# Images, Representations, and Programming

It’s an old idea.  Someone you know holds up a photograph depicting something familiar, say a beautiful car, maybe a Corvette, and asks you “what this?”.  You answer, “it’s a Corvette,” and are greeted with the cheeky response, “No! silly, it’s a picture.”

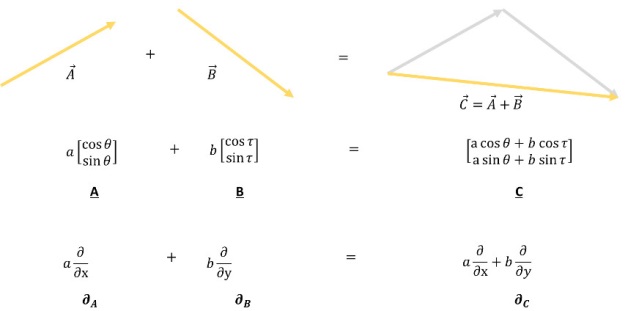
This simple joke, while annoying, makes an important philosophical point about keeping a clear distinction between the image of a thing and the thing itself. As important as this distinction is for basic reasoning and logic, it is much more important to keep it straight in the practice of mathematics and computing – particularly in the study of vectors.

Formally, a vector is any kind of object that belongs to a class of like objects that all ‘obey’ a set of rules that define how they combine to form new objects also in the same class. For simplicity, a vector will be denoted in underlined, bold face but, as will discussed below, there are other common ways to denote the vectors, all of which suffer from ‘Corvette-problem’ above. The set of combination rules are:

1. There is a combination rule ‘+’ such that **U** + **V** is a vector if **U**, **V** are vectors
2. **U**  + **V** = **V** + **U** (order doesn’t matter)
3. **U** + (**V** + **W**) = (**U** + **V**) + **W** (the combination rule is associative)
4. **0** + **U** = **U** + **0** = **U** (there is a zero vector)
5. **U** + (**-U**) = **0** (there is a way to add up vectors to get a zero one)
6. There is a scaling rule such that the product k**U** is a vector, (k is an ordinary complex number)
7. k(**U** + **V**) = k**U** + k**V**
8. (k+l)**U =** k**U** + l**U** (where k & l are ordinary complex numbers)
9. k(l**U) =** (kl)**U**
10. 1**U** = **U**

Some purist out there may object and point out that only occasionally does an author actually enumerate all 10 items above separately (even though such a purist will concede that all 10 must be there in some form or another). The purist may also go on to say that some authors prefer 0**U** = **0** to rule #5. But none of these details are particularly important.

What is important is that the rules are abstract and simple. They apply equally well to vectors defined as a directed arrows as they do to vectors defined as column arrays of numbers as they do to vectors defined in terms of partial derivatives. They apply equally well to vectors that we can observe and touch, for example pulls and pushes on an object, as they do to those that live in an abstract space like column arrays or partial derivatives, whose sole existence is built from ideas in the mind and symbols on the page.



As the study of vectors deepened, several clarifying points made computation with them very simple. The most powerful, and hence most dangerous, realization is the point that an arbitrary vector can be decomposed in terms of primitive vectors, usually referred to as basis vectors. This realization, which arguably finds its crystallization in the work of Descartes, reduces the infinity of possibilities into a manageable number of chunks and is the driving force between the 10 rules listed above.

The manageable chunks consist of a set of basic vectors whose number equals the number of dimensions in the space (1 for a line, 2 for a plane, 3 for a volume, and so on) and a list of the numbers whose length also equals the number of dimensions.

And here is the first of the traps. Once the basis vectors are agreed-upon and understood, they can be pushed to the back and the list (called a list of components) can be manipulated without much additional thought. The list becomes a stand-in for the original object in analogy for the way that the image of the car becomes a stand-in for the car itself. The list is now a representation of the original object.

This blurring between the original object and its representation becomes even more fuzzy with some additional reflection. A list is also a valid choice as an original object in the vector space since it also obeys the 10 rules (with the appropriate definition of ‘+’ and ‘x’). To show how strange this is in the physical world, consider the possibility of getting into the picture of the car, kicking over its motor, and taking it for a spin.

It’s no wonder that otherwise well-trained and intelligent people get hung up over vectors and their manipulations each and every day. Functionally, every object shown in the figure above is equivalent to a list of numbers.

Now suppose that one wanted to represent these abstract objects in a computer language. Well, as long as one was careful, one could actually exploit these ambiguities and simply say that the list will always be the representation. This is actually what most, if not all, languages do, though they differ in the terminology, with many choosing array, some choose vector, and others stay with list.

Of course, most users aren’t careful about maintaining that distinction and, I suppose, most aren’t even really conscious of it. But one hopes that at least the language creators do.

In most cases, this hope is realized. Many languages make the distinction between a heterogeneous list (not a vector) and a homogeneous list (which is, or at least can be, a vector). Some languages, like those underlying the computer algebra system Maple, use the word vector to connote a special kind of list. However, sadly, sometimes a language gets befuddled and either loses these distinctions or creates ones where none exist.

An example of the later problem comes from the numpy/scipy family of packages used in the Python programming language. To properly discuss this minor defect in what is really a great set of packages, I need to add one more ingredient that adds a few more ingredients to the vector space turning it into a metric space.

In a metric space, there is added to the original 10 rules an additional notion of the length of a vector. A new combination rule, usually denoted with a dot ‘.’, allows for two vectors to be combined to produce not another vector but a number, specifying how much of the length of one of the two lies along the other. This combination is defined such that **A**.**B = B**.**A** and that **A**.**A** is the square of the length of **A**. This combination rule is called variously as a dot product, an inner product, or a scalar product.

Once defined, another operation can be derived from these 11 rules. This operation, called the cross product, mixes the components from various places in the list to get new components. It depends on the dot product to bring meaning to the idea of having the component from one dimension multiplying the component from another dimension and, like the dot product, actually results in an object that doesn’t (properly) belong in the vector space. In other words, both the dot and the cross products take two vectors and produce something different.

In addition, both rules belong to the space itself since they both apply to any two pairs of vectors. Unfortunately, the numpy/scipy team missed this concept entirely.

In numpy, the vector space as a whole can be thought of as being represented by the family of functions that make up numpy proper. These functions include the function ‘array’ for making a new array and the function ‘cross’ for taking the cross product. Strangely, the function ‘dot’ is not found in the collection of numpy functions but rather is a member function of the ‘array’ object itself. A minor flaw in a really fine set of packages but a solid proof that it isn’t always easy to the tell the image from the thing.

# Balance and Duality

There is a commonly used device in literature that big, important events start small. I don’t know if that’s true. I don’t know if small things are heralds of momentous things but I do know that I received a fairly big shock from a small, almost ignorable footnote in a book.

I was reading through *Theory and Problems in Logic*, by John Nolt and Dennis Rohatyn, when I discovered the deadly aside. But before I explain what surprised me so, let me say a few words about the work itself. This book, for those who don’t know, is a Schaum’s Outline. Despite that, it is actually a well-constructed outline on Logic. The explanations and examples are quite useful and the material is quite comprehensive. I think that the study of logic lends itself quite nicely to the whole approach of Schaum’s since examples seem to be heart of learning logic and the central place where logicians tangle is over some controversial argument or curious sentence like ‘this sentence is false’.

As I was skimming Nolt and Rohatyn’s discussion about how to evaluate arguments I came across this simple exercise

<Is the argument below deductive?

Tommy T. reads *The Wall Street Journal*

∴ Tommy T. is over 3 months old.

-Theory and Problems in Logic, Nolt and Rohatyn>

Their answer (which is the correct one) is that the argument above is not deductive. At the heart of their explanation for why it isn’t deductive is the fact that while it is highly unlikely that anyone 3 months old or younger could read *The Wall Street Journal*, nothing completely rules it out. Since the concept of probability enters into the argument, it cannot be deductive.

So far so good. Of course, this is an elementary argument so I didn’t expect any surprises.

Nolt and Rohaytn go on to say that this example can be made to be deductive by the inclusion of an additional premise. This is the standard fig-leaf of logicians, mathematicians, and, to a lesser extent, scientists the world over. If at first your argument doesn’t succeed, redefine success by axiomatically ruling out all the stuff you don’t like. Not that that approach is necessarily bad; it is a standard way of making problems more manageable but usually causes confusion in those not schooled in the art.

For their particular logical legerdemain, they amend the argument to read

< All readers of *The Wall Street Journal* are over 3 months old.

Tommy T. reads *The Wall Street Journal*

∴ Tommy T. is over 3 months old.

-Theory and Problems in Logic, Nolt and Rohatyn>

This argument is now deductive because they refuse to allow the possibility (no matter how low in probability) that those amongst us who are 3 months old are younger cannot read *The Wall Street Journal*. They elevate to metaphysical certitude the idea that youngsters such as they can’t by simple pronouncement.

Again there are really no surprises here and this technique is a time honored one. It works pretty well when groping one’s way through a physical theory where one may make a pronouncement that nature forbids or allows such and such, and then one looks for the logical consequences of such a pronouncement. But a caveat is in order. This approach is most applicable when a few variables have been identified and/or isolated as being the major cause of the phenomenon that is being studied. Thus it works better the simpler the system under examination is. It is more applicable to the study of the electron than it is to the study of a molecule. It is more applicable to the study of the molecule than to an ensemble of molecules and so on. By the time we are attempting to apply it to really complex systems (like a 3-month old) its applicability is seriously in doubt.

Imagine then, my surprise by the innocent, little footnote associated with this exercise that reads

<There is, in fact, a school of thought known as *deductivism* which holds that all of what we are here calling “inductive arguments” are mere fragments which must be “completed” in this way before analysis, so there are no genuine inductive arguments

-Theory and Problems in Logic, Nolt and Rohatyn>

Note the language used by the pair of logicians. Not that the deductivism school of thought wants to minimize the use of inductive arguments or maximize the use of deductive ones. Not that its adherents want to limit the abuses that occur in inductive arguments. Nothing so cautious as that. Rather the blanket statement that “there are no genuine inductive arguments.”

A few minutes of exploring on the internet led me to slightly deeper understanding of the school of deductivism but only marginally so. What could be meant by no genuine arguments? A bit more searching led me to some arguments due to Karl Popper (see the earlier column on [Black Swan Science](http://aristotle2digital.blogwyrm.com/?p=64)).

[These arguments, as excerpted from Popper’s *The Logic of Scientific Discovery*](http://www.csus.edu/indiv/m/merlinos/sci/popperDeduct.html), roughly summarized, center on his uneasiness with inductive methods as applied to the empirical sciences. In his view, an inference is called inductive if it proceeds from singular statements to universal statements. As his example, we again see the black-swan/white-swan discussion gliding to the front. His concern is for the ‘problem of induction’ defined as

<[t]he question whether inductive inferences are justified, or under what conditions...

-Karl Popper, The Logic of Scientific Discovery>

Under his analysis, Popper finds that any ‘principle of induction’ that would solve the problem of induction is doomed to failure since it would necessarily be a synthetic statement, not an analytic one. From this observation, one would then need a ‘meta principle of induction’ to justify the principle of induction and a ‘meta-meta principle of induction’ to justify that one and so on, to an infinite regress.

Having established this initial work, Popper jumps into his argument for deductivism with the very definite statement

<My own view is that the various difficulties of inductive logic here sketched are insurmountable.

-Karl Popper, The Logic of Scientific Discovery>

And off he goes. By the end, he has constructed an argument that banishes inductive logic from the scientific landscape, using what, in my opinion, amounts to a massive redefinition of terms.

I’ll not try to present anymore of his argument. The interested reader can follow the link above and read the excerpt in its entirety. I would like to try to ask a related but, in my view, more human question. To what end is all this work leading? I recognize that it is important to understand how well a scientific theory is supported. It is also important to understand the limits of knowledge and logic. But surely, human understanding and knowledge are not limited by our scientific theories nor are they adequate described by formal logic. Somehow, human understanding is a balance between intuition and logic, between deduction and induction.

Popper’s critiques sound too much like the sounds of someone obsessing over getting the thinking just so without stopping to ask if such a task is worth it. Scientific discovery happens without the practitioners knowing exactly how it happens and what to call each step. Should that be enough?

Of course, objectors to my point-of-view will be quick to point out all the missteps that logicians can see in the workings of science – all the black swans that fly in the face of a white-swan belief. My retort is simply “so what?”

Human existence is not governed solely by logic nor should it be. If it were, a part of the population would be frozen in indecision because terms were not defined properly, another part would be stuck in an infinite loop, and the last part would be angrily arguing with itself over the proper structure. There is a duality between induction and deduction that works for the human race – a time to generalize from the specific to the universal and a time to deduce from the universal to the specific.

Perhaps someday, someone will perfect deductivism in such a way so that scientific discovery can happen efficiently without all the drama and controversy and uncertainty. Maybe… but I doubt it. After all, we know that we humans aren’t perfect – why should we expect one of our enterprises to be perfectible?

# The Power of Imagination

A couple weeks ago, I wrote about the subtle difficulties surrounding the mathematics and programming of vectors. The representation of a generic vector by a column array made the situation particularly confusing as one type of vector was being used to represent another type. The central idea in that post was that the representation of an object can be very seductive; it can cloud how you think about the object, or use it, or program it.

Well this idea about representations has, itself, proven to be seductive and has lead me to think about the human capacity that allows imagination to imbue representations of things with a life of their own.

To set the stage for this exploration, consider the well-known painting entitled *The Treachery of Images* by the French painter Magritte.



The translation of the text in French at the bottom of the painting reads “This is not a pipe.” Magritte’s point being that the image of the pipe is a representation of the idea of a pipe but is not a pipe itself; hence his choice of the word ‘treachery’ in the title of his painting.

Of course, this is exactly the point I was making in my earlier post but a complication in my thinking arose that sheds a great deal of light on the human condition and has implications for true machine sentience.

I was reading Scott McCloud’s book *Understanding Comics* when he presented a section on what makes sequential art so compelling. In that section, McCloud talks about the inherent mystery that allows a human, virtually any human old enough to read, to imagine many things while reading a comic. Some of the things that the reader imagines include:

* Action takes place in the gutters between the panels
* Written dialog as actually being spoken
* That strokes of pencil and pen and color are actually things.

You, dear reader are also engaging in this kind of imagining. The word you are reading – words that I once typed – are not even pen and pencil strokes on a page. The whole concept of page and stroke, is, of course, virtual; tracings of different flows of current and voltage humming through micro-circuitry in your computer.

Not only is that painting of Magritte’s shown above not a pipe, it’s not a painting. It is simply a play of electronic signals on a computer monitor and a physiological response in the eye. And yet how is that it is so compelling?

What is the innate capacity of the human mind to be moved by arrangements of ink on a page, by the juxtaposition of glyphs next to each other, by movement of light and color on a movie screen, by the modulated frequencies that come out of speakers and headphones? In other words, what is the human capacity that breathes life into the signs and signals that surrounds us?

Surely someone will rejoin “it’s a by-product of evolution” or “it’s just the way we are made”. But these types of responses, as reasonable as they may be, do nothing to address the root faculty of imagination. They do nothing to address the creativity and the connectivity of the human mind.

As a whimsical example, consider this take on Magritte’s famous painting, inspired by the world of videogames.



Humans have that amazing ability to connect to different ideas by some tenuous association to find a marvelous (or at least funny) new thing. The connections that lead from the ‘pipe’ you smoke to the virtual ‘pipe’ in Mario Brothers are obvious to anyone whose been exposed to both of them in context. How can we come to understand it and, perhaps, imitate it?

Maybe we really don’t want machines that actual emulate human creativity but we won’t know or understand the limitations of machine intelligence without more fully exploring our own. And surely one vital component of human intelligence is the ability to flow through the treachery of images into the power of imagination.

# Fallacies, Authority, and Common Sense

Logical fallacies are everywhere. Just make a search using the string ‘logical fallacies list’ (e.g. [here](https://en.wikipedia.org/wiki/List_of_fallacies), [here](http://www.logicallyfallacious.com/index.php/logical-fallacies), and [here](http://utminers.utep.edu/omwilliamson/ENGL1311/fallacies.htm)) and you’ll come across many lists citing many fallacies that an arguer can employ and why they are wrong, bad, or otherwise socially unacceptable. The authors of such lists argue that it is desirable when crafting a valid argument to avoid as many as possible and when consuming an argument to be sensitive to their presence.

And yet the number of fallacies in day-to-day discourse never seems to diminish. So clearly people aren’t getting the message.

Of course, not everyone making an argument is really interested in making their argument valid. Certainly, politicians are interested more in getting votes or passing their particular bills into law than they are ever interested in logic and logical fallacies. Advertisers also bend the rules of good logic to make their product stand out so that potential customers will select their product over a competitor’s. So people who fall into these classes reject the message because embracing it would compromise their goals.

But there is another facet worth considering as well. There is a possibility that people do get the message and simply reject it since they judge that the message itself is flawed. Is it possible that some people’s common sense allow them, perhaps unconsciously, to see that some arguments about fallacies are themselves fallacious? Is it possible that some people who argue about avoiding fallacies are engaging in fallacies about fallacies?

Now before I explain how some arguments about fallacies can be fallacious, I would like to clarify a couple points. First, I think the best definition is provided by the Stanford Encyclopedia of Philosophy which says that a fallacy is [deceptively bad argument](http://plato.stanford.edu/entries/fallacies/); an argument where the conclusion that does not follow from the premises being offered and that it is not manifestly obvious why. Second, that that definition, while being the best out there, is still fairly inadequate. The reason being that if one can detect the fallacy how deceptive is it actually? The point here is that the very concept of a fallacy is a slippery one and, in fact, there is substantial controversy about the nature of fallacies as can be seen [from the long discussion here.](http://plato.stanford.edu/entries/fallacies/#DouAboFal)

So for the sake of this post, I am going to argue that a fallacy is a bad argument that is deceptive for people who are not trained in detecting and correcting it.

Some fallacies are relatively easy to detect and fix. The simplest ones seem to originate in deductive reasoning. The following example of the [fallacy of the undistributed middle](https://en.wikipedia.org/wiki/Fallacy_of_the_undistributed_middle) comes from syllogistic logic:

All dogs have fur

My cat has fur

Therefore my cat is a dog

These types of errors are easy to see even if they are not easy to explain. These types of fallacies are benign because they aren’t very deceptive.

A much more common and truly deceptive fallacy comes in the form of equivocation, where the meaning of a terms changes mid-argument and, if one isn’t careful, one misses it and becomes either confused or, worse, convinced of an invalid conclusion.

When the argument is simple, equivocation fairly easy to find as in this example:

The end of life is death.  
Happiness is the end of life.  
So, death is happiness.

Clearly the word ‘end’ in the first line means the termination or cessation whereas the word ‘end’ in the second means goal or purpose. When the argument is much larger in length or involves an emotional subject it is much harder to detect equivocation. As an example on that front, I once read a blog post (unfortunately I can’t source it anymore) where the author was celebrating a story in which an Amish man boarded a bus and challenged the people onboard about television. As the story goes, the Amish man asked how many of the passengers had a TV and every hand went up. He then asked them how many of them thought TV was bad and almost every hand went up as well. He then asked why, if they thought it was bad, did they tolerate a TV in their homes. The blogger obviously didn’t notice or care that the definition of TV had changed from the first question, where it meant the device, to the second question, where it meant the programming. All that mattered was the emotional delivery.

Perhaps the trickiest kind of fallacy concerns appeals to authority. And it is in this case where we find fertile ground where grow the fallacy of fallacies.

An appeal to authority can actually be a reasonable thing to do when the dealing with custom, or policy, or doctrine. As long as the authority is proper, the appeal can be a solid piece in an argument. When the appeal is to the authority of the public or to someone whose motives are questionable, then the [appeal to authority becomes a fallacy](http://www.logicallyfallacious.com/index.php/logical-fallacies/21-appeal-to-authority).

That said, an appeal to authority is never valid when it comes to science. Nonetheless, it is a common place appeal offered by those who talk about ‘settled science’. They tell us that a scientific conclusion is valid based solely on the idea that ‘***X*** percent of the scientists in the world agree on proposition ***Y***’. They also tell us that anyone who objects is necessarily engaged in a logical fallacy by either ignoring a proper appeal to authority (the ***X*** percent of scientists who believe proposition ***Y***) or by making an incorrect appeal to authority (the ***100 – X*** percent of scientists who reject proposition ***Y***).

To my way of thinking, as a physicist, this type of argument goes against common sense and is just wrong. Consider the case in physics at the turn on the 20th century. A majority of scientists felt that mankind had basically all the rules in place. The science of mechanics was well understood in terms of Newton and his 3 laws and the science of electricity, magnetism, and optics had just been united by Maxwell. Sure there was this pesky little problem with the ultra-violet catastrophe, but the majority of scientists were willing to ignore this or believe that a small tweak was all that was needed to fix things. Of course, that ‘small tweak’ ushered in the science of quantum mechanics that forever changed the way we think about science and philosophy.

Now a careful reader may argue that I indulged in a logical fallacy of my own about the majority of scientists when I pronounced that they were willing to ignore or believe all that was needed was a small tweak. After all, was I there to interview each and every one of them? But that assessment is backed up by an overwhelming amount of evidence that shows that the advancement of Planck was surprise to the physics community.

So what to make of those ‘settled science’ folk? Well they seem to want to ignore the logic underpinning the scientific method by appealing to authority as if scientific conclusions are immutable as long as they are based on a kind of popularity. They also use the form and structure developed to explain logical fallacies as an additional appeal to authority (in this case to the community of logicians rather than scientists) to dismiss anyone who believes the contrary to their doctrine as being illogical. And here they commit a two-fold error. By failing to recognize that there is no certainty encompasses either science or logic in their entirety, these individuals use the machinery of avoiding fallacies as a logical fallacy itself. They look on those who support the doctrine as pure in motive and look upon those who reject it as either corrupt or unqualified and stupid. They heap on layer after layer of emotionalism while telling their critics that they are mired in emotional thinking.

Fortunately, it seems, the human mind has a built-in safety valve in the form of common sense that allows us to reject these fallacies of fallacies even if we don’t know why we do it. I suppose, intrinsic to the human condition, is a natural skepticism for just how far logic can takes us. After all, it is a tool not a god and we should treat it as such.

# A Heap of Equivocation

As I write this week’s entry to Aristotle to Digital, I am reflecting on the life and times of Yogi Berra, who just died at the ripe old age of 90. I fervently hope that he is resting in peace. In my opinion, he earned it.

In an earlier column, published about a year ago, I wrote about [Yogi Berra Logic](http://aristotle2digital.blogwyrm.com/?p=54) as I termed the legendary witticisms of one of the greatest catchers to have ever played the game of baseball. In tribute to his life and passing, I thought I would revisit that whimsical posting with some more thoughts on what made Yogisms have such timeless attraction and talk a little about some other playful uses of natural language.

Before I go deeply into these points, I would like to correct the record about Yogisms. Several people have used the word malapropisms to describe the various nuggets of thought that he would utter. This is an incorrect application of malapropism, which is defined as:

< **malapropism** - the mistaken use of a word in place of a similar-sounding one, often with unintentionally amusing effect, as in, for example, “dance a *flamingo*” (instead of *flamenco*). >

I’m not saying that Yogi never used a malapropism in his life. I am saying that most, if not all, of his Yogisms don’t fall into this category. Rather they fall into the category of equivocal speech. The meaning of words changes, often extremely quickly, from one part of the Yogism to another and one has to read them with the various contexts that they span in mind.

A host of Yogisms can be found at the Yogi Berra Musuem and Learning Center’s [list of Yogisms](http://www.yogiberramuseum.org/just-for-fun/yogisms/). To illustrate the point of equivocation in some detail take the Yogism

<The future ain’t what it used to be – Yogi Berra>

On the surface, this expression doesn’t seem to have any meaning and would surely throw any natural language analysis software for a loop in trying to assign one. And yet, there actually is at least a little meaning as evidenced by the smile, chuckle, chortle, snicker, or belly-laugh that each of us has as a reaction upon reading it.

But surface impressions are rarely more than skin deep (a Yogism of my own perhaps?) and with a bit of imagination we can easily parse out some meaning and, perhaps, even profound meaning. I base this expectation on the fact that Yogi Berra was not a stupid man by any measure – his accomplishments alone should testify to that – and that his Yogisms strike a chord in so many peoples mind.

There are, at least, two fairly poignant meanings that can be mined with a fair amount of confidence from the Yogism above. The first is that the hope and aspirations for the future that filled his head at a younger age are now replaced with far less hopeful ideas for what the future holds now that he has grown older. On other words, the next 20 years looked brighter to him when he was younger compared to how he perceives the same 20 year span into the future now that he is an older man. The second is that when he was younger, say 25 years old, and looking forward to what the world would offer when he was 40, he had huge dreams of what might come true. Now that he has turned 40, he’s found that ‘the future’ wasn’t as wonderful as he imagined it might be.

Notice the structure of this particular Yogism. It invokes these two ideas compactly and with humor in a way that a plainer and more logical composition that avoided equivocation cannot do. It’s a masterpiece of natural language if not of pure predicate logic and I think we should be thankful for that.

I don’t know with certainty but I suspect that the next example of natural language gymnastics would have likely captured Yogi’s fancy as well. It is known as the [continuum fallacy](https://en.wikipedia.org/wiki/Continuum_fallacy).

In the continuum fallacy, natural language is used to allow one to cross a fuzzy line without even knowing one is doing it. One form of the continuum fallacy (really the [sorites paradox](https://en.wikipedia.org/wiki/Sorites_paradox), but they are essentially the same thing) reads something like this.

We can all agree that 1,000,000 grains of sand can be called a heap

We can also agree that if we take 1 grain away from this heap, it’s still a heap

Then we can also agree that 999,999 grains of sand can also be called a heap

And in continuing in this fashion we can soon arrive at the idea that a heap of sand need not have any sand in it at all.

In the general explanations of why this line of argumentation is a fallacy, analysts will cite that reason being the vague nature of the definition of heap (vagueness of predicates). Certainly it is true that at some poorly undefined (or undefinable?) line exists between where the heap turns into a non-heap.

This ‘paradox’ is not confined to linguistics. Take the image below.

The color gradient from red to yellow is so gradual that it is hard to say that any single color is really different from its neighbors and yet red is not yellow and yellow is not red. And where does the orange begin and end?

This vagueness seems to be a [universal feature that is built into most everything](https://books.google.com/books?id=3_1snsgmqU8C&pg=PA1037&hl=en#v=onepage&q&f=false). And while it may throw linguists, logicians, perceptual psychologists, and computers into a tizzy, I suspect that Berra, the playful king of vagueness, would have had as much fun with this as with uttering Yogisms.

# Representing Time

Time is a curious thing. John Wheeler is credited with saying that

<Time is defined so that motion looks simple>

Certainly this is a common, if unacknowledged, sentiment that pervades all of the physical sciences. The very notion of an equation of motion for a dynamical system rests upon the representation of time as a continuous, infinitely divisible parameter. Time is considered as either the sole independent variable for ordinary differential equations or as the only ‘time-like’ parameter for partial differential equations. Regardless, the basic equation of the motion for most physical processes takes the form of

<\[ \frac{d}{dt} \bar S = \bar f(\bar S; t) \] >

where the state <$$\bar S$$> can be just about anything imagined. By its very structure, these generic form implies that we think of time as something that can be as finely divided as we wish – how else can there be any sense made of the <$$\frac{d}{dt}$$> operator.

Even in the more modern implementations of cellular automata, where the time updates occur at discrete instants, we still think of the computational system as representing a continuous process sampled at evenly spaced times.

The very notion of continuous time is inherited from the ideas of motion and here I believe that Wheeler’s aphorism is on target. The original definition of time is based on the motion of the Earth about its axis with the location of the Sun in the sky moving continuously as the day winds forward. As the invention of time keeping evolved, items, like the sundial, either abstracted the sun’s apparent motion to something more easily measured, or replaced that motion with something more easily controlled like a clock. Thus time for most of us takes on the form of the moving hands of the analog clock.



The location of the hands is a continuous function of time, with the angle that the hour and minute (and perhaps second) hand make with respect to high noon going something like <$$\sin(\omega t)$$> where the angular frequency <$$\omega$$> is take to be negative to get the handedness correct.

But as timekeeping has evolved does this notion continue to make sense? Specifically, how should we think about the pervasive digital clock



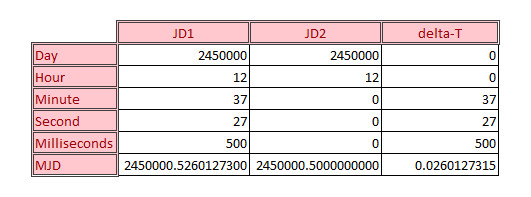
and the underlying concepts of digital timekeeping on a computer.

Originally, many computer systems were designed to inherit this human notion and time is internally represented in many machines as a double precision floating point number. But does this make sense – either from the philosophical view or the computing view?

Let’s consider the last point first. Certainly, the force models <$$f\bar(S\bar;t)$$> used in dynamical systems require a continuous time in the calculus but they clearly cannot get such a time in the finite precision of any computing machine. At some level, all such models have to settle for a time just below a certain threshold that is tailored for the specific application. So the implementation of a continuous time expressed in terms of a floating point variable should be replaced with one or more integers that count the multiples of this threshold time in a discrete way.

What is meant by one or more integers is best understood in terms of an example. Astronomical models of the motion of celestial objects are usually expressed in terms of Julian date and fractions therein. Traditional computing approaches would dictate that the time, call it $$JD$$ would be given by a floating point number where the integer part is the number of whole days and the fractional parts the numbers of hours, minutes, seconds, milliseconds, and so on, added together appropriately and then divided by 86400 to get the corresponding fraction. Conceptually, this means that we take a set of integers and then contort them into a single floating point number. But this approach is not only a form of unnecessary mental gymnastics but is actually quite wrong in a numerical sense.

Consider the following two modified Julian dates, represented by their integer values for days, hours, minutes, seconds, and milliseconds and by their corresponding floating point representations



In an arbitrary-precision computation, the sum of $$JD1 = JD2 + delta-T$$ would be exact but a quick scan over the last two digits of the three numbers involved shows that the floating point representation doesn’t capture the correct representation exactly.

Of course this should come as no surprise since this is an expected limitation of floating point arithmetic. The only way to determine if two times are equal using the floating point method is to difference the two times in question, take the absolute value of the result and to declare sameness if the value is less than some tolerance. Critics will be quick to point out that this fuzziness is the cost of gains having fact performance and that this consideration outweighs exactness but this is really just a tacit admission of the existence of a threshold time below which one does not need to probe.

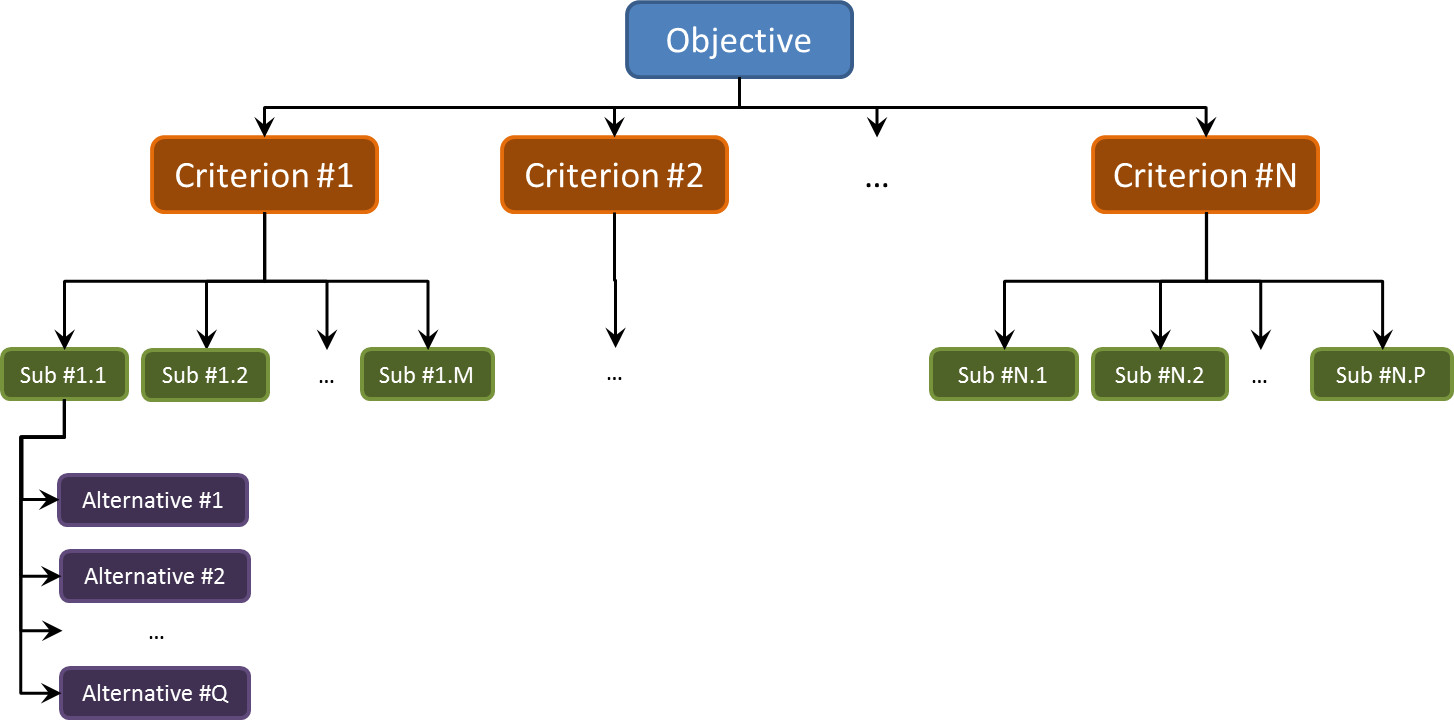
Arbitrary precision, in the form of a sufficient set of integers, circumvents this problem but only to a point. One cannot have an infinite number of integers to capture the smallest conceivable sliver of time. Practically, both memory and performance considerations limit the list of integers in the set to be relatively small. And so we again have a threshold time below which we cannot represent a change.

And so we arrive at the contemplation of the first problem. Is there really any philosophical ground on which we can stand that says that a continuous time is required. Certainly the calculus requires continuity at the smallest of scales but is the calculus truth or a tool? Newton’s laws can only be explored to a fairly limited level before the laws of quantum mechanics becomes important. But are the laws of quantum mechanics really laws in continuous time? Or is Schrodinger’s equation an approximation to the underlying truth? The answer to these questions, I suppose, is a matter of time.

# Making Rational Decisions

I recently came across an interesting method for combining qualitative and quantitative data on a common footing to allow for a mathematically supported framework for make complicated decisions where many criteria are involved. The method is called the [Analytic Hierarchy Process](https://en.wikipedia.org/wiki/Analytic_hierarchy_process).

The Analytic Hierarchy Process (AHP), which was invented Thomas L. Saaty in the 1970s, uses a technique based on matrices and eigenvectors to structure complex decision making when large sets of alternatives and criteria are involved and/or when some of the criteria are described by attributes that cannot be assigned objective rankings are in play. It is especially useful in group-based decision making since it allows the judgements of disparate stake-holders, often with quite different points-of-view, to be considered in a dispassionate way.

In a nut-shell, the AHP consists of three parts: the objective, the criteria, and the alternatives. Criteria can be sub-divided as finely as desired, with the obvious, concomitant cost of more complexity in the decision making process. Each alternative is then assigned a value in each criterion and each criteria is given a weighting. The assessments are normalized and matrix methods are used to link the relative values and weightings to give a ranking. Graphically, these parts are usually presented in hierarchical chart that looks something like:

A nice tutorial exists by Haas and Meixner entitled [An Illustrated Guide to the Analytic Hierarchy Process](https://mi.boku.ac.at/ahp/ahptutorial.pdf) and this posting is patterned closely after their slides. The decision-making process that they address is buying a car. This is the objective (‘the what’) that we seek to accomplish. We will use three criteria when selecting the car to buy: Style, Reliability, and Fuel Economy.

Two of these criteria, Style and Reliability, are qualitative or, at least, semi-qualitative, whereas the Fuel Economy is quantitative. Our alternatives/selections for the cars will be AutoFine, BigMotors, CoolCar, and Dynamix.

The first step is to make assign numerical labels to the qualitative criteria. We will use a 1-10 scale for Style and Reliability. Since we are weighing judgements, the absolute values of these scores are meaningless. Instead the labels indicate the relative ranking. For example, we can assume that the 1-10 scale can be interpreted as:

* 1 – perfectly equal
* 3 – moderately more important/moderately better
* 5 – strongly more important/strongly better
* 7 – very strongly more important/very strongly better
* 9 – extremely more important/extremely better

with the even-labeled values slightly greater in shading than the odd labels that precede them. This ranking scheme can be used to assign weightings to the criteria relative to each other (for example style is almost strongly more important than reliability – 4/1) and to weigh the alternatives against each other in a particular criteria (for example AutoFine is moderately better than CoolCar in reliability).

To be concrete, let’s suppose our friend Michael is looking to buy a car. We interview Michael and find that he feels that:

* Style is half as important as Reliability
* Style is 3 times more important as Fuel Economy
* Reliability is 4 times more important as Fuel Economy

Based on these responses, we construct a weighting table

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel Economy** |
| **Style** | 1/1 | 1/2 | 3/1 |
| **Reliability** |  | 1/1 | 4/1 |
| **Fuel Economy** |  |  | 1/1 |

where the first number in the entry corresponds to the row and the second to the column. So the ‘4/1’ entry encodes the statement that Reliability is 4 times more important as Fuel Economy. The omitted entries below the diagonal as simply the reverses of the one above (e.g. 1/2 goes to 2/1).

This table is converted to a weighting matrix $$\mathbf{W}$$ which numerically looks

\[ \mathbf{W} = \left[ \begin{array}{ccc} 1.000 & 0.5000 & 3.000 \\ 2.000 & 1.000 & 4.000 \\ 0.3333 & 0.2500 & 1.000 \end{array} \right] \; .\]

In a similar fashion, we interview Michael for his judgments of each automobile model with each criteria and find corresponding weighting matrices for Style and Reliability:

\[ \mathbf{S} = \left[ \begin{array}{ccc} 1.000 & 0.2500 & 4.000 & 0.1667 \\ 4.000 & 1.000 & 4.000 & 0.2500 \\ 0.2500 & 0.2500 & 1.000 & 0.2000 \\ 6.000 & 4.000 & 5.000 & 1.000 \end{array} \right] \; \]

and

\[ \mathbf{R} = \left[ \begin{array}{ccc} 1.000 & 2.000 & 5.000 & 1.000 \\ 0.5000 & 1.000 & 3.000 & 2.000 \\ 0.2000 & 0.3333 & 1.000 & 0.2500 \\ 1.000 & 0.5000 & 4.000 & 1.000 \end{array} \right] \; .\]

Finally, we rank the fuel economy for each alternative. Here we don’t need to depend on Michael’s judgment and can simply look up the [CAFE](ohttp://www.nhtsa.gov/fuel-economy) standards to find

\[\mathbf{F} = \left[ \begin{array}{c}34\\27\\24\\28 \end{array} \right] mpg \; .\]

Saaty’s method directs us to first find the eigenvectors of each of the $$4 \times 4$$ criteria matrices and of the $$3 \times 3$$ weighting matrix that correspond to largest eigenvalues for each. Note that the Fuel Economy is already in vector form. The L1 norm is used so that each vector is normalized by the sum of it elements. The resulting vectors are:

\[\mathbf{vW} = \left[ \begin{array}{c}0.3196\\0.5584\\0.1220 \end{array} \right] \; ,\]

\[\mathbf{vS} = \left[ \begin{array}{c}0.1163\\0.2473\\0.0599\\0.5764 \end{array} \right] \; ,\]

\[\mathbf{vR} = \left[ \begin{array}{c}0.3786\\0.2901\\0.0742\\0.2571 \end{array} \right] \; ,\]

and

\[\mathbf{vF} = \left[ \begin{array}{c}0.3009\\0.2389\\0.2124\\0.2479 \end{array} \right] \; .\]

A $$4 \times 3$$ matrix is formed whose columns are $$\mathbf{vS}$$, $$\mathbf{vR}$$, $$\mathbf{vF}$$ which is then left multiplied into $$\mathbf{vW}$$ to give a final ranking. Doing this gives:

|  |  |
| --- | --- |
| **AutoFine** | 0.2853 |
| **BigMotors** | 0.2702 |
| **CoolCar** | 0.0865 |
| **Dynamix** | 0.3580 |

So from the point-of-view of our criteria, Dynamix is the way to go. Of course, we haven’t figured in cost. To this end, Haas and Meixner recommend scaling these results by the cost to get a value. This is done straightforwardly as shown in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Ranking** | **Cost** | **Normalized Cost** | **Value = Ranking/Normalized Cost** |
| **AutoFine** | 0.2853 | $20,000 | 0.2899 | 1.016 |
| **BigMotors** | 0.2702 | $15,000 | 0.2174 | 1.243 |
| **CoolCar** | 0.0865 | $12,000 | 0.1739 | 0.497 |
| **Dynamix** | 0.3580 | $22,000 | 0.3188 | 1.122 |

With this new data incorporated, we now decided that BigMotors gives us the best value for the money. Whether our friend Michael will follow either of these two recommendations, is, of course, only answerable by him but at least the AHP gives some rational way of weighing the facts. I suspect that Aristotle would have been pleased.

# Technique and Judgment – Distinguishing ‘How’ and ‘What’

I suppose this column grew out a confluence of things that made me realize yet another limitation of the computers in their gallant attempt to unseat the human race in its mastery of the planet. This limitation, unfortunately, also impacts the human being first learning a new skill. What is this limitation you may ask – it’s the inability to distinguish technique from judgment. Fortunately humans can grow out of it, computers not so much (that is to say not at all).

On the face of it, this limitation seems to be a mismatching of concepts bordering on a non sequitur. After all technique is how one does whereas judgment centers on the ability to decide or conclude. What do the two have to do with each other?

To illustrate, let’s consider the average student in one of the STEM programs at a university. The student spends large amounts of time in mathematical courses learning the techniques associated with Calculus, Linear Algebra, Differential Equations, Vector Analysis and the like. A good student earning good grades succeeds at tests with questions of the sort:

“Given a vector field $$\vec f(x,y,z) = x^2 \hat \imath + \sin(y) \hat \jmath + \frac{1}{3 z^3) \hat k$$ compute the divergence $$\nabla \cdot \vec f(x,y,z)”

A successful completion of this problem leads to the answer $$\nabla \cdot \vec f(x,y,z) = 2x +\cos(y) - \frac{1}{z^4} $$ demonstrating that the student knows how to compute a divergence. To be sure, this skill and the others listed above, are important skills to have and are nothing to sneeze at, but they don’t take one far enough. Without the judgment of knowing what to do, the technique of how to do it becomes nearly useless.

To illustrate this, our student, having gotten straight A’s in all her subjects now moves onto an engineering class where she is asked to solve a problem in electricity and magnetism that says something like

“Given the following distribution of charges and arrangements of conducting surfaces, compute the electric field.”

Suddenly there is no specification on what technique to use, no indication how to solve the problem. All that is being asked is a ‘what’ – what is the electric field. Prudent judgment is needed to figure out how to solve the problem. And here we find the biggest stumbling block for the human (and a nearly insurmountable obstacle for current computing).

Lawvere and Schanuel, in their book Conceptual Mathematics: a first introduction to categories, summarize this distinction when they note

There will be little discussion about how to do specialized calculations, but much about the analysis that goes into deciding what calculations need to be done, and in what order. Anyone who has struggled with a genuine problem without having been taught an explicit method knows that this is the hardest part.

The distinction between the ‘what’ and the ‘how’, between the judgment needed to determine what and when to do a thing and the technique needed to perform the thing is often complicated and subtle. Much like the intricate interplay between language and thought the interaction between judgment and technique has no clean lines. In fact, viewed from a certain perspective, a technique can be thought of as the language of doing and judgment as the thought associated with what should be done. How we do a thing often affects what we think can be done.

* 

 !

"!#$

!$%&'#!!

'%!(#)!$

\*!+#

,$%%#

-.(%!#

/0

$/123/!#

4$

5-%$62$7-!'#-

8

9://&;;</\*=$!$ +>=?@<<A<\*!+<<(A<BCDBE ?E+<';;

/;;<%<</\*=$!$ +F

-%;;<%<</\*=$!$ +>@=A<\*!+G

<BCDB ?HD;;'/#%$

'%/$'/ $%!

'#('$

%!#

$$/"-%(/

'/%!!

'

9:/%$

&#F

1!&$/%#-

%'&#I(&

/%#-&%%%'\*!

%%+#

5/1$%'438&$

 !

%"% $%%!

%$#-!

%/%='

#

%$%(

.%#

3'!%%(

* #$//$%

(%#

.6'%#



 !

"!#$

!$%&'#!!

'%!(#)!$

\*!+#

,$%%#

-.(%!#

/0

$/123/!#

4$

5-%$62$7-!'#-

8

9://&;;</\*=$!$ +>=?@<<A<\*!+<<(A<BCDBE ?E+<';;

/;;<%<</\*=$!$ +F

-%;;<%<</\*=$!$ +>@=A<\*!+G

<BCDB ?HD;;'/#%$

'%/$'/ $%!

'#('$

%!#

$$/"-%(/

'/%!!

'

9:/%$

&#F

1!&$/%#-

%'&#I(&

/%#-&%%%'\*!

%%+#

5/1$%'438&$

 !

%"% $%%!

%$#-!

%/%='

#

%$%(

.%#

3'!%%(

* #$//$%

(%#

.6'%#

# Summing Up is (NP) Hard to Do

One can’t really be alive, work with computers, writing algorithms to solve hard problems, and be ignorant of the importance of a good algorithm. Most texts on scientific programming, which formed the bulk of what I read in my formative technical years, possess at least on cautionary tale about how slow a computation can be if the algorithm is bad.

One classic tale I like, which is often trotted out with gusto, is the calculation of the Fibonacci numbers. As a reminder, the nth Fibonacci number is the sum of the two previous ones

\[ F\_n = F\_{n-1} + F\_{n-2} \; . \]

So to compute $$F\_100$$, one need to compute $$F\_99$$ and $$F\_98$$, each of which need the two below them. Computing naively causes a lot of repetition since one computed $$F\_98$$ and $$F\_97$$ to compute $$F\_99$$ and $$F\_97$$ and $$F\_96$$ for $$F\_98$$ and so on. Of course, a fast algorithm exists which essentially involves recording these values once computed, thus ensuring that they are computed only once in the recursion.

While compelling, this cautionary tale is a bit of a cheat. Someone actually knows the best algorithm and it happens to be quite fast and, generally speaking, it isn’t all that interesting to compute the Fibonacci numbers oneself, since they are immutable.

Much more interesting are the situations that are dynamic. Finding the best route to visit an arbitrary list of cities is the stuff of legends with which the famous [Traveling Salesman Problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem) (TSP) wrestles. This problem is often listed as hard, or rather NP-Complete, a class of algorithms that belongs to the larger class of NP-Hard problems.

Roughly speaking, NP problems (or which NP-Complete and NP-Hard belong) are computational problems that do not possess known algorithms that can find solutions in polynomial time. What that means is that as the input data set, usually thought of as consisting of $$N$$ entries, is increased (i.e. $$N$$ grows) the computing time required to find a solution grows much fasters than any polynomial in $$N$$. Usually the growth is exponential, although worse cases, such as combinatorial growth, also occur. This is in sharp [contrast with the P class of computational problems](https://en.wikipedia.org/wiki/P_versus_NP_problem) that do possess known algorithms that can find solutions in polynomial time.

The curious thing about NP problems are that while they are not quickly solvable, they are quickly checkable. This phrasing means that while it takes a long time to find a solution for a large input data set, once the solution has been found it can be verified in a relative short period of time that scales as a polynomial in $$N$$.

This distinction is hard to understand and this difficulty is compounded by the fact that the NP class has three qualifiers making for NP, NP-Complete, and NP-Hard as well as containing the P class. In addition, problems like the TSP are very specialized and complex making them very hard to relate to in general

Note: The reason NP-Hard lies partially outside the NP boundary is because some NP-Hard may not even be decidable. Undecidable problems possess no definitive answer to the computing question with which they are posed. That such problems exist inescapably follows from Turing’s analysis of the halting problem.

So I searched for a simplified problem that I could sink my teeth into. A problem that would be tractable enough to play with but still demonstrated the ‘NP-ness’ in a way that I could relate. After some trial-and-error, it seems that Subset Sum Problem (SSP) fits the bill.

The SSP asks the following question. Given a sum $$S$$ and a set of integers $$Q$$ is there a proper subset of $$Q$$ that sums to $$S$$. It is provable that the SSP is NP-Complete: a solution exists (i.e. the problem is decidable) ; it is hard to find a solution, especially when $$Q$$ has a lot of elements; and it is easy to check the solution once the answer is revealed.

That the SSP is decidable is actually trivial to see. Given a subset $$P \in Q$$ of integers, it is easy to sum them together and to test whether the result equals $$S$$. By playing with a simple configuration for $$Q$$, it was easy to demonstrate that it is hard to just find a subset that sums to the desired value and that is equal easy to check once an answer is proffered.

Consider the following $$3 \times 4$$ block of integers between 1-9, inclusive.

The sum of all of them is 53 and the smallest number in the set is 1, so it is also easy to answer no to the SSP when $$S$$ is outside this range. But what about the rest of the values? Can any value between 1 and 53 be represented? (It should be clear that 1 is represented trivially by one of the 1s in Q and that 53 is represented by the whole set).

For a test,

# Do Web Searches Influence How We Think?

One of the central questions about linguistics asks how much thinking influences language and how much language, in turn, influences thinking. Clearly there is a link between how we express a thought and what the thought entails. The words we know, the familiar phrases we invoke certainly shape how others perceive our expressed thought; but what of ourselves? How much of a feedback loop exists wherein the manner in which we express a thought influences how that thought is perceived by the ourselves is unknown. Each of us plays the role of both speaker and listener when speaking and often how we express ourselves influences how we perceive ourselves – at least as we imagine how we are perceived by through others eyes.

Clearly there is a link but measuring that link and determining which direction is strongest (thought to language vs language to thought) is ephemeral. Natural language has been around for so long and with no formal metrics kept until what is essentially yesterday in terms of the march of history that it seems nearly impossible to find out much by examining everyday speech. Much like the fish that lives in the ocean, we are often even unable to see the medium of speech in which we swim since it is so pervasive.

Mathematical languages have offered better laboratories. Formal mathematical expression is relatively young – perhaps 500 years old – and is continually refined. By examining the evolution of mathematics, both the thoughts codified and the method used for the codification, a consensus has been reached that the form of the expression is often as important as what is being expressed. Why else are there what amounts to holy wars over mathematical notation. The camps of Newton and Liebniz often warred with each other over the best way to denote the calculus. Friction of this sort continued on into the 18th and 19th centuries with arguments over vectors and quaternions, sets and logic, and so on. The general observation is that that better notation means better thinking.

This sentiment is also very much alive and well in the equally contentious ‘discussions’ about which programming language is best. Adherent s on all sides will talk about the advantages and disadvantages that each has but, regardless of the particulars, one thing is clear; each programmer thinks his favorite language provides him the best avenue to express his thought, while simultaneously holding that other languages limit just how a programmer even thinks about how to solve a problem.

In a nutshell, all three disciplines – linguistics, mathematics, and computer programming – possess practitioners, who at one time or another, expressed the old proverb:

<If all you have is a hammer, everything looks like a nail. Old Proverb>

Unfortunately, all three of these disciplines pre-date modern big data metric collection and analysis. But there is a modern language-and-thinking loop that is amenable to quantitative analysis. Each and every day, millions of web-enabled searches are performed. Some are looking for news, some researching information, some surfing for gossip, some rummaging for products or services, but all are getting auto-completion tossed there way.

It would be interesting and revealing to see to what extent auto-completion influences how we think. I am sure each of us has started typing a search string into Amazon or Bing or Google only to find ourselves distracted by an auto-completion that was suggested by other searchers. Surely these entities have the data and probably are sitting on it for commercial purposes but whether they are looking at it from a scientific point-of-view is unknown – although the probability is against it.

Perhaps there is a sociology or psychology department out there that can craft a well-thought out experiment, complete with control and test groups, where they ask students to perform research on the internet in support of a course. The students are then given two identically-looking search engines. The test group would get an unmodified search engine and the test group on with a different approach to auto-complete; maybe with more suggestive or more distracting completions relevant to the stated research goal. Once the data were reduced and analyzed new insights into how to understand the way language and thought interact would be available. These insights may then shape how we think about thinking, and so on.

# Just a Game?

A couple of weeks ago, I explored the notion of computational complexity and how amazing amounts of complexity can show up in seemingly simple contexts. In the particular case examined, the Subset Sum problem, the familiarity of the simple activities of ‘looking’ and ‘adding numbers’ masks a surprising conclusion that an algorithm that scales as a polynomial in time is most likely impossible to find. The Subset Sum is said to be an NP-Complete problem.

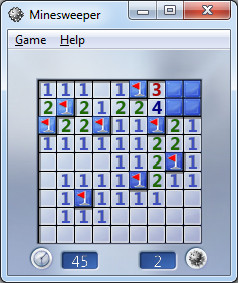
Computational complexity can be lurking in even tamer and more familiar guises. One of the more innocent-looking locations is in the simple computer game that comes with Windows – Minesweeper.

Before talking about the formal aspects of how Minesweeper is NP-Complete, let’s look at how the complexity arises from a practical example. For simplicity, I’ll be looking at three separate cases drawn from the beginner’s level of Minesweeper.

In all cases, one starts with zero information as to the placement of the mines and so the first move is always a blind-luck guess.



If it works out, one gets information about the number of mines in the eight nearest-neighbor spaces. If one is lucky, the information is enough to solve the puzzle outright. Case 1 is such a case. After one successful blind guess, the information available on the board was sufficient to get to the end stage where it is obvious how to solve the puzzle

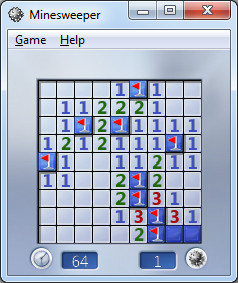


The two remaining mines need to go directly next to the ‘3’ and the ‘4’ and so the remaining two spaces can be explored without fear.

Sometimes, no amount of logic can save you and there is nothing to do but guess as in Case 2. After the initial blind-guess the board looked like



Solving for the obvious spaces leads to the final configuration with two unexplored spaces and one remaining mine. No amount of deductive or inductive logic will save the day and a 50/50 guess is required (for those who care, I guessed wrong).

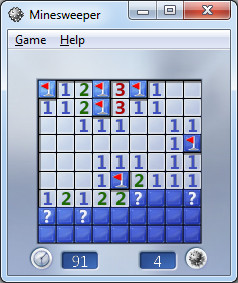


Both of these cases have the luxury of being straightforward. The amount of thinking was kept to a minimum and the options were well understood (although not always successfully so).

The true complexity of the game creeps becomes apparent in the in-between Case 3. After the bling guess and resolution of the easy spaces, the board looked like

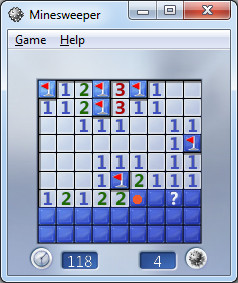


There is no way to easily finish off the board but nor is the final solution up to chance. It is a matter of exploring possibilities. Since ‘1’ spaces are easiest to deal with, let’s focus on the line of three ‘1’s found on the lower right. Let’s tentatively mark the rightmost ‘1’ as having a mine directly below it by placing a ‘?’ on the space. Making that assumption leads to a propagation of other possible mine spaces, each marked with a ‘?’, as shown below

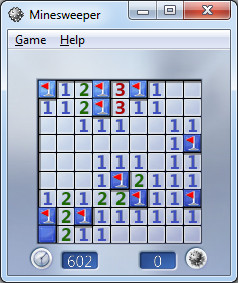


The entire set of ‘?’s is consistent – that is to say, we don’t have a contradiction anywhere.

Next suppose that we assume a mine just below the middle ‘1’ on the row of three on the right. Putting a ‘?’ there leads to an inconsistency, since if there is a mine at the location of the ‘?’ there is no way to satisfy the ‘1’ and ‘2’that are next to each other. The red dot (which is photo-edited in since Minesweeper doesn’t provide a marker for inconsistency) drives this point home.



Similar logic holds for putting the ‘?’ under the leftmost ‘1’ and so the only consistent guess was the first one. Acting on that guess leads immediately to success.



The true complexity of Minesweeper becomes obvious upon reflection of what can happen when the field is bigger and the number of mines is increased. Each blind luck move at the beginning of the game can lead to numerous possible avenues to explore. Each ‘?’ placed on the board can spawn additional channels to explore the difficulty can grow exponentially.

This argument is a plausible, heuristic one so there may be a tendency for the reader to respond ‘But you really haven’t proven it!’. That is a point I am willing to concede – I haven’t proven it – but someone has.

In the spring of 2000, mathematician Richard Kaye of Birmingham University was able to prove that [Minesweeper is NP-Complete](http://web.mat.bham.ac.uk/R.W.Kaye/minesw/ordmsw.htm). This means that Minesweeper is a decision problem (did you solve it or did you detonate), it is very hard to find a solution, and it is very easy to check the solution. He also went on to [construct various logic gates in Minesweeper](http://web.mat.bham.ac.uk/R.W.Kaye/minesw/ASE2003.pdf) and using these, proved, in 2007, that [Minesweeper played on an infinite board with infinite mines (although fewer mines than spaces) is also Turing complete](http://web.mat.bham.ac.uk/R.W.Kaye/minesw/infmsw.pdf).

So the next time someone asks chides you for wasting time playing Minesweeper, respectfully point out that you are wasting time, you are researching one of the most important open questions in computer science.

# Metaphors and Videogames

The genesis of this post was some time off that I finally took from work and the chance to sit down and play some videogames and relax. In the process of playing an old favorite and then discussing it with my friends I got a new appreciation for some of the subtle points of how language and thinking shape each other.

To set the stage, I must confess that many years back, I willingly allowed myself to be swept-up by the *Doom* craze. I recall that when the free sample came out a lot of guys at work (me included) loaded up the game as a diversion from the code-slinging the powers-that-be demanded. A few scant years later found me in possession of *Doom* and *Doom 2* and the corresponding Battle Books (walkthroughs with a much cooler name) and having spent countless hours dodging imps, cacodemons, and the like.



Fast forward nearly 20 years (sigh…) and similar scene is now playing out again, although in a different residence and on a quite different platform. Gone was the old DOS-based, Windows-running Pentium machine I spent over $2000 to buy. In its place is a sleek, black PS3 which cost about a tenth of that price with decidedly more graphics and computational horsepower. Gone are the old CD-ROM versions of the games, each requiring a lengthy installation. In its place is a DVD bearing the name *Doom: BFG Edition* (I wonder – what does BFG stand for?) containing *Doom’*s1-3; each capable of starting up right out of the box with no installation. Both changes are improvements to be sure. Unfortunately, not everything was an improvement.

After playing the original *Doom* on the PS3 for a short while, I began to become very irritated with the awkwardness I was experiencing. Certainly a large portion of that was initially due to my rust and age. After all, two decades had passed since I regularly played arcade and first-person shooter type games, and my hand-eye coordination was not as fresh and sharp. But my expectation was that the irritation would fade as I got more practiced and that expectation was not met.

Upon some reflection (once my adrenaline had dissipated), I realized that it was a particular game mechanic that was getting in the way. In the original *Doom*, which was released for the PC, the developer had a vast number of input keys/devices to choose from. The keyboard alone offers over 100 different keys, including the arrow keys. Throw in a mouse and one has a potent set of device combinations to map to the game mechanics.

The situation is quite different on the PS3. While it is true that the two analog sticks each offer full 2-degrees of freedom, they are almost always reserved for in-game translation and rotation. That leaves only 14 keys for interacting with the cyber-environment with the triangle, square, circle, and ‘x’ being the primary ones.

For much of Doom’s game play, which is primarily moving, turning, and shooting, the smaller number of key’s is not an impediment. The place where the mechanic breaks down is weapon selection. At the start of the game, your character is simply armed with… well his arms. As the game progresses, additional weapons are acquired, each with its own strengths and weaknesses. During battle with a host of foes, you need to be able to switch weapons smoothly to maximize you lethality and minimize your chances of dying. On the PC, weapons-switches happened by selecting a number key in the range 1-6, inclusive. Selecting ‘1’ meant you were using bare knuckles, ‘3’ was the shotgun, and ‘6’ meant you were yielding the BFG (there’s that abbreviation again – I wonder…). In the PS3 version, random access of the weapons list is replaced by an awkward cycling through of the weapons in one direction by repeatedly pressing the circle button.

It is really hard to select the correct weapon while being attacked and, even when it is calm, it is easy to accidently go past the weapon you want and have to circle back to get it. Speed and fluidity are hallmarks of *Doom* and this method of switching weapons really detracts from feeling immersed in the game.

This is a case where the language betrays the message; a case where how the idea is expressed is a road-block to the idea itself. It is an example of a user-interface metaphor that simply doesn’t work.

Often we don’t notice when the metaphor expressed in a user-interface is good. That is as it should be, with the language (the particular button sequence being used) cleanly expressing the intention (the actions on the screen). But we clearly notice when the metaphor is bad. It sticks out like a sore thumb.

As a user community we’ve even come to expect compromised or even bad metaphors from software we need to use. Examples here range from mandated applications at our work, or programs we use as a means to an end (e.g. tax preparation software).

But within the context of gaming, a bad metaphor can’t be ignored. After all, the purpose of the game is to have fun. If the metaphor is bad it lessens or even destroys the fun. Since the game is the end and not the means to another end, there is no way to justify awkwardness of expression. This conclusion holds broadly for much of how we use language. We will slog or way unhappily through an obscurely written tax form or government regulation simply because we must. But we will reject an unartfully constructed joke or an awkwardly written story. In the case of the videogame, we get to see the whole phenomenon work in a highly compact and specialized language – the language of buttons, joysticks, symbols, and lights and shadows played out on a screen.

# Statistics and Logical Ostriches

I forget exactly where I heard this, who said, or who it was about. All I remember about the pithy little saying I am about to express (rather than continuing to allude to it) is the laughs that it caused. I was driving along listening to some show about politics when one of the commentators suggested that some member of the government was ‘often wrong but never in doubt’.

Once my laughter faded – a process that took some time as that gem struck my funny bone – I began to reflect not only on the truth that was being conveyed but also why it worked that way. So many of our most prominent citizens, from all professions, seem to not understand just what logic was all about.

The examples are so numerous that I’ve begun to filter them out to the point where I can’t even cite a specific example. But the scenario is quite clear. A media figure cites a study that, perhaps, mildly suggests that item A and item B is correlated and runs with it to suggest that A causes B. A politician takes the words of another politician out of context, twists and turns them this way and that, and ends by constructing something with exactly the opposite meaning. This typically occurs in its most egregious form when there are big stakes on the line, the kind that engender large emotions, but it isn’t confined to that.

The effect is quite clear but what about the cause. Now if I am not careful, I’ll end up doing exactly what I am criticizing – drawing sweeping generalizations supported by a paucity of data but a lot of belief. So let me say simply that I think that what causes wrong conclusions walking hand-in-hand with little or no doubt is that it is uncomfortable to deal with the doubt.

Each time we try to make an inductive argument we are dealing, whether formally or not, with statistical data and probabilistic inferences. As Harry Gensler point out

<Much of our everyday reasoning deals with probabilities. We observe patterns and conclude that, based on these, such and such a belied to be probably true. This is inductive reasoning – Harry Gensler, Introduction to Logic>

Deductive reasoning is an all or nothing undertaking. Everything is locked down, certain, and the outcomes are never in doubt. The rules for proceeding from premises are established and have no room for negotiation. The application can be in error but sufficient effort always results in valid reasoning. The only open question is the soundness of the argument, which hinges on whether the premises are true.

In contrast with deductive arguments, inductive ones generally get murkier as data are added. Certainly a sufficient number of observations are needed to grasp a pattern but too many observations begin to undermine it. In addition, patterns are simple when they involve one or two variables but as the number of considerations grow so to do the number of combinations, each with its own rule. Inductive arguments always take the form of statistical syllogisms that works most cleanly the less we know. These arguments may not be reliable (the analog to sound) but they can be quite strong (the analog to valid) if the probabilities we assign to the premises are high. This is most easily done if we don’t have a host of additional factors making exceptions here and there.

Gensler provides a nice example of this point using college football and I want to give him full credit for a good teaching device. That said, I will present its flavor only after having switched to professional football.

Suppose we know, having analyzed the total plays called by the Pittsburgh Steelers, that 80% of the time they throw the ball deep on a second and short yardage situation when they are at least 30 yards from their own end zone. Suppose, also that we see during this Sunday’s divisional playoff game between them and the Denver Broncos, that the Steelers have 2nd-and-3 down with the ball on Denver’s 47-yard line. We can reason thus:

* The Steelers throw the ball deep 80% of the time with second and short yardage when they are not backed up their goal
* The Steelers have the ball on Denver’s 47-yard line with a 2nd-and-3 down
* There is an 80% chance the Steelers will throw deep

This seems simple enough, but Gensler often adds to this form the additional line that reads ‘This is all we know’. At first glance this may seem to be superfluous but as the next example shows, saying that we want to limit our facts may actually beneficial.

Suppose we know, again from the sample of plays, that the Steelers run 70% of the time when they are ahead by at least 10 points. Now if we knew nothing about down and distance but knew that they were winning 20-3, we might reason

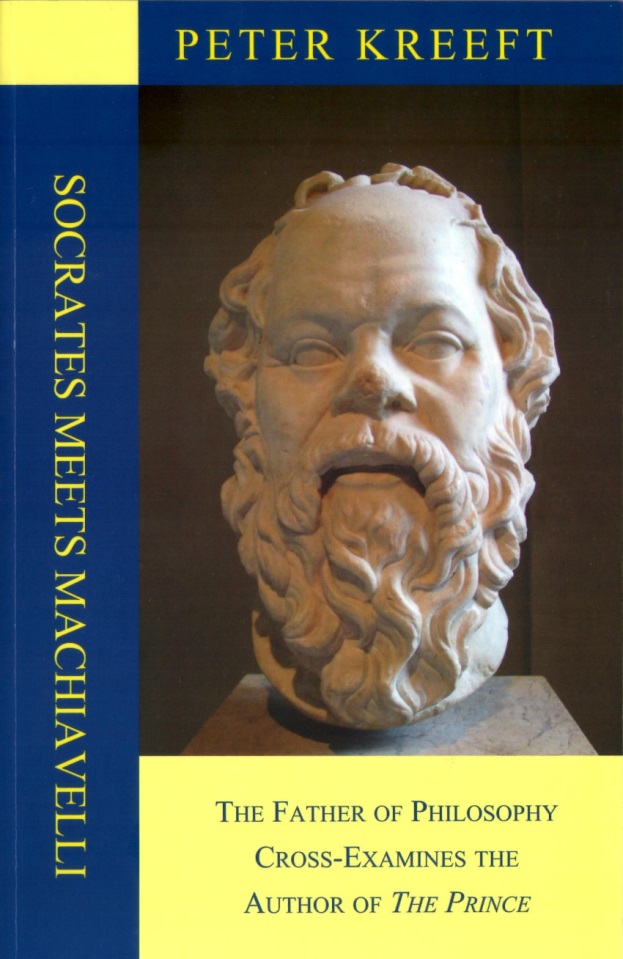
* The Steelers run the ball 70% of the time when they have at least a 10-point lead
* The Steelers are leading Denver by the score 20-3
* That is all we know
* There is a 70% chance the Steelers will run the ball

So far so good! But now consider that the score is 20-3 and that they have the ball on Denver’s 47-yard line in a short-yardage second down. What should we conclude? Do they run or pass. Perhaps they should just punt. Now layer in additional considerations like the time left in the game, weather conditions, and critical injuries and the situation really gets complicated.

And this is exactly why I think that there is such a correlation between certainty and intellectual short-sightedness. It is easier and more comforting to focus on a single aspect of a complex scenario, find a pattern and merciless apply it. The result is a logic ostrich, with its head in the sand, comforted that it can’t see the messy possibilities. Socrates was right – the only wisdom is in knowing that one is not wise (although it is possible that an ostrich is still better looking than Socrates).

# Socrates, Machiavelli, and Kreeft

I recently had a chance to read *Socrates Meets Machiavelli: The Father of Philosophy Cross-Examines the Author of the Prince*, by Peter Kreeft. Kreeft is a faculty member in the Department of a Philosophy, Morrissey College of Arts and Sciences, at Boston College. He is well-known as a Christian Apologist and, by his own words, is a teacher and not a great philosopher. Nonetheless, I’ve found many of his works to thought provoking and well-argued as any good philosopher should do. Even when I don’t agree with his conclusions I can see how he got from point A to point B why he would lean the way he did.



In *Socrates Meets Machiavelli*, Kreeft attempts to make a modern-day Socratic dialog out of an imaginary meeting between the two titular characters in some limbo after both have died. The work is intellectually stimulating but ultimately it fails at achieving the form and function of its more celebrated predecessors. Even though Kreeft’s writing is not as poetic or skilled as Plato’s, the reason for this dialog falling far short is that there is a sense that Kreeft has his thumb on the scale from the start when the two characters meet.

To understand this point, consider Plato’s classic work *The Euthyphro*, in which Socrates meets up with the title character outside the Athenian courthouse. Socrates is there to face the charges that would lead to his death. In contrast, Euthyphro is there to accuse not to rebut an accusation. Both are Athenian citizens meeting on common ground. Yes, Socrates is clearly the smarter of the two and he is far more artful in demonstrating the holes in Euthyphro’s arguments than the latter is in filling them. Despite that, Socrates doesn’t win in the dialog – he simply succeeds in pointing out that the truth is far more elusive than Euthyphro would like to believe.

From the start, the meeting between Socrates and Machiavelli is of a different cast. Consider the following exchange about 4 pages in

<

Machiavelli: I did die, didn’t I?

Socrates Correct again.

Machiavelli: So is this Heaven of Purgatory?

Socrates It is Purgatory for you and Heaven for me.

Machiavelli: How can that be?

Socrates It is for me a continuation of the most heavenly task I knew on earth: to inquire of the great sages, to pursue wisdom from those who know. For they are the opposite of myself, who do not know. And it will be Purgatory for you as it was to my fellow citizens of earth. But here no one has the power to give the gadfly a swat and send him away to the next world. You must endure my questions.

Machiavelli: So you *are* my torturer.

Socrates No, I am your friend.

Machiavelli: My inquisitor.

Socrates No, your teacher.

Machiavelli: By means of inquisition.

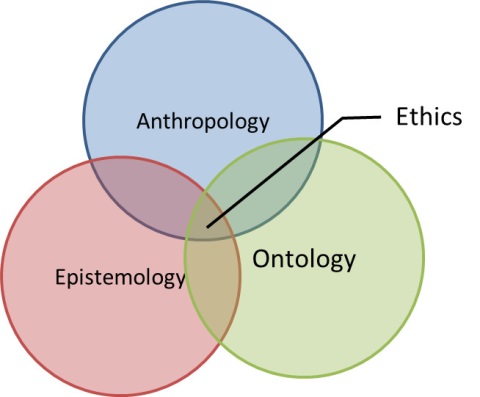
Socrates No, by means of inquiry. The unexamined life is not worth living, you know.

>

From this early stage Kreeft establishes the position of authority and power of Socrates over Machiavelli. As a result, the entire work feels more like a polemic and less like a pursuit of truth; more of a monolog by Socrates to teach Machiavelli the error of his ways rather than a dialog in which both separate error from fact.

Nonetheless, the book is well worth reading if for no other reason than the structured approach that Kreeft’s uses to analyze philosophy and ethics and the strong conclusions that follow from its use. In the interest of full disclosure, I am self-taught in my philosophical pursuits so, to the professional (which Kreeft, in his lecture course [*Ethics: A History of Moral Thought*](http://www.amazon.com/Modern-Scholar-Ethics-History-Thought/dp/B001GKVBVK/ref=sr_1_1?ie=UTF8&qid=1453040901&sr=8-1&keywords=History+of+Moral+Thought+Kreeft), points out is a bit like being an intellectual whore), maybe the following point is pedantic. Still I will pluck up some courage and put it forward.

Until digesting Kreeft, I had looked at the fundamental base of philosophy to be built from [ontology](https://en.wikipedia.org/wiki/Ontology) (the study of what the world is or the study of being) and [epistemology](https://en.wikipedia.org/wiki/Epistemology) (the study of what can be known). Kreeft makes a subtle but powerful division of ontology into two pieces: anthropology, here defined to be the study of the human being; and ontology, defined to be the study of the being of what remains. Ethics then finds itself at the intersection.



Of course, divisions of these sorts are artificial in that how we know what we know and why we know it is as much a function of who we are as people as it is a function of what is out there to be known. This division is not unlike the quantum mechanical division between system, environment, and observer. All are made from the physical world and are inextricably linked and yet it is useful to draw boundaries in order to understand.

And understanding is the great reward that comes from reading Socrates Meets Machiavelli. No matter where one stands on the ethics questions raised in the text or whether one agrees with Kreeft, there is no denying that he makes a powerful argument that the practical man is a philosophical man. That there is simply no way to succeed in the here-and-now without first deciding on firm meta-physical principles and then applying them is a central message; a message that makes one’s investment with this book worthwhile.

# Nintendo and Complexity

Several months ago, I wrote about the proof, due to Richard Kaye of Birmingham University in 2000, that the seemingly innocent-looking game [Minesweeper is an example of an NP-Complete problem](http://aristotle2digital.blogwyrm.com/?p=402). The essence being that no algorithm for solving the problem is known that scales as a polynomial function of the board size.

I suppose that it was inevitable that analysis of this sort would be extended to a host of other games. After all, most computer scientists no doubt enjoy gaming as much as they enjoy computers. In addition, unless there is some odd aspect of computer-scientist biology, each of them was once a child and, no doubt, was captivated by play. But it was with a particular satisfaction, that after aimlessly wandering across the internet, I discovered the charming paper entitled [*Classic Nintendo Games are (Computationally) Hard*](http://arxiv.org/abs/1203.1895).

This paper, written by Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta, is an exploration of five of the classic 8-bit Nintendo franchises: *Mario*, *Donkey Kong*, *Legend of Zelda*, *Metroid*, and *Pokémon*. Using an approach much like what Kaye used in his analysis, Aloupis *et al* examine the generalized form of each of these games. A generalized form means that the size and structure of an individual board is open to manipulation but that the basic rules of the game are not altered. Room take on arbitrary sizes and configurations and the number of non-playing characters (NPCs) can be unlimited. Other than these liberties, the underlying mechanics of the game is maintained.

Now I will confess that I never sunk much time into *Donkey Kong*, *Legend of Zelda*, and *Metroid* (much to my regret) and that I had only a passing flirtation with the Mario franchise but Pokémon is a different story. I’ve spent countless hours with that franchise and don’t regret a minute of it. So I thought I would discuss a little of how *Pokémon* can be thought of as NP-hard.

The first and most primitive notion is what Aloupis *et al* call the ‘decision problem of reachability’. This is a rather big and forbidding name for a basic problem that almost every gamer has asked himself during gameplay: ‘Given where I am and my current status can I reach that spot there?’ This is a particularly familiar problem in such large and open worlds like *Elder Scrolls V: Skyrim*.

The image below is an actual screenshot from Skyrim, where a player encounters a dragon on a mountaintop. Because this meeting is essential to the storyline, the game limits access to the summit until the player has completed certain changes in status (i.e. solve a number of puzzles or quests, etc.). Of course, the player doesn’t know this when the game commences and often the question as to whether the player can find a way up to the summit is left undecided until the plot advances to the point where the way becomes clear. The mountains that appear in distance remain undecided for many players throughout their interaction in the game. They don’t figure in the plot and their accessibility is unknown unless the player manages to reach the peak. In absence of such a feat, the player to conclude the answer as ‘maybe’ – thus showing how art imitates real life. I am sure that even the developers are not sure what answer to give.



Fortunately, the Pokémon games, being 2-dimensional worlds, are much simpler. But not so simple that the paper by Aloupis *et al* is a trivial read. In fact, I found it to be quite difficult in the sense that the framework being used familiar only to the specialist and I have neither the time nor the inclination to get completely up to speed.

Instead, I would like to convey the flavor of it in terms of a Pokémon player. The general idea about ‘decision problem of reachability’ is captured in the game mechanic familiar to anyone who has ever played Pokémon, the NPC enemy trainers.

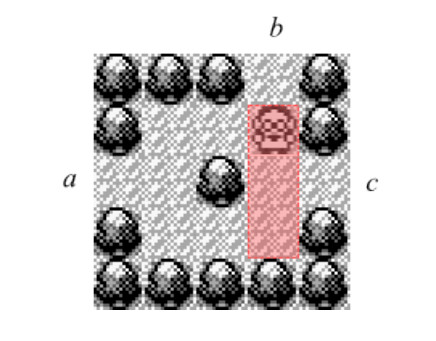
For those you are unfamiliar or who simply want me to define my terms carefully, here is a quick summary. In Pokémon, the player takes on the role of a young trainer, who has in his possession between 1 and 6 pocket monsters. His goal is to level up his Pokémon primarily through engaging in battles with other Pokémon either wild or in the possession of other trainers. As the player roams the world, he encounters other trainers who, if the conditions are right, challenge the player to a battle.

Each enemy trainer possesses a set location, a direction in which he faces, and a line-of-sight. There are two ways to trigger a battle with an enemy trainer. First the player can ‘sneak up’ on the enemy trainer by moving next to him without entering his line-of-sight. Taking to the enemy trainer initiates the battle but leaves the NPC in its set location. Second, the player can walk through the enemy trainer’s line-of-sight. The moment the player enters a space within the line-of-sight his motion is stopped and the enemy trainer moves from his set location to challenge the player. Alternatively, a player may choose to avoid the enemy trainer by either avoiding talking to the NPC or by not entering his line-of-sight. This latter option is not always available.

Depending on which choice the player makes, certain areas in the game become accessible or inaccessible as the NPCs move to open or block certain spaces. This behavior, when generalized, is at the heart of the proof by Aloupis *et al*.

The basic structure of their analysis is the creation of certain playable scenarios, called gadgets. There are 6 gadgets that they use to prove NP-hardness: *Start*, *Finish*, *Variable*, *Clause*, *Check*, and *Crossover*. The precise nature of these gadgets is involved, so I’ll only talk a little about the Variable gadget as it is easy to understand.

The Variable gadget’s purpose is to provide the player with a single choice that flips a switch between two positions. The construction Aloupis et al provide is



The red rectangle indicates the line-of-sight of the enemy trainer and the set location of the trainer is (2,4) (2nd row, 4th column from the upper left – denoted by the caricature of a man with glasses in a lab coat). A player entering at *a* has two choices. First, the player may choose to sneak up on the enemy by take the top fork and beating him in battle, thus leaving the exit at *c* open. Second, the player may choose to take the bottom fork, forcing the enemy to come down to (3,4) to battle. Winning the battle then leaves the exit at *b* open.

The other 5 gadgets are constructed with similar elements and mechanics but often they are larger and more complicated to understand. And the details are not all that important unless one wants to understand the technical details. Rather, the basic message is that even cloaked behind the guise of a game, even ones as fun as the five famous Nintendo franchises analyzed, logic and decidability are all around us.

# A Flip of the Coin

Ideological conflicts are often the bitterest of arguments that appear in the race of Man. Whether the religious wars of a post-reformation Europe, the polarizing arguments of modern US politics, the simple disputes over which is better PCs or Macs, or whether somebody should be a dog-person or a cat-person, these conflicts are always passionate and, while important, they are, in some aspect, pointless. Clearly adherents of both sides always have a good reason for supporting their position; if they didn’t they wouldn’t support it so vehemently. Those bystanders without a strong opinion one way or another are left to just shake their heads.

One such ideological conflict is the strong, often mean-spirited, argument between the Frequentists and the Bayesians over the right way to characterize or define probabilities. For much of this particular cultural hotspot, I’ve been a bystander. By training and early practice, an outsider would have characterized me as a Frequentist since I am comfortable with and enjoy using sampling techniques, like classical Monte Carlo, to investigate the evolution of a given probability distribution. Over the past 6 or 7 years, I’ve come to a better appreciation of Bayesian methods and find myself in the ever-growing position of seeing the complementary utility of both.

Nonetheless, finding a simple scenario that captures the essential aspects of these schools of thought and is easy to articulate has been elusive – that is until recently. I now believe I’ve found a reasonably compact and understandable way to demonstrate the complementary nature of each of these techniques through the flip of a coin. (Although I am quite sure that there is great room to improve it across the board).

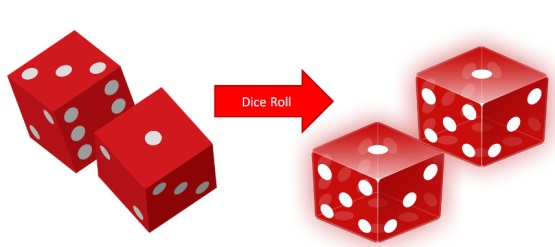
Before diving into the coin flip model, I would like to summarize the differences between the Frequentist and Bayesian camps. While the coin-flip model is strictly my own – based on my own thoughts and observations, the following summary is heavily influenced by the online book entitled [*Entropic Inference and the Foundations of Physics*](http://www.albany.edu/physics/ACaticha-EIFP-book.pdf), by Ariel Caticha.

The reader may ask why the two camps matter and why people just can’t agree on a single definition of probability. Why do the two camps even matter

On the notion of probability, Caticha has this to say

<The question of the meaning and interpretation of the concept of probability has long been controversial. Needless to say the interpretations offered by various schools are at least partially successful or else they would already have been discarded. But the different interpretations are not equivalent. They lead people to ask different questions and to pursue their research in different directions. Some questions may become essential and urgent under one interpretation while totally irrelevant under another. And perhaps even more important: under different interpretations equations can be used differently and this can lead to different predictions. – Ariel Caticha>

The Frequentist model of probability is based on what I call the gambling notion. Take a probability-based scenario, like throwing a pair of dice, and repeat for a huge number of trials. The probability of a random event occurring, say the rolling of snake eyes (two 1’s)



is empirically determined by how frequently it shows up compared to the total number of trials. The advantage that this method has is that, when applicable, it has a well-defined operational procedure that doesn’t depend on human judgement. It is seen as an objective way of assigning probabilities. It suffers in two regards. First, the notion of what random means is a poorly-defined philosophical concept and different definitions lead to different interpretations. Second, the method fails entirely in those circumstances where trials cannot be practically repeated. This failure manifests itself in two quite distinct ways. First, the question being asked could be entirely ill-suited to repeated trials. For example, Caticha cites the probability of there being life on Mars as one such question. Such a question drives allocation of resources in space programs around the world but is not subject to the creation and analysis of an ensemble of trials. Second, the scenario may naturally lend itself to experimental trials but the cost of such trials may be prohibitively expensive. In these cases, it is common to build a computational model which is subject to a classical Monte Carlo method in which initially assumed distributions are mapped to final distributions by the action of the trial. Both the mechanics of the trial and the assumed initial distribution are subjective and so are the results obtained.

The Bayesian approach relies heavily on Bayes theorem and, in doing so, allows the user to argue consistently about issues like the life-on-Mars question without needing to define random or determine how to implement trials. This approach eschews the notion of objective determination of probability and, instead, views probability as an epistemological question about knowledge and confidence. Thus two observers can look at the same scenario and quite naturally and honestly assign quite different probabilities to the outcome based on each’s judgement. The drawback of this method is that it is much harder to reach a consensus in a group setting.

Now to the coin-flip model that embodies both points-of-view. The basic notion is this. Suppose you and I meet in the street and decide to have lunch. We can’t agree on where to go and decide to pick the restaurant based on a coin flip done in the following fashion. I’ll flip a quarter and catch it against the back of my hand. If you call the outcome correctly we go to your choice; alternatively we go to mine. What is the probability that we end up in your restaurant versus mine?

Well there are actually two aspects of this problem. The first is the frequentist assignment of probability to the coin flip. Having observed lots of coin flips, we both can agree that the prior to the flip, the probability that it being a heads or tails is 50-50.



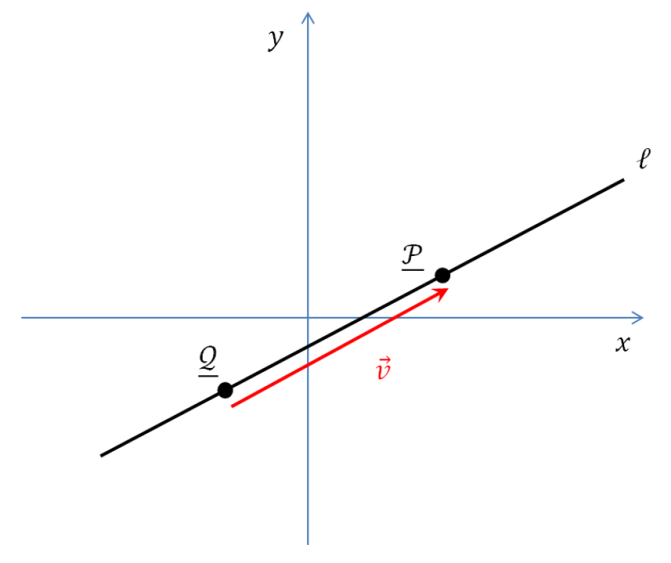
But after I’ve flipped the coin and caught it, the probability of it being heads is a meaningless question. It is either heads or tails – it is entirely deterministic. It is just that you don’t know what the outcome is. So the probability of you picking the right answer is not amenable to trials. Sure we could repeat this little ritual every time we go out to lunch, but it won’t be an identical trial. Your selection of heads versus tails will be informed based on all your previous attempts, how you are feeling that day, and so on.

So it seems that we need both concepts. And after all, we are human and can actually entertain multiple points-of-view on any given subject, provided we are able to get past our own ideology.

# Vectors as Instructions

For many of us, it’s hard to imagine how the standard blackboard construction of lines and points in the plane can hold any mystery or subtlety. What could be simpler than drawing a line in two dimensions and decorating it with some points, each bearing a label? However, many of the intuitive notions that we have about things aren’t sufficiently well-defined to survive the transition from casual use to rigorous proof. This is especially true in all things associated with getting a computer to perform a set of functions for us; the first step in building good software often being the sharpening of fuzzy ideas. The human capacity to ‘figure it out’ has yet to be designed into a set of instructions.

The issue that arises in constructing a line with points in the plane is in understanding the distinction between the ordered pair of numbers used to denote a given point and the ordered pair of numbers used to denote a vector. Take the line $$\ell$$ in the plane.



The points $${\mathcal P}$$ and $${\mathcal Q}$$ fall on the line and have coordinates given by the ordered pairs

\[ {\mathcal P} = \left( \begin{array}{c} x\_1 \\ y\_1 \end{array} \right) \]

and

\[ {\mathcal Q} = \left( \begin{array}{c} x\_0 \\ y\_0 \end{array} \right) \; .\]

Note that there is nothing special about writing the points in terms of column arrays, rather than the more usual $$(x\_i, y\_i)$$ notation. The reason for doing this is a matter of notation convenience that should become clear below.

Intuitively, we understand that if the vector $$\vec v$$ stretches from $${\mathcal Q}$$ to $${\mathcal P}$$ it components are given by

\[ [\vec v]\_x \equiv v\_x = (x\_1 – x\_0) \]

and

\[ [\vec v]\_y \equiv v\_y = (y\_1 – y\_0) \; ,\]

where the notation $$[\vec v]\_i$$ should be read as ‘get the ith component of the vector $$\vec v$$’.

At this point, there is a strong temptation to write the vector $$\vec v$$ in the same fashion

\[ \vec v = \left( \begin{array}{c} x\_1 – x\_0 \\ y\_1 – y\_0 \end{array} \right) \]

as we did the points $${\mathcal P}$$ and $${\mathcal Q}$$.

This approach certainly provides some benefits. That the notational forms for points and vectors look the same fits the visual picture of $$\vec v$$ connecting $${\mathcal Q}$$ to $${\mathcal P}$$. But the cons of such an approach outweigh the benefits. An individual point on the line is a zero-dimensional object whose address in space is given by the ordered pair. The vector is a one-dimensional object, a portion of the line that behaves like a directed line segment.

In addition, the vector is completely insensitive to the change in the origin. What to make of two new points $${\mathcal R}$$ and $${\mathcal S$$ specified by the ordered pairs

\[ {\mathcal R} = \left(\begin{array}{c} x\_2 \\ y\_2 \end{array} \right) = \left(\begin{array}{c} x\_0 + a \\ y\_0 +b \end{array} \right) \]

and

\[ {\mathcal S} = \left(\begin{array}{c} x\_3 \\ y\_3 \end{array} \right) = \left(\begin{array}{c} x\_1 + a \\ y\_1 +b \end{array} \right) \; ?\]

Depending on the choices for the values $$a$$ and $$b$$, $${\mathcal R}$$ and $${\mathcal S$$ may not even fall on $$\ell$$ and yet they have the same vector $$\vec v $$ connecting them as does $${\mathcal Q}$$ and $${\mathcal P}$$.

Mathematicians like to draw the distinction points and vectors but they are often clumsy about it. Take, for example *A course in mathematics for students of physics*, Vol. 1 by Bamberg and Sternberg. These authors identify the vector $$\vec v$$ as an equivalence class and they use the cluttered notation

\[ {\mathcal P} “+” \vec v = {\mathcal Q} \]

to define it in terms of the more primitive points. They also use different delimiters around the column arrays which specify the components: parentheses for one and square brackets for the other. Although it isn’t important which is used for which, note, for completeness, that the notation used in this column is opposite of Bamberg and Sternberg.

In this notation, the distinction between vector and points is front and center but at the cost of complication in the visual presentation. A parametric line would be specified as the set

\[ \ell(t) = \left\{ \left. \left(\begin{array}{c} x\_0 \\ y\_0 \end{array} \right) + t \left[ \begin{array}{c} v\_x \\ v\_y \end{array} \right] \right| t \in \mathbb{R} \right\} \; , \]

of all points related by the real number $$t$$, where the components of $$\vec v$$ are as specified above.

A cleaner way of thinking about these distinctions is to regard the relations more as computing instruction rather than as mathematical definitions. This allows a cleaner form of the notation and the defining equation

\[\vec v = {\mathcal P} - {\mathcal Q}\]

to be interpreted as ‘the vector $$\vec v$$ contains the instructions on how to move from the $${\mathcal Q}$$ to the point $${\mathcal P}$$. The equation of the parametric line, now cast in the abstract form without the column arrays, would be the set

\[ \ell(t) = \left\{ \left. {\mathcal Q} + t \vec v \right| t \in \mathbb{R} \right\} \; , \]

of all points formed by moving a variable amount $$t$$ from $${\mathcal Q}$$ according to the instructions held in $$\vec v$$.

The translation to objects in a computer program is now much more straightforward and natural than trying to parse what is meant by an equivalence class. To be clear, I am not criticizing the notion of an equivalence class nor its citation by Bamberg and Sternberg. Rather I am simply saying that viewing vectors in the context of directed line segments is much more natural in this computer-centric age.

# 

# Bare Bones Halting

It’s a rare thing to find a clear and simple explanation for something complicated – something like the Pythagorean Theorem or calculus or whatnot. It’s even rarer to find a clear and simple explanation for something that challenges the very pillars of logic – something like Gödel’s Theorem or the halting theorem. But J. Glenn Brookshear treats readers of his text *Computer Science: An Overview* to a remarkably clear exploration of the halting theorem in just a scant 3 pages in Chapter 12. I found this explanation so easy to digest that I thought I would devote a post to it, expanding on it in certain spots and putting my own spin on it.

Now to manage some expectations: it is important to note that this proof is not rigorous; many technical details needed to precisely define terms are swept under the rug; but neither is it incorrect. It is logically sound if a bit on the heuristic side.

The halting theorem is important in computer science and logic because it proves that it is impossible to construct an algorithm that is able to analyze an arbitrary computer program and all its possible inputs and determine if the program will run to a successful completion (i.e. halt) or will run forever, being unable to reach a solution. The implications of this theorem are profound since its proof demonstrates that some functions are not computable and I’ve discussed some of these notions in [previous posts](http://aristotle2digital.blogwyrm.com/?s=Halting&searchsubmit.x=0&searchsubmit.y=0). This will be the first time I’ve set down a general notion of its proof.

The basic elements of the proof are a universal programing language, which Brookshear calls Bare Bones, and the basic idea of self-referential analysis.

The Bare Bones language is a sort of universal pseudocode to express those computable functions that run on a [Turing Machine](http://aristotle2digital.blogwyrm.com/?p=199) (called Turing-computable functions). The language, which is very simple, has three assignment commands:

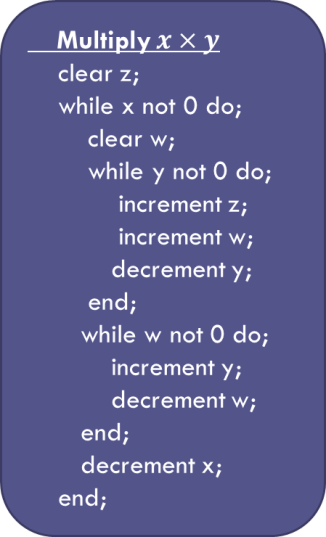
* Clear a variable
* Increment a variable
* Decrement a variable

In addition, there is one looping construct based on the while loop that looks like

while x not 0 do;

end;

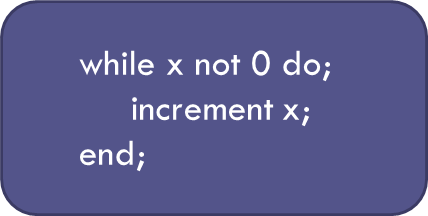
While primitive, these commands make it possible to produce any type of higher level function. For example, Brookshear provides an algorithm for computing the product of two positive integers:



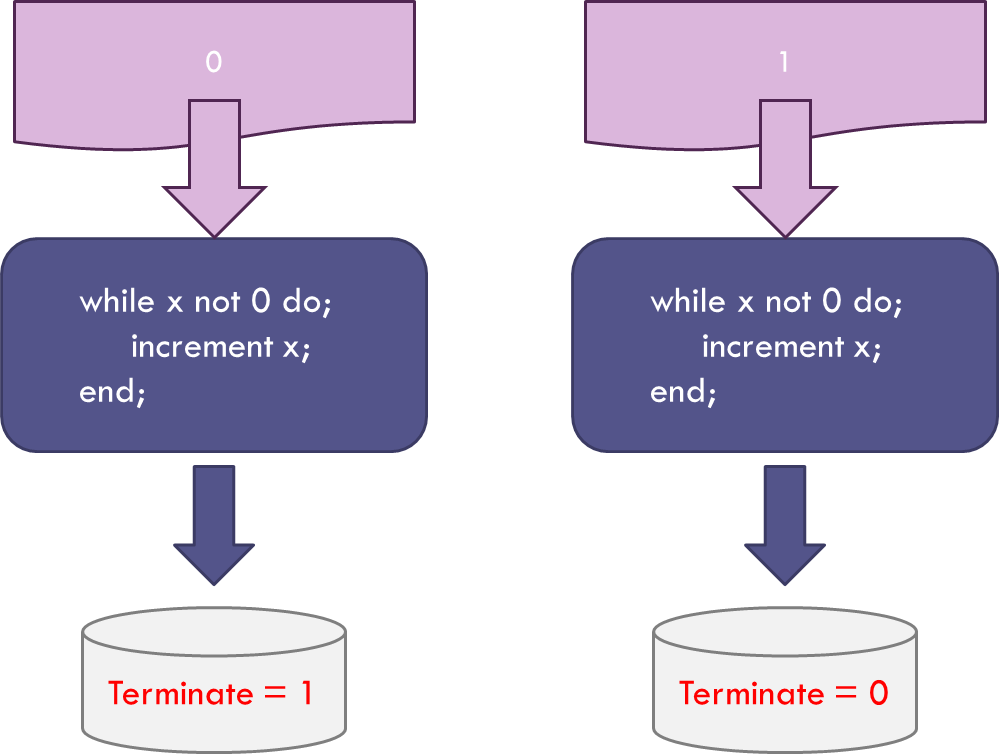
*Note:* for simplicity only positive integers will be discussed – but the results generalize to any type of computer word (float, sign int, etc.).

Now Bare Bones is a universal language in the sense that researchers have demonstrated that all Turing-computable functions can be expressed as Bare Bones algorithms similar to the Multiply function shown above. Assuming the [Church-Turing thesis](https://en.wikipedia.org/wiki/Church%E2%80%93Turing_thesis), which states that the set of all computable functions is the same as the set of all Turing-computable functions, allows one to conclude that Bare Bones is sufficiently rich to encode all algorithms that express computable functions.

One particularly important Bare Bones algorithm is something I call x-test (since Brookshear doesn’t give it a name).



The job of x-test is simple: it either terminates when x is identically zero or it runs forever otherwise.



The Terminate flag is assigned a value of 0 if x-test fails to halt and 1 if it does.

The next step brings in the concept of self-reference. The function x-test takes an arbitrary integer in and gives the binary output or 0 or 1. Since x-test is expressed as a set of characters in Bare Bones and since each character can be associated with an integer, the entire expression of x-test can be associated with an integer. To be concrete, the string representation of x-test is:

‘while x not 0 do;\n increment x;\nend;’

A simple python program, called string\_to\_int

def string\_to\_int(s):

ord3 = lambda x : '%.3d' % ord(x)

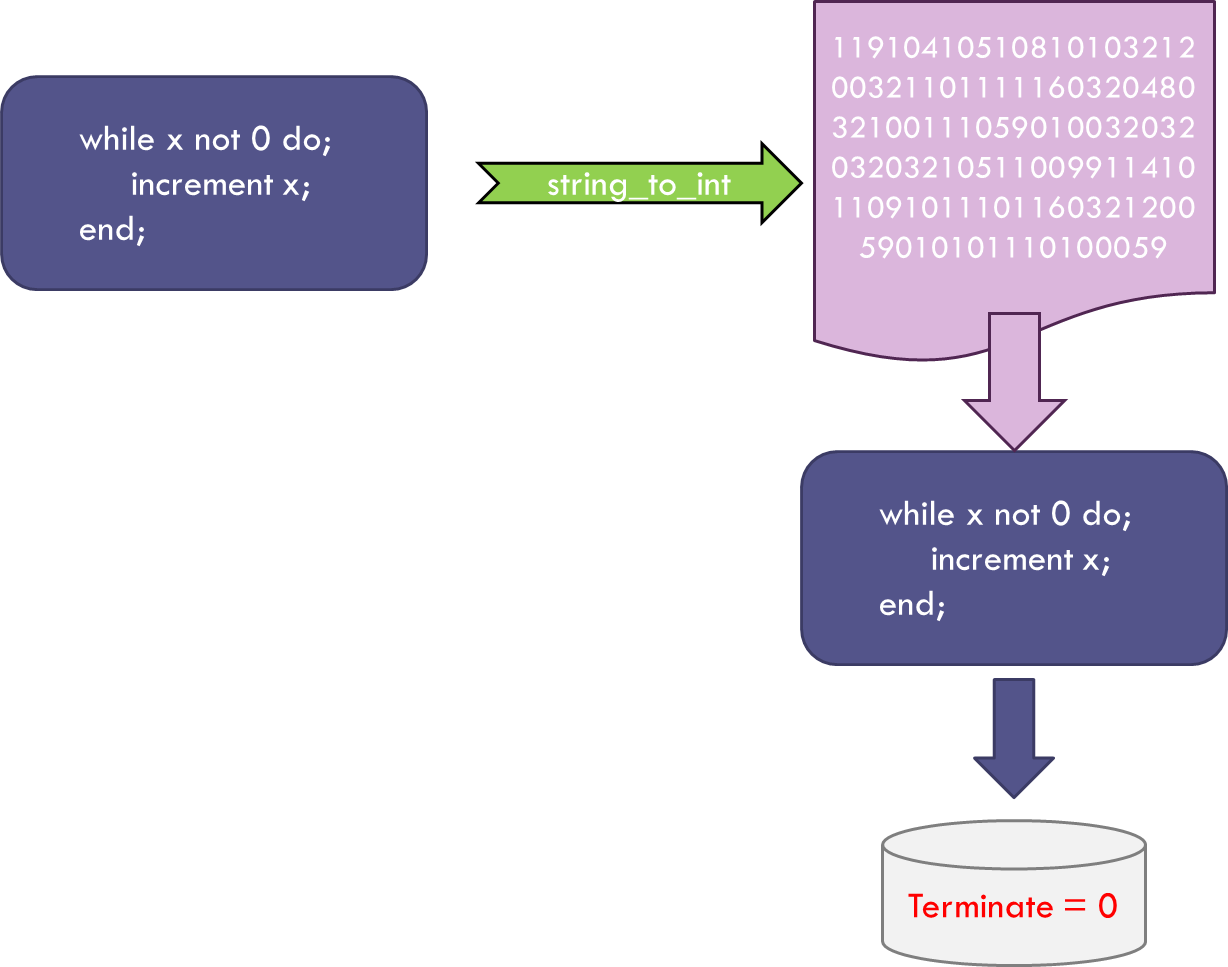
return int(''.join(map(ord3, s)))

transforms this string to the (very long) integer

11910410510810103212003211011111603204803210011105901003203203203210511009911410110910111011603212005901010111010005

Using this integer in x-test results in the Terminate flag equal 0.

Of course, any expression of x-test in any programming language will result in a nonzero integer and so x-test never has a chance to terminate.

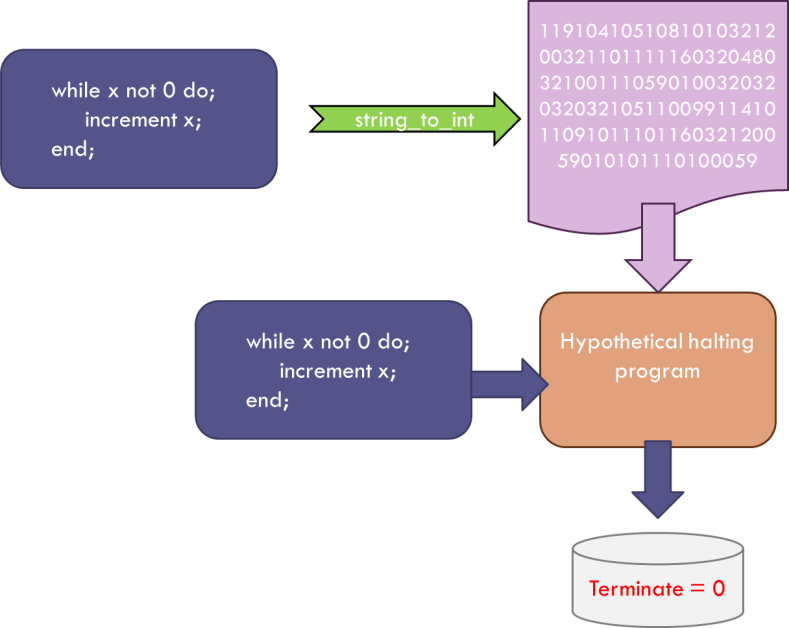


But there are clearly functions that do terminate when they are expressed as input for their own execution (one obvious example is the duel function to x-test where the while loop stops if x is non-zero). Such functions are said to be self-terminating.

Although it may look like a lot of effort for nothing, this notion of self-terminating functions is the key to the halting theorem and all that remains is to combine the above results with some clever thinking.

The clever thinking starts with the assumption that there exists a function, which I call the Hypothetical Halting Program (HHP), that can analyze any arbitrary program written in Bare Bones along with any corresponding input and determine which combinations will terminate (decidable) and which will not (undecidable).

Again, to be concrete, let’s look at what the HHP would do when looking to see if x-test is self-terminating.

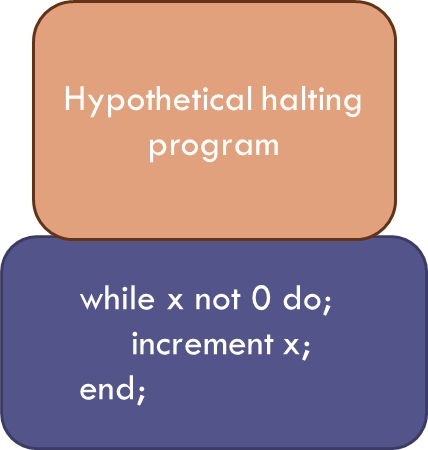


The HHP would be given the x-test function and the corresponding input (x-test in integer form) and would determine, just as we did above, that x-test won’t terminate – that is, x-test is not self-terminating.

Now such a function that works only on x-test is easy to understand; the text above that argued against x-test being self-terminating is one such instantiation. However, the goal is to have the HHP be abale to analyze all such algorithms so that we don’t have to think. Already, warning bells should be going off, since the word ‘all’ is a pretty big set, in fact infinite, and infinite sets have a habit of behaving badly.

But let’s set aside that concern for the moment and just continue our assumption that HHP can do the job. As we progress, we’ll see that our assumption of the existence of the HHP will lead to a contradiction that rules out this assumption and, thus, gives the desired proof. Operating without this foresight, we argue that since the HHP is computable it must be able to find expression as a Bare Bones program. And since the HHP analyzes Bare Bones programs, it must be able to analyze itself and determine if it is self-terminating – spitting out a 1 if so and a 0 otherwise.

Now we can construct a new program, called the HHP++, that is the sequential running of the HHP followed by x-test.



We can always do this in Bare Bones by first running HHP and then assigning its output (0 or 1) to the input of x-test. And here is where it gets interesting.

Take the HHP++ and let it test itself to see if it is self-terminating. If it is self-terminating, the HHP piece will confirm it so and will output a value of 1. This value is then passed to the x-test piece which then falls to halt. If the HHP++ is not self-terminating, the HHP piece will output a 0 which, when passed to the x-test piece, halts the operation. Thus the HHP++ is a program that is neither self-terminating nor not self-terminating and so the only way that can be true is that our original assumption that the HHP exists is false.