invaluable additions to the system. Although they are justly recognized as paying tribute to Bach, they are much more than that. They reveal an incompleteness of Bach's model; it had not expressed the universe of fugue. Hindemith and Shostakovich revealed that tonality is not coequal to motive, the true essence of the form. They did this by modeling the fugue in music of doubtful tonality.

The fugal cycles of Hindemith and Shostakovich are like Turing's response to Hilbert, or Gödel's response to Russell. As comprehensive as was Bach's *Well-Tempered Clavier*, if not responded to, it would have left the impression that a fugue must be centered in a key. Hindemith and Shostakovich demonstrated that tonal centeredness was not essential.

Another lesson we may draw is the proximity of fugue to logic, math, and science — especially logic. There is a reason why so many scientists and mathematicians have been drawn to it; they like its logic. They also hear in it a tonal analogue of their own disciplines. As we saw in the C Major fugue of Book II, Carl Sagan was one of the fugue's most ardent admirers. He liked that fugue so much that he shot a recording of it into space so that any aliens out there might enjoy it too.

Bach also believed in a connection between music, math, and science. He revealed it in his association with Gottfried Leibniz, the greatest mathematician of his day. Leibniz is remembered by musicians for his assertion that, "Music is the hidden arithmetical exercise of a mind unconscious that it is calculating" (as quoted in Mizler's *Musikalische Bibliothek*). The link between Bach and Leibniz was Lorenz Christoph Mizler, who studied briefly with both men, both of whom later joined his society for musical sciences.

Today Leibniz is also recognized as the first person to imagine the possibility of a computer. His invention of a symbolic system for the notation of polynomial equations in calculus provided the foundational logic that computers use today.

Martin Davis (U. C. Berkeley) has paid tribute to the contribution of Leibniz in: *The Universal Computer: The Road from Leibniz to Turing*. Davis writes that Leibniz dreamed of, "an encyclopedic compilation, of a universal artificial mathematical language in which each facet of knowledge could be expressed, of calculational rules which would reveal all the logical interrelationships among these propositions. Finally, he dreamed of machines capable of carrying out calculations, freeing the mind for creative thought" (p. 4).

I would draw your attention to Davis's use of the word *artificial*. Leibniz's vision was of more than a computer; he dreamed of *artificial intelligence*. While Leibniz's dream of the computer has been realized, artificial intelligence has not.

We began this analysis with a reference to Douglas Hofstadter's *Gödel*,

Escher, Bach: an Eternal Golden Braid — a book about computer programming and artificial intelligence. The difference between the visions of Leibniz and Hofstadter is that Gödel and Turing intervened. Because of their work we know that artificial intelligence will never be as intelligent as what it purports to model. It will always be incomplete. The bigger the programs and computers get, the more aware we will become of that incompleteness.

For these reasons Hofstadter concluded that Leibniz’s dream of artificial intelligence would be difficult, if not impossible, to achieve. In the twenty-three years since making this assessment, Hofstadter has been proven wrong but once. In 1997 a computer called “Deep Blue” defeated Garry Kasparov in chess, a defeat that Hofstadter predicted would not happen (which illustrates the truth of John Neumann’s adage that, *in mathematics you don’t understand things, you just get used to them*).

Versus Tacit and Truthful Knowing⁵

Let us rise, if we can, to the summit of the highest intelligence:
for *there* reason will see what in itself it cannot see.

Boethius: Prose 5, Book V
*The Consolation of Philosophy*⁶

Remember the quotation from Einstein at the beginning of this essay: “The human mind has first to construct forms, independently, before we can find them in things.” This amazing statement admits that the self-referential paradox lurks behind every formal system. It crouches there, waiting to melt you down. It explains why so often we fall prey to the fallacy of the antecedent confirming the consequent. In situations where we want to find something badly enough, our minds are prone to construct the desired object then confirm that it exists by “finding” it, sometimes where it truly does not exist.

But there is another way of knowing that does not involve self-reference. In his book by the same title, Steven Garber calls this type of knowing: *Tacit Knowing, Truthful Knowing*. “Tacit” means that it is: “silent, unspoken, quiet, implied without being openly expressed or stated, understood” (OED). Garber pays tribute to Polanyi’s *Personal Knowledge: Towards a Post-Critical Philosophy* (University of Chicago Press, 1972) as having been helpful in the formation of his own thinking.

A chemist by profession, Michael Polanyi (1891-1976) made important contributions to philosophy as well. In his research Polanyi was skeptical of reductionist tendencies, advocating instead that logic and facts do not convey the

⁵ I wish to acknowledge Ken Myers’s interview with Steven Garber in Volume 42 of the *Mars Hill Audio Journal* as helpful in the organizing of my own thoughts in the last section of this analysis.

⁶ Boethius, *Consolatio Philosophiae*, trans. Richard H. Green (Mineola: Dover Publications, 2002), 103.

essence of what is known. Knowledge is the result of what he called a “fiduciary framework” of: “tacit assent and intellectual passions, the sharing of an idiom and of a cultural heritage, [and an] affiliation to a like-minded community.”

By “tacit assent” Polanyi implied that there are ways of knowing that we don’t have to work at, logically deduce, or experiment with, in order to “know.” This type of knowing is the gift of our community, our religions, and very likely, our genes. It is like the instinctual behavior of the Monarch butterfly, every other generation of which migrates thousands of miles between the Americas. The generation that migrates is not born of parents that did. This knowledge, not learned, is innate to the creature. It is the same knowledge by which every culture, in every time, has known that it isn’t good to cause another person pain, it isn’t right to steal from somebody, it isn’t good to tell a lie, it isn’t good to cheat on your wife, it isn’t good to murder somebody.⁷

Garber points out that Polanyi advocated the integration of all ways of knowing — faith (a revealed knowledge) with experience. He argued that we should reject the Enlightenment assumption that reason provides the only basis for “reasonable” truth.

This is the very truth that Gödel established: there are truths that reason has not enough reason to know. Polanyi’s ideas are eminently compatible with Gödel’s: we can know something that we cannot prove. The writings of both men stipulate that science, math, and logic (and music) are not the final arbiters of truth and that rational thought will never absorb the ultimate truth.

⁷ In *The Abolition of Man*, C. S. Lewis refers to the vast body of things-we-know-but-can’t-prove as “the Tao.” Reading Lewis’s *Abolition* (and listening to Bach) will fix one of the modern fixation upon what Mike Rogers calls “tacit assumptions of rationalistic inquiry” and “conventional quantitative values” (*Journal of Music Theory Pedagogy* Vol. 17, p. 15). Bach and Lewis demonstrate how proof is not coequal to truth, and value not synonymous with numbers. Indeed, the most valued truths of the human experience cannot be quantified or proved.