On Information Systems and Patent Eligibility

David B. Conrad

# Introduction

Much uncertainty exists today regarding whether, and to what extent, software-related inventions are eligible for patent protection in the United States. This unease among practitioners about where statutory boundaries lie finds roots in the conflict between the nature of computer software and the Patent Act’s permissive recitation of eligible subject matter.

The Supreme Court has been unable to articulate exactly what it meant when it held forty years ago in *Gottschalk v Benson* that procedure for solving a mathematical problem, in the form of a computer program, may not be patented. Such an “algorithm,” the Court wrote, amounted to patenting the mathematical principle itself. The problem with this language in *Benson* is that *all* computer programs are by definition a procedure for solving one or more mathematical problems. All computer programs are algorithms, or a collection of algorithms. They are are simply a series of instructions for a computer system to follow—a series of steps. Why then do computer programs not expressly fall under Section 101’s identification of a “process” or “method”?

Additionally, computer programs are provided to a computer as software in tangible form, whether that form is as fixed as on optical disc media, or as ephemeral as a time-series of electrical impulses. The very nature of the computer hardware changes upon executing the instructions of program. It is now a “new” computer, because the tangible hardware components now function in an entirely different way. Shouldn’t such a programmed computer be a “new and useful … machine” for which a patent can be obtained?

These arguments have been made for decades to support the patentability of computer software. Yet the Supreme Court, with one exception, has rejected these arguments and held that claims sought for computer-related inventions are ineligible subject matter. The Supreme Court’s reasoning in these opinions do not provide an adequate as a framework to evaluate patent eligibility. The Court has used language describing exceptions to patent eligibility such as “[p]henomena of nature, though just discovered, mental processes, and abstract intellectual concepts.” This is not a clear articulation of the principles at issue. All inventions, as the Court itself has warned, can be reduced to an abstract idea.

This paper proposes a different framework in which to evaluate computer-related inventions. It begins by taking a step back to look at computers in their most general sense—as an *information system*, meaning it manipulates data to produce more useful information. Information systems, whether implemented by a digital computer executing a software program, by a human using pen and paper to solve a mathematical formula, or by an organized group of people in communication, are the “basic tools of scientific and technological work” that the Court has consistently sought to preserve as free for mankind to use. Thus, as an initial matter, information systems are not eligible for patent protection. When the patentee claims to have invented an information system or method of using one, the patentee has only claimed a bare idea, no different than claiming a notion of humans thinking a certain way and talking to each other about their thoughts. Exceptions to this principle that may warrant patent protection arise only when the applicant has solved a specific technological problem encountered while implementing the information system, or has applied the information system as a tool in a “new and useful way” to transform a physical substance.

This framework, in fact, reveals that the Supreme Court has taken an almost entirely consistent approach to patent eligibility over the past 150 years, since long before the arrival of digital computers. The holdings are not *ad hoc* or arbitrary. We can now abandon vague notions such as “mathematical principles” or “abstract ideas” as guideposts, and understand when to disregard recitations of technology as “insignificant post-solution activity.” Finally, it is safe for us affirm that a novel computer program or algorithm may, under appropriate circumstances, be declared patentable.

Below, this text discusses first how to define an “information system” and the role that a general-purpose computer plays. Next, the paper reviews the principles of patentability declared by the Supreme Court. Then we examine how the Supreme Court has responded to the appearance of computer-based information systems. Finally, the author proposes a framework for evaluating the patentability of information systems, whether computer-based or otherwise.

# Information Systems and the Information Age

## Information

We frequently hear that civilization is presently witnessing the “Information Age.” For the past several decades, the advancement of technology has been driven by a new paradigm of understanding that surpasses the mere tangible world. People now know that *information*, surprisingly, is not just a philosophical notion, but something can actually be quantified, measured, analyzed, and transformed in a well-defined mathematical sense.

Claude Shannon is widely credited with launching the scientific field of Information Theory through his publication of a “A Mathematical Theory of Communication.” Shannon explored the fundamental mathematics at issue when dealing with information and the communication of information in order. Some readers may be surprised to learn that, according to Shannon’s theory, the first paragraph of this paper contains only 1740 bits of information, even though its computer representation takes of 2784 bits of storage space.

These mathematical understandings about what information is and how to measure it have led to an information revolution, resulting in advancements in digital computers, communication, data compression, quantum computing, philosophy[[1]](#footnote-24). There is even an aspect of theoretical physics that posits that all the information that makes up the three-dimensional world as we experience is actually encoded as a two-dimensional structure—just like a hologram appears to have depth, yet is physically flat.

But for our purposes, the recent investigation into the nature of information itself, rather than its application to hard sciences and engineering, is important. The elevation of information as something to be understood allows us to look introspectively into how the processing of information is applied as a “basic tool” among humans. One area studying information looks at how information is used systematically in organizations. Unsurprisingly, this field of theory is called “information systems.”

Before we describe what an information system is we must first address the ambiguity of the word *information*. Because information has several semantic meanings, researchers of information systems have generally adopted two other words—data and knowledge—that help distinguish among these different meanings.[[2]](#footnote-25) Briefly, *data* refers to the syntax, *information* is its interpretation, and *knowledge* is how people use the information.[[3]](#footnote-26). Stated another way, data is input that is processed and output into meaningful information to convey knowledge as understanding, experience, accumulated learning, and expertise.[[4]](#footnote-27) For example, “January 29, 1966” may be a sequence of letters and numbers (data), processed as a person’s birthday (information), and can inform us when to throw a big party because the person has turned fifty years old (knowledge).

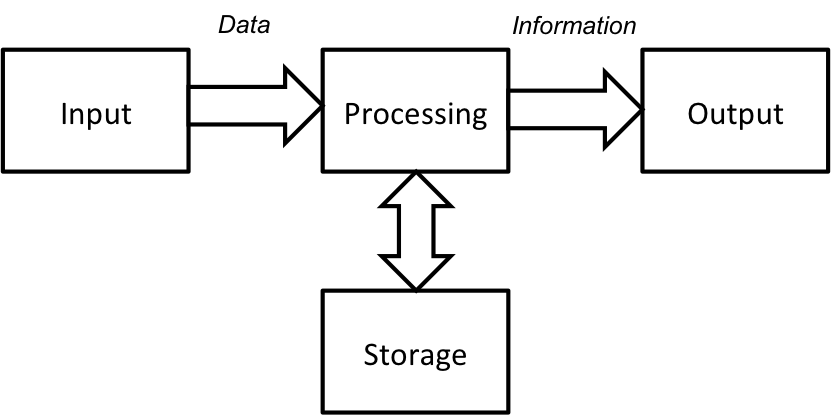
An *information system* is a system that manages and processes data and information to create knowledge.[[5]](#footnote-28) We use system here in its broadest sense as a set of connected things or parts forming a whole, not in any computer-based or technical sense. One commentator provides a simple example to understand what an information system is as a general concept:

A patient who consults a family doctor usually first tells the doctor about the symptoms. With this information, the doctor examines the patient and makes a diagnosis. Afterward, the doctor determines the treatment to heal the patient. For example, based on the diagnosis, the doctor may write the patient a prescription for some medication. Finally, the doctor must document the symptoms, the diagnosis, and the treatments. [[6]](#footnote-29)

In this example, the doctor receives data from a patient as a description of symptoms; the doctor processes the data based upon pre-existing medical knowledge to derive useful information in the form of a diagnosis. The doctor now has the knowledge to determine what treatments to use on the patient. The doctor manages this information by storing it with the data and associated knowledge.

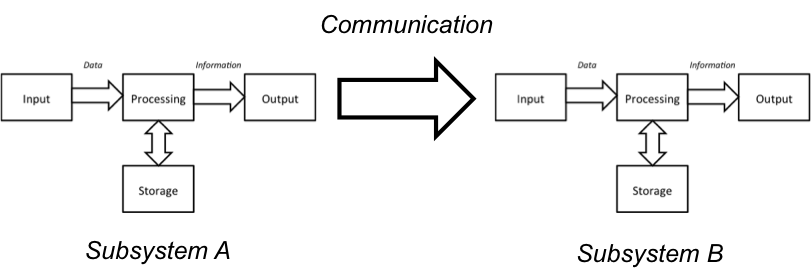
No computers are present in this example, even though they could have been used. Instead, we are presented with merely an organization of humans—a patient and a doctor—following a defined process: communicating symptoms and processing those symptoms to determine a treatment. In fact, understanding that the concept of an information system is not bound by any computer technology is critically important for our purposes.

The basics of an information system are four-fold, as seen in Fig. \_\_\_ below. Data is received by an input means. Processing means manipulate the data to produce information. An output means presents or displays the information in useful form. Storage means help retain the information, which may be only temporary as during the processing, or may be for much longer, until recalled at a later time.



Input-Processing=Storage-Output

Oftentimes times a fifth aspect is involved: communication. And that is because an information system can be described as a larger system comprising of various *information subsystems*. In order to interoperate, these systems must talk to each other, exchanging the data and information that is being processed, as seen in Figure \_\_\_ below.



Network-Subsystems

The reader can see that an information system is actually a very simple model of how a human being interacts with the environment. [[7]](#footnote-32) The senses (e.g., sight and hearing) are the input means. The processing and storage means are the brain, which seeks to comprehend the data collected by the senses and remember them and their meaning. The output means are the muscles controlled by motor neurons (e.g., voice and limb movement). Finally, humans interact with each other in organizations by communication, whether oral, written, through non-verbal cues, or by computers. A collection of people as a whole, such as a news organization, can input data of current events, work together to create meaningful information from the chaos, and output useful information in print, television or the Internet. Returning to our medical diagnosis example, the patient and doctor are each an information subsystem comprising an overall system where both actors are in communication.[[8]](#footnote-33)

One final consideration that is important when discussing information systems arises because of their need to communicate with their surrounding environment. A logical boundary exists between the information system and its environment, or between two subsystems, called an *interface*. The interface describes the characteristics of the means of inputing, outputing, and communicating. For example, when humans and computers interact, we often refer to this boundary as a human-computer interface. Various forms of human-computer interfaces have been employed over the years, such as hardware input devices like keyboards, voice recognition, and touch screens, or software such as graphical windows desktops or Internet search engines. Interfaces need not involve computers. A standard paper form for an inter-office memo is an example of a human-human interface in a corporation. Nor must interfaces involve humans. Computers, for example, sense the physical world using hardware instruments such as atmospheric sensors or gyroscopes. And computer software programs communicate with each other using a well-defined set of instructions called an application programming interface, referred to commonly as an API.

A variety of activities have been conventionally performed in information systems in order to accomplish the goal of creating meaningful information from data.[[9]](#footnote-34) The table below identifies several, although this list is certainly not exclusive.[[10]](#footnote-35) [[11]](#footnote-36)

|  |  |
| --- | --- |
| Activity | Description |
| Recording/originating | Collecting, creating, or updating data |
| Classifying | Arranging data according to characteristics and properties |
| Arranging | Organizing data into a logical structure, such as a table |
| Sorting | Ordering data in some sequential arrangement |
| Calculating | Applying mathematical formulas or operations on data |
| Summarizing | Reducing voluminous data to a more convienient and concise form |
| Storing | Persisting data (either temporarily or for long-term preservation) for future recall |
| Retrieving | Recalling stored data |
| Reproducing | Copying data (either temporarily for for long-term usage) |
| Communicating | Sending or distributing information between information subsystems, whether to a person or a machine |

As can be seen by these functions, information systems are not limited to a particular implementation environment. Many of these activities are the kind people have long been involved in, before digital computers, and even since the beginnings of recorded civilization. Ancient Bablyonians organized data into tables, just as today we create tables in spreadsheets on computer systems.[[12]](#footnote-37) More than two millenia ago, librarians at the Great Library of Alexandria stored and organized the books and records in rooms and bins by subject and author, and created a catalogue list of the holdings in a separate collection of scrolls.[[13]](#footnote-38)

Even though humans and human organizations can be the sole parties involved in processing information, in the modern world researchers have been more concerned with the application of technology to these methods.[[14]](#footnote-39) Many tools and technologies have been created over the years to replace or assist people, such as notebooks, file cabinets, mechanical adding machines, typewriters, slide rulers, scientific formulas, and telephones.[[15]](#footnote-40)

The most important addition to the mix has been the general-purpose digital computer. The computer has several main advantages over humans and mechanical tools. First, it is *faster* and more accurate.[[16]](#footnote-41) Second, it can make immediate decisions and *automatically* carry them out.[[17]](#footnote-42) Most importantly, as explained in the next section, a general-purpose digital computer is a *universal* calculating machine. As a universal machine, the computer can be programmed to perform *any* calculation that can be imagined. The computer just needs to be given the appropriate instructions and provided sufficient resources to carry it out.

## A Brief History of the Computer as an Information System

The computer was not always digital. In fact, the word “computer” originally referred to, simply, a person that calculates.[[18]](#footnote-44) But mental calculations do not come easily to the human mind,[[19]](#footnote-45) and from a very early time people sought tools to help them count and compute.[[20]](#footnote-46) The abacus is one example that is commonly known, which was used by the Sumerians and Egyptians to count and tally numbers since before 2000 B.C.E.[[21]](#footnote-47) Millenia later, in the sixteenth century, Galileo Gallilei and others developed the geometrical sector, or compass, which was two conjoined rulers inscribed with scales that helped solve various mathematical problems. A descendant of the sector was the slide rule, developed around 1642, that found use as recently as the 1960s by NASA engineers, pilots, and astronauts working on the Apollo moon missions.[[22]](#footnote-48) Charles Babbage created the difference engine in the 1850s, that used wheels and gears to compute polynomial functions.[^CITE] A very curious mechanical calculator called the water integrator, which took up space the size of a room, was developed around 1928 in the Soviet Union.[[23]](#footnote-49) Water residing in compartments represented stored numbers involved in calculations, and the water was manipulated such that its rate of flow through various piping represented certain mathematical operations. The water integrator reportedly could model and help solve complex mathematical principles such as calculus and differential equations.

These and other mechanical devices or automata, although incredibly useful, were limited to solving calculations for a particular problem or a narrow set of problems. They were not general-purpose in the sense that they could not be readily modified by users, changing their operation through a different sequence of instructions, in order solve a different kind of calculation. Charles Babbage proposed such a general-purpose mechanical system that would be programmed by punch cards, called the Analytical Engine, but was not able to complete it.[^CITE][[24]](#footnote-50).

The general-purpose computer did not arrive until 1936, when British mathematician Alan Turing published the mathematical foundation for the *universal computing machine*.[[25]](#footnote-51) Turing described a hypothetical machine that would read and print a sequence of symbols on a finite paper tape according to a set of rules. Turing proposed the idea of “a single [such] machine that can be used to compute any computable sequence.” Stated more generally, Turing’s thesis was that his universal computing machine could mimic any other computing machine if supplied with a description of its operation.

The universal nature of the machine Turing described, while a seemingly simple concept, has profound implications. If a machine could be reduced to a description in a series of symbols, it is no longer a *specific* machine. It is a *generic* machine because it can mimic the performance of other computing machines, or its description can be be applied to another computer to be imitated there. [[26]](#footnote-52) The point, here, is that the user need only think up the bare idea, and it is mathematical truth that one can write down a procedure in way the computer can understand in order to make the computer perform that idea.[[27]](#footnote-53) In short, if you could conceive it, you could build a computer that does it.[[28]](#footnote-54)

Essentially all mainstream computer environments are universal in this sense. Every modern computer can mimic every other computer.[[29]](#footnote-55) A tangible demonstration of this concept can be seen when desktop computers “emulate” other computers, such as when a desktop computer emulates an Atari 2600 console to run decades-old video games,[[30]](#footnote-56) a computer created by Apple runs software designed to operate only on Microsoft Windows computers,[[31]](#footnote-57) or when an Apple computer emulates a complete Microsoft Windows operating system environment.[[32]](#footnote-58)

This universality is what makes computers unique from all other tools that have preceeded their arrival, all of which are machines that solve only a very specific problem. In principle, nothing separates an individual from writing down an idea in a series of steps–an algorithm–in language he or she understands–like English–and translating that procedure into a program in a language that can be understood and run by a modern computer. One of the early innovators of computing systems, John von Neumann described, what happens after this translation takes place:

An automatic computing system is a (usually highly composite) device, which can carry out instructions to perform calculations of a considerable order of complexity. The instructions which govern this operation must be given to the device in absolutely exhaustive detail….These instructions must be given in some form which the device can sense: Punched into a system of punchcards or on teletype tape, magnetically impressed on steel tape or wire, photographically impressed on motion picture film, wired into one or more fixed or exchangeable plugboards-this list being by no means necessarily complete. All these procedures require the use of some code to express the logical and the algebraical definition of the problem under consideration, as well as the necessary numerical material. Once these instructions are given to the device, it must be able to carry them out completely and without any need for further intelligent human intervention. At the end of the required operations the device must record the results again in one of the forms referred to above. The results are numerical data; they are a specified part of the numerical material produced by the device in the process of carrying out the instructions referred to above.[[33]](#footnote-59)

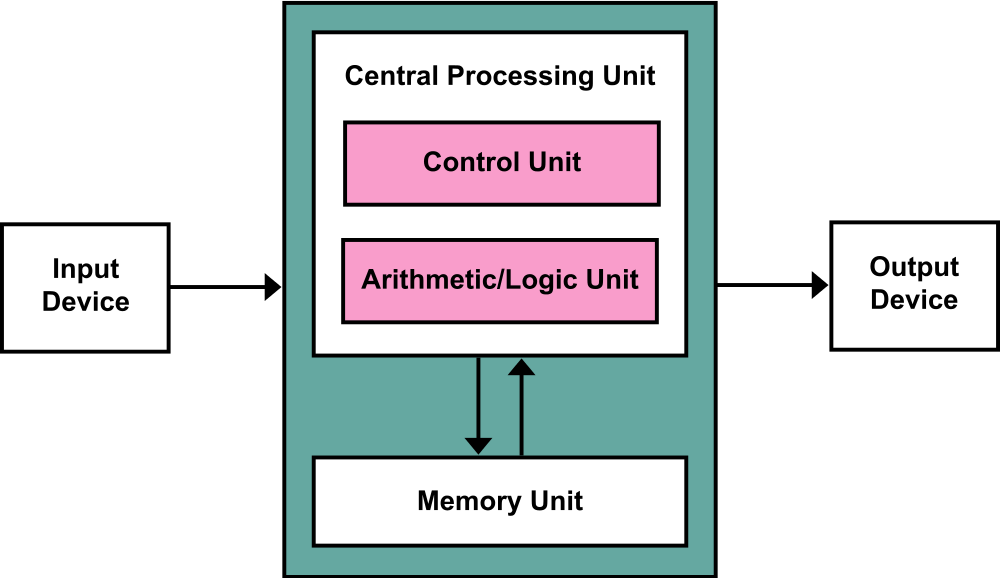
There are limits to this principle however, in that Turing’s universal computer is a theoretical ideal. Actual computers, however, are physical devices bound by limitations forced upon it by the physical world. Computers take time to compute and have limited room to store data and information. Thus, while a given computer could, in theory, perform any procedure intended to solve a problem, in practice it may not be able to effectively do so because its operation may not complete for millions of years.[[34]](#footnote-60) Since the introduction of the universal computer, much innovation has been focused on improving the architecture of the general purpose computer to make it faster or more efficient.

The first big step taken to advance a tangible architecture for the universal computer was taken by John von Neumann around 1945, when he developed the concept of a “stored-program computer.” von Neumann was the leader of the computer project at Princeton Institute of Advanced Study, charged with development of the EDVAC computer. Previous iterations, such as the ENIAC project, were not universal computing machines rather something built out of wartime necessity to specifically calculate artillery trajections and operated by re-routing cables and flipping switches. Instead of supplying paper tape as in Turing’s hypothetical or movable cables and switches as in ENIAC to read instructions, von Neumann designed a machine that stored the instructions that the computer would follow in its memory alongside its working data. As one commentator described:

Thanks to Turing’s abstract work, , von Neumann knew that, by making use of coded instructions stored in memory, a single machine of fixed structure could in principle carry out any task for which an instruction table can be written…. von Neumann saw that this was the means to make concrete the abstract universal computing machine of “On Computable Numbers.”[[35]](#footnote-61)

In such a machine, the instructions to follow are loaded into storage, and then run. Complex procedures to control the operation of the computer could now but implemented, such as loops and conditional jumps (if *this* then do *that*). This ability is what makes the computer a general-purpose tool of extraordinary power.[[36]](#footnote-62) By loading new procedures into the computer, what was once a universal machine capable of any conceivable computation, is now a specific machine that will accomplish, exactly, only a specific function designated by the whims of its programmer.

In his “First Draft of a Report on the EDVAC,” von Neumann analogized the computer architecture he created, seen in Figure \_\_\_\_, to a human.[[37]](#footnote-63) The computer required a central component comprising an arithmetic unit and control unit, as well as a memory unit. He described these as corresponding to the associative neurons in the human nervous system, as in the brain. All calculation and operation would be carried out by these components. But von Neumann also recognized “a necessity of getting the original definitory information from outside into the device, and also getting the final information, the results, from the device to the outside.”[[38]](#footnote-64) These functions would be performed by typing, punching, photographing light, magnetizing tape or an analogous means that could be operated or sensed by humans.



vonneumann-architecture

What is striking about von Neumann’s architecture and his description of it is the resemblance to our earlier description of an information processor. The computer receives data from the environment as input, processes that data into something useful, and outputs the processed information. At a high-level, we can think of the computer as a drop-in replacement for a human node in an information processing system. As MIT professor Joseph Weisenbaum explained:

We have already agreed that it is entirely proper and even useful … to see man as an information processor. And since the computer, the Turing machine, is a universal information processor, it is natural to compare man as seen from that perspective with the computer.[[39]](#footnote-70)

Of course, computers and humans differ in the internal architecture of their tangible embodiments. One consists of electrical conductors typically made today out of copper and silicon, and the other is fashioned from a network of biological, organic neurons. The differences in the internal architectures means they have different characteristics. Digital computers rely on hard, exacting calculations, which is a result of the binary on-and-off nature of their underlying logic. Humans, on the other hand, think about the world in a fuzzy, probabilistic manner.[^CITE]

And just as the wiring of each human brain is unique, so to do differences exist in general-purpose digital computers. Although the von Neumann architecture is the most common foundation for modern general-purpose computers (and especially in commercial desktop and handheld computers) it is not the only such architecture. The Harvard architecture, for example, is a different kind of stored-program computer that holds program instructions and working data in separate memories, rather than a single memory.[^CITE] And within the von Neumann architecture, memory can be deconstructed into different kinds of physical embodiments, such as a combination of external RAM, comprising fast semiconductor chips, and disk drives, which are slower magnetic discs read from and written to by a moveable arm.[[40]](#footnote-71)

But these distinctions play no role in evaluating the digital computer as an information processor. A computer can be provided measurements for the energy, angle, wind and gravitational influences on ballistic artillery, and its program can calculate a trajectory to hit its target. But so can a human with pen and paper and an understanding of the mathematics of physics. A human can also, without aid of any digital devices or writing to store intermediate results, can hear the crack of a bat, observe the flight of a baseball, and with an eager run and outstretched hand catch the pop fly as it reaches the end of its arc. When we are discussing the processing of data into information, the means by which this transformation is carried out does not matter in the first instance. When information is the subject, there is typically nothing innovative about the type of information processing device employed, whether that tool is a digital computer, a mechanical device, or the human mind itself. Nor would it matter if a new breed of information processor were developed in the form of a better computer–or even a replacement to the digital computer as we know it. The end result is the same: a tool is used to turn data is turned into useful information to create knowledge.

# Information Systems as Patentable Subject Matter

## Patentable Subject Matter: “New and Useful” But Not “Laws of Nature, Physical Phenomena, and Abstract Ideas”

The overarching purpose of the Patent Act is to encourage innovation by leveraging the economic framework of a awarding limited monopoly to inventors. The founding fathers wrote this framework into the Constitution by granting Congress the power “[t]o promote the Progress of Science and Useful Arts, by securing for limited Times to Authors and Inventors of the exclusive Right to their respective Writings and Discoveries.”[[41]](#footnote-74) This clause incorporates a balancing act: granting exclusivity to creators (an intrinsically harmful social act) so long as it advances innovation in the “useful arts.” The first Patent Act attempted to enact this balance by limiting patentable subject matter to “any useful art, manufacture, engine, machine, or device, or any improvement therein not before known or used.”[[42]](#footnote-75) When the framers used the term “useful arts,” they meant what we in modern times call “technology.”[[43]](#footnote-76) One can contrast “useful arts” with the concepts of ‘liberal arts’ and the ‘fine arts.’[[44]](#footnote-77)

Early jurisprudence, carried over from English cases, acknowledged that “principles” are not patentable. In the 1852 case *Le Roy v Tatham*, the Supreme Court stated: “[A] principle is not patentable. A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right.”[[45]](#footnote-78) The Court acknowledged that “[t]he word principle is used by elementary writers on patent subjects, and sometimes in adjudications of courts, with such a want of precision in its application, as to mislead.”[[46]](#footnote-79) Perhaps a better explanation is one by Justice Story, who explained that “[a] principle means an elementary truth, or power.”[[47]](#footnote-80) In a machine that performs its functions by motion, for example, the motion in question is a principle, and the specific machinery employed is an application of the principle.[[48]](#footnote-81)

Justice Story’s example of motion as a principle proved prescient, as years later, in 1853, the Supreme Court addressed addressed that very issue in *O’Reilly v. Morse*, also known as The Telegraph Patent Case.[[49]](#footnote-82) In 1840, Samuel Morse had obtained two patents for his invention of electro-magnetic telegraphs, which were reissued in 1848. Several claims were for specific machinery employed to use of electro-magnetism to send writing over a distance.[[50]](#footnote-83) The Court gave pause, however, when it reached is eighth claim. His patent, however, was not limited to the telegraph machine he had created; in his words:

Eighth. I do not propose to limit myself to the specific machinery or parts of machinery described in the foregoing specification and claims; the essence of my invention being the use of the motive power of the electric or galvanic current, which I call electro-magnetism, however developed for marking or printing intelligible characters, signs, or letters, at any distances, being a new application of that power of which I claim to be the first inventor or discoverer

The Court recognized that in this claim Morse was trying to claim a principle–the *bare idea* of using of a kind of motion, rather than the specific apparatus employing that kind of motion. The claim would have allowed Morse to exclude others from using any methods of writing using electro-magnetism-including those Morse had -not described and indeed had not invented, and therefore could not describe when he obtained his patent.-[[51]](#footnote-84)

The Court in *Oreilly v. Morse* contrasted its claim 8 with a 19th-century English case, *Neilson v. Hartford*, 151 ER 1266 (1841). In that case, the patentee had discovered that heated air could be pumped into a furnace to improve the process of converting high-carbon iron to lower-carbon iron or steel and obtained a patent on the use of a heated chamber to provide the heated air.[[52]](#footnote-85) The defendants argued that the patent covered merely the abstract principle–a fundamental truth–that injecting hot air instead of cold air into a furnace created better iron.[[53]](#footnote-86) The Exchequer Court disagreed, holding that because even assuming this principle were well-known cannot be patented, the patentee had created a mechanical mode of applying the principle to furnaces. The invention interposed a receptacle between a blowing apparatus and a furnace, and in the receptacle allowed the application of external heat to warm the air before it was introduced into the furnace.[[54]](#footnote-87) Thus, the patent was allowed because it claimed a specific manner for applying a principle, not the principle itself.

In 1874, the Supreme Court once again addressed patent-eligibility in *Rubber-Tip Pencil Co. v. Howard.*[[55]](#footnote-88) In that case, the Court acknowledged that the patent was for a particular manufacture: a rubber eraser head that could be used to attach to a lead pencil. Thus, it faced a scenario more akin to that in *Neilson* rather than *Morse*. Yet the Court found the patent ineligible because it concluded it claimed no more than a bare idea, even though it claimed a useful thing! First, the Court pointed out that the patent was not for the combination of the rubber head with the pencil. This patent was for the rubber head by itself, and that was a different matter. The Court then examined the claim to identify what the patentee identified as the inventive aspects. The Court described the invention as being rubber with a hole, without the need for a particular shape of the head, and without any particular form and shape of the hole. None of these pointed to an invention because it meant that “[a]ny piece of rubber with a hole in it is all that is required thus far to meet the calls of the specifications.” The last aspect put forth in the patent, that the cavity must be slightly smaller in diameter than whatever pencil into which it is to be used, “add[ed] nothing to the”patentable character of the invention.“[[56]](#footnote-89) That part was conventional, because”[e]verybody knew, when the patent was applied for, that if a solid substance was inserted into a cavity in a piece of rubber smaller than itself, the rubber would cling to it.“[[57]](#footnote-90) The patent amounted to nothing more than claiming a piece of rubber with a hole, along with”not new" ideas that: rubber will attach to anything larger than the hole into which that thing is inserted, and rubber is good for erasing.

These cases demonstrate that the inquiry is one of whether the claim seeks to patent a bare idea itself, rather than a specific application of that idea. Subsequent cases expand upon how to describe what are bare ideas: they are “scientific truth[s], or the mathematical expression of it,”[[58]](#footnote-91) “forces of nature,”[[59]](#footnote-92) “natural phenomena,”[[60]](#footnote-93) and, unhelpfully, “abstract ideas.”

In *The Telephone Cases*, in 1888, the Court tried once again to explain how to tell the difference between discovery of a principle in the abstract, and a patentable invention:

In this art–or, what is the same thing under the patent law, this process, this way of transmitting speech–electricity, one of the forces of nature, is employed; *but electricity, left to itself, will not do what is wanted. The art consists in so controlling the force as to make it accomplish the purpose.* It had long been believed that if the vibrations of air caused by the voice in speaking could be reproduced at a distance by means of electricity, the speech itself would be reproduced and understood. *How to do it was the question.*

Bell discovered that it could be done by gradually changing the intensity of a continuous electric current, so as to make it correspond exactly to the changes in the density of the air caused by the sound of the voice. *This was his art. He then devised a way in which these changes of intensity could be made and speech actually transmitted.* Thus his art was put in a condition for practical use.[[61]](#footnote-94)

The complexity of this analysis, unfortunately, appeared to rise exponentially when innovations involving to general purpose computers began to arrive at the patent office, and for decades the Court struggled with a framework for evaluating the eligibility of computer-related patents.

## The Arrival of Digital Computers at the Supreme Court

The Court addressed patent-eligibility issue as it involved computers in piecemeal through a trio of cases that, for a long time formed the foundation of Section 101 analysis. The first was *Gottschalk v. Benson*.[[62]](#footnote-96) The patent in *Benson* involved a particular way of using a shift register to convert binary coded decimal (BCD) form to binary.[[63]](#footnote-97). BCD form is a mapping of each digit of decimal number (such as the ‘5’ and ‘3’ in 53) with its pure binary representation in ones and zeros (such as 0101 for ‘5’ and 0011 for ‘3’ to make 0101 0011 for 53). The claimed method recite steps to convert a number represented in BCD form to its binary representation .

The Supreme Court held that the patent was -so abstract and sweeping as to cover both known and unknown uses of the BCD to pure binary conversion,- that the patent was ineligible.[[64]](#footnote-98) The Supreme Court’s concerns were two-fold: 1) the practical effect of the patent was a monopoly on the algorithm for converting from BCD to pure binary; and 2) because the patent had -no substantial practice application except in connection with a digital computer,- the restriction of the invention to a computer was, in essence, no restriction at all.[[65]](#footnote-99)

*Benson* can be seen as simply a continuation, in the digital age, of the same principles dating back to *O’Reilly v. Morse* and its progeny about attempts to claim bare principals. Unfortunately, *Benson* is also a misapplication of those principals. There were two claims-at-issue: Claim 8 involved a method using a computer component called a “reentrant shift register,” and applying a series of adding, masking, and shifting steps to the register reach the conversion result. Claim 13, however, did not involve any special components, and more generically described its steps as testing, detecting, and repeating.

The Court’s concern about patenting the mathematical formula for the conversion doesn’t apply equally to both claims. Whereas Claim 13 reads like a description of how a human might tackle the BCD conversion process, Claim 8 differs greatly in form and substance because it describes how to perform the conversion entirely within the specific shift register device. The Court did not analyze these two claims separately. In fact, the Court recognized that “the method sought to be patented *varies the ordinary arithmetic steps a human would use* by changing the order of the steps, changing the symbolism for writing the multiplier used in some steps, and by taking subtotals after each successive operation.”[[66]](#footnote-100) This variation is no accident. It is a specific change to the general mathematic principles underlying the conversion in order to achieve the end result using only a single, specific computing device: a reentrant shift register.[[67]](#footnote-101)

The CCPA did recognize the distinction. Claim 8, it held, was patentable because it involved a “particular apparatus” of a shift register.[[68]](#footnote-102) Claim 13 differed because it “could *in theory* be any kind of writing implement and any kind of recording medium.” Yet, Claim 13 “had no practical use other than the more effective operation and utilization of a machine known as a digital computer,” and thus was patentable because it involved the “useful arts.” The Supreme Court’s criticism only has application against Claim 13, and ignores any important differences due to the specific apparatus in Claim 8. Commenters of all stripes have since criticized the holding in *Benson* as a political battle of punting the football between Congress, the Patent Office, and the courts, for who should set guidelines on patenting (or not) innovations in the emerging digital computer space. Donald Chisum wrote for example:

Benson is a failure. The failure is perpetuated in *Bilski* by deadlock in the Supreme Court. The persistence of controversy over the patentability of software for 38 years and the inability of a majority in the Supreme Court to provide significant guidance on the patentability of intangible methods verify the weakness of Benson’s reasoning. The vagueness of the reasoning in Benson temporarily served the interests of those opposed to software patenting, but in the long run, *Benson* served no one’s interest, certainly not the public interest Its ambiguity allowed software patent proponents to subvert any bar that software patent opponents desired. The ambiguity also deterred legitimate inventors of software-implemented inventions from applying for patent protection.[[69]](#footnote-103)

The next decision came in *Parker v. Flook*, where the Supreme Court analyzed the eligibility of a patent describing a method of updating alarm limits during catalytic conversion processes.[[70]](#footnote-104) The patentee argued that a particular mathematical formula was useful when -updating the value of an alarm limit on any process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons;- however, his formula could be used outside of simply catalytic conversion processes.[[71]](#footnote-105) The patentee argued that, because his patent did not pre-empt all uses of the formula, the method was patent-eligible.[[72]](#footnote-106) The Supreme Court disagreed. Limiting the use of the algorithm to a particular end-use (here, for the conversion of hydrocarbons) added only purely conventional or obvious post-solution activity. The Court suggested that the formula, even if newly discovered, must be assumed to have already been known as part of the eligibility analysis. When this formula was viewed in context with the rest of the claim, which was merely activity conventionally known in the industry, the patent was ineligible.[[73]](#footnote-107)

*Flook* extended the holding in *Benson,* which reasoned that pure mathematics was not eligible, to note that limiting the use of the formula to a specific use could not make an invention eligible. It must be applied in a novel setting. Otherwise the claim amounts to monopolizing the “basic tools of scientific and technological work.” Here, the use of a general-purpose computer for “automatic monitoring-alarming” “simply provides a new and presumably better method for calculating alarm limit.”

Whether any invention involving computers could be patented remained unclear until the last of the pre-*Bilski* trio, *Diamond v. Diehr*. That case involved a -process for molding raw, uncured synthetic rubber into cured precision products.-[[74]](#footnote-108) Industry practice had used a formula to calculate the time when a mold should be opened to create a perfect, cured product.[[75]](#footnote-109) However, consistently creating perfectly cured products was difficult because one of the variables in the formula, the temperature of the press, could not be precisely measured.[[76]](#footnote-110) The patentee solved the problem by measuring the temperature inside the press, feeding the constant temperature measurements to a computer, and, based on the formula, automatically opening the press.[[77]](#footnote-111) After looking at the claim as a whole, the Supreme Court held that the patent was not an attempt to patent a mathematical formula; instead, the patent claimed a particular *application* of that formula and was thus eligible for protection.[[78]](#footnote-112)

*Diehr* halted the inevitable extension of *Flook* to the point of making all inventions involving general purpose computers unpatentable. The distinction between patents the two cases, both claiming the use of mathematical formulas, is what the former did with its results. Whereas in *Flook* the result of the claim was merely an updated alarm limit, the claim in *Diehr* resulted in activating the opening of the press. It was thus purportedly a novel application of mathematic principles to achieve an improved result of a technological process:

The claims describe a process of curing rubber beginning with the loading of the mold and ending with the opening of the press and the production of a synthetic rubber product that has been perfectly cured—a result heretofore unknown in the art.

After *Diehr*, a long period of neglect at the Supreme Court followed. Then, since a few years ago, the Court has taken a more proactive approach toward defining the direction of patent law jurisprudence. The Court has looked particularly closely at patent-eligibility and addressed ambiguities in that area of law in a trio of cases. These opinions, however, have still provided very little concrete guidance as to the inner and outer limits of patent-eligibility, especially as it concerns information systems.

## The Return to Section 101

In 2010, the Supreme Court returned to the issue of patent-eligibility in *Bilski v. Kappos*, while considering a patent that purportedly covered hedging of risk.[[79]](#footnote-114) The Court decided to address the issue of whether the Machine-or-Transformation test should be used in analyzing whether a method patent covered eligible subject matter.[[80]](#footnote-115) The Machine-or-Transformation test articulated a simple, bright-line rule: a method is not patent-eligible unless it is tied to a particular machine or apparatus (machine), or transforms an article into a different state or thing (transformation).[[81]](#footnote-116) The Federal Circuit had held that this test was the *sole* test for patent-eligibility.[[82]](#footnote-117) Though the Supreme Court ultimately agreed that the patent was ineligible, it disagreed with the Federal Circuit’s reasoning.[[83]](#footnote-118) In a widely anticipated opinion among practitioners, the Supreme Court emphatically declared that the Machine-or-Transformation test was a -useful and important clue [and] an investigative tool- for determining the eligibility of a claimed method, but it was not the sole test.[[84]](#footnote-119) The Supreme Court also made clear that Section 101 did not eliminate all business method patents either.[[85]](#footnote-120)

Ultimately, the Supreme Court found the patent in *Bislki* invalid for claiming an abstract idea without adding anything more to the underlying abstract principle.[[86]](#footnote-121) Only a cursory examination of the patent took place, however. The Supreme Court also did not articulate a framework to take the place of the Machine-or-Transformation test.[[87]](#footnote-122) It would take two more cases to begin structuring a replacement framework.

In *Mayo Collaborative Servs. v. Prometheus Labs., Inc.,*[[88]](#footnote-123) the asserted patents involved the -use of thiopurine drugs in the treatment of autoimmune diseases, such as Crohn’s disease and ulcerative colitis.-[[89]](#footnote-124) The inventors of the asserted patents had discovered parameters which could be used to determine the range of effective dosages of thiopurine drugs for an individual patient and had patented a method based on these parameters.[[90]](#footnote-125) When analyzing the patent claims, the Supreme Court recited a two-step analysis. The first step required determining whether a patent preempted the use of a natural law.[[91]](#footnote-126) SCOTUS found that the patent set forth -relationships between concentrations of certain metabolites in the blood- and the effectiveness of a dosage-effectively, a law of nature.[[92]](#footnote-127) The second step required the court to determine whether there existed some inventive concept that could transform the patent and make it cover eligible subject matter.[[93]](#footnote-128) The Supreme Court found that the three additional steps in the patent claims did not:

To put the matter more succinctly, the claims inform a relevant audience about certain laws of nature; any additional steps consist of well-understood, routine, conventional activity already engaged in by the scientific community; and those steps, when viewed as a whole, add nothing significant beyond the sum of their parts taken separately.[[94]](#footnote-129)

The asserted patents, therefore, were ineligible for patent protection.[[95]](#footnote-130) In particular, as in *O’Reilly v. Morse*, the Supreme Court worried about preemption: the possibility of the asserted patents covering all ability of -doctors to apply the applicable laws when treating their patients.-[[96]](#footnote-131). Yeta a question still remained as to whether and how this framework applied to method patents beyond the realm of biological sciences.

In *Alice Corp. PTY v. CLS Bank Int-l*, the Supreme Court answered that question and adopted the *Mayo* two-step framework as the framework for analyzing patent eligibility of method patents–regardless of the technological field.[[97]](#footnote-132) The *Alice* patent involved a method of -exchanging financial obligations between two parties using a third-party intermediary to mitigate settlement risk.-[[98]](#footnote-133) The Supreme Court applied the *Mayo* framework and affirmed that the patent was invalid for covering patent-ineligible subject matter.[[99]](#footnote-134) As the first step of the *Mayo* analysis, the Supreme Court held that the patent itself covered the abstract idea of intermediated settlement, a concept which was a -fundamental economic practice long prevalent in [] system[s] of commerce.-[[100]](#footnote-135) At the second step, the Supreme Court analyzed whether the additional steps of the method, including the use of a computer to maintain -shadow records- to reconcile those transactions for which a party has sufficient resources, amounted to an -inventive concept- that would transform the patent beyond the abstract idea; they did not.[[101]](#footnote-136) Again, one of the major concerns of the Supreme Court was the issue of preemption.[[102]](#footnote-137) Here, the Supreme Court emphasized that a general purpose computer, by itself or performing conventional steps, failed to make an ineligible patent eligible.[[103]](#footnote-138)

## A Synthesis of the Court’s Response to Information Systems

The Court’s recent jurisprudence has reasoned by reference to “abstract ideas,” “laws of nature,” or “fundamental principles.” These phrases are not helpful because when evaluating inventions involving information systems because they too are amorphous concepts. Information systems, such as general purpose computers, are often not themselves abstract–they are in fact often typically tangible embodiments such as a specially-written software program running on specifically-procured computers. And all inventions, even the most physical, can be reduced to an abstract notion or idea, even if that notion is something like “improve reliability by minimizing the number of moving parts” or “use electro-magnetism to transmit characters over a distance.”

The balance the Court has sought has been to permit monopolization of only specific *applications* of ideas to a new and useful end, whether the idea was new or not. In the *Telegraph Case* the Court permitted several apparatuses claimed for an electromagnet telegraph, but not the idea of an electromagnet telegraph itself regardless of form, even if no one had before used electromagnetics to print characters at a distance. In the *Rubber-Tip Pencil* case, the Court suggested a specific claim for a combination of a pencil and rubber tip may be eligible (although it may fail for novelty or obviousness), but the patentee could not claim using any rubber, regardless of form, with the idea of attaching to it a pencil.

Once computers emerged, the Court immediately recognized that the use of general-purpose computers added nothing to attempts to claim basic principles. Benson could not claim a mathematic formula even if he had limited its use to computers. A simpler way to understand the problem in Benson is to view it as a claim for a *naked information system*. Benson’s claim, described as claim for an algorithm, was for an information system because it accepted data (a number in BCD-form) and processed the data to more useful information in binary form. Because it was nothing more than an information system, it was nothing more than the mathematic principle itself, could be performed by humans, and provided no further advancement than the mathematics itself. Benson, according to Justice \_\_\_\_\_, claimed the bare idea of converting numbers from one form to another, regardless of how it was implemented.[[104]](#footnote-140)

The decision in *Parker v. Flook* follows directly from this analysis. Parker claimed an information system that accepted input data (such as the current alarm base, predetermined alarm offset, and present value of a process variable) and processed the data using a mathematic formula to create information in the form of an updated alarm limit. The problem was that the claim was for nothing further than the information system. Just as in *Morse*, the claim reduced to the bare idea of using a principle (here, the mathematical formula rather than electromagnetics) for a desired purpose (catalytic chemical conversion of hydrocarbons instead of printing characters at a distance). Just as in *Benson*, Parker did not advance the useful arts because he did not claim any new and useful end for the information created from his claimed method.

In *Diehr*, the Court recognized that the claim was for something more than an information system, even though it it was an information system. It was the application of the system to achieve a useful, tangible result. The inventions accepted data inputs (including various constants, elapsed time of the press, and the temperature of the mold) and processed the data to create useful information (by using the Arrhenius equation to determine the total required cure time). The claim took one step further that made it a specific application of this information system for a useful end: “opening the press automatically” based on the output of the information system. Whether this application was novel or non-obvious was a different question.[^CITE] It met the threshold of eligibility because it was not merely a computer-based information system transforming data into information.

In *Bilski*, the Court addressed the issue more generally by looking an information system that did not involve computers, but involved instead people and organizations. The claim involved initiating “transactions,” such as purchases. These transactions are mere data or information: knowledge that a consumer agrees to purchase a commodity from a provider at a particular rate. Thus, the claim, viewed as an information system, takes data as input (prior purchases and market participants for the commodity having a counter-risk position) and processes the data to generate new information (other transactions that would balance the provider’s risk position).

The claims in *Mayo* were also directed at non-computer information systems. In fact, the claim very closely tracks our example of a doctor diagnosing a patient in Section \_\_\_\_. In the claimed method a doctor–or anyone or thing, really–accepted input data (determining the level of 6-thioguanine introduced by a drug) and producing information from the data (whether to increase or decrease the drug dose based on whether the level is above or below a specific range). The claim did not address a specific implementation for this system, such as a new way to administer the drug or determine its concentration in blood, and thus claimed no more than the basic natural principle of the ideal dosing for the drug. Nor did it apply the information about the drug levels to a new, useful end. Its steps stopped at the generating of information, and thus achieved nothing more than the updated alarm limits in *Flook*.

Finally, *Alice* addressed a computer-based information system creating and generating information from data. It was more complex than previous, inserting a third party that helped the system exchange information (credits or debits) created by processing data such as start-of-day and end-of-day balances of participating financial institutions. The involvement of a computer was only nominal.

It can be seen that each of these cases since *Benson* involve an information system or process performed by such a system, whether computer-implemented or achieved by people alone. With this understanding we can create a more specific framework than the two-step process in *Mayo* that

# Framework for Analysis

## Does the inventor claim an information system?

## Is the inventor applying the information system to create a “new and useful” machine or transformation of a “thing”?

## Is the inventor claiming a “new and useful” solution for a problem arising from implementing an information system in a particular technological environment?

## Finally, has the inventor claimed a specific technological embodiment sufficiently narrow, such that preemption of the idea is proportionate to the inventor’s contribution?

# Conclusion

1. Is Semantic Information Meaningful Data?, Philosophy and Phenomenological Research Vol. LXX, No. 2, March 2005. ## The Information System [↑](#footnote-ref-24)
2. See also: \* Information is data that has been processed into a form that is meaningful to the recipient. Davis and Olson, Management Information Systems (1985), 200. \* Data is the raw material that is processed and refined to generate information. Silver and Silver, Systems Analysis and Design (1989), 6. \* Information equals data plus meaning. Checkland and Scholes. Soft Systems Methodology in Action (1990), 303. \* Luciano Floridi, The Philosphy of Information (2011) [↑](#footnote-ref-25)
3. Wil van der Aalst and Christian Stahl, Modeling Business Processes (ISBN 9780262015387) http://mitpress.mit.edu/sites/default/files/9780262015387\_sch\_0001\_0.pdf [↑](#footnote-ref-26)
4. Eileen M. Trauth et al, Information Literacy: An Introduction to Information Systems (1st Ed. 1991) [↑](#footnote-ref-27)
5. Ken Laudon and Jane Laudon, Management Information Systems (12th ed. 2012) (“Information systems are interrelated components working together to collect, process, store, and disseminate information to support decision making, coordination, control, analysis, and visualization in an organization.”)" [↑](#footnote-ref-28)
6. Wil van der Aalst and Christian Stahl, Modeling Business Processes (ISBN 9780262015387) http://mitpress.mit.edu/sites/default/files/9780262015387\_sch\_0001\_0.pdf [↑](#footnote-ref-29)
7. Cf. John von Neumann, First Draft of a Report on the EDVAC (June 30, 1945). [↑](#footnote-ref-32)
8. “[A]n information system manages and processes information. This definition is general and allows different interpretations. For example, it is not clear whether ‘information system’ refers only to software systems or also to humans, such as a family doctor who manages and processes information.” Wil van der Aalst and Christian Stahl, Modeling Business Processes (ISBN 9780262015387) http://mitpress.mit.edu/sites/default/files/9780262015387\_sch\_0001\_0.pdf [↑](#footnote-ref-33)
9. See Joseph Valacich and Christoph Schneider, Information Systems Today - Managing in the Digital World (4th Ed. 2010) (“Information systems are combinations of hardware, software, and telecommunications networks that people build and use to collect, create, and distribute useful data, typically in organizational settings”). [↑](#footnote-ref-34)
10. Eileen M. Trauth et al, Information Literacy: An Introduction to Infromation Systems (1st Ed. 1991) at 87. [↑](#footnote-ref-35)
11. See Joseph Valacich and Christoph Schneider, Information Systems Today - Managing in the Digital World (4th Ed. 2010) (“Information systems are combinations of hardware, software, and telecommunications networks that people build and use to collect, create, and distribute useful data, typically in organizational settings”). [↑](#footnote-ref-36)
12. Enfish, LLC v. Microsoft Corp., No. 2:12-CV-07360-MRP, 2014 WL 5661456 (C.D. Cal. Nov. 3, 2014) (noting “[f]or millennia, humans have used tables to store information.” and citing Martin Campbell-Kelly et al., The History of Mathematical Tables: From Sumer to Spreadsheets 19 (Oxford 2003), for showing an example of ancient Mesopotamian table for year 1295 B.C.). [↑](#footnote-ref-37)
13. Heather Phillips, The Great Library of Alexandria?, Library Philosophy and Practice (Aug. 2010) (http://unllib.unl.edu/LPP/^phillips.htm) [↑](#footnote-ref-38)
14. For example, one set of commentators recognized the fungibility of the definition of “information systems” and for their purposes limited it only to those implemented as a software system. Wil van der Aalst and Christian Stahl, Modeling Business Processes (ISBN 9780262015387) http://mitpress.mit.edu/sites/default/files/9780262015387\_sch\_0001\_0.pdf [↑](#footnote-ref-39)
15. See, e.g., Eileen M. Trauth et al, Information Literacy: An Introduction to Infromation Systems (1st Ed. 1991) at 87 (listing several information processing tools). [↑](#footnote-ref-40)
16. See, e.g., Eileen M. Trauth et al, Information Literacy: An Introduction to Infromation Systems (1st Ed. 1991) at 87 (listing several information processing tools). [↑](#footnote-ref-41)
17. See, e.g., Eileen M. Trauth et al, Information Literacy: An Introduction to Infromation Systems (1st Ed. 1991) at 87 (listing several information processing tools). [↑](#footnote-ref-42)
18. “Once upon a time, a ‘computer’ was a human being, usually female, who did calculations set for her by men in suits.” John Naughton, The true fathers of computing, The Guardian, http://www.theguardian.com/technology/2012/feb/26/first-computers-john-von-neumann (Feb. 25, 2012). [↑](#footnote-ref-44)
19. See Alan Turing, “On computable numbers, with an application to the Entscheidungsproblem”, Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265, at 231, available at http://plms.oxfordjournals.org/content/s2-42/1/230.full.pdf (comparing “a man in the process of computing a real number to a machine which is only capable of a finite number of conditions”). [↑](#footnote-ref-45)
20. An interesting gallery of mechanical calculators and computers can be found on the blog posting located at http://io9.com/the-strangest-and-most-beautiful-calculators-humanity-h-1668731927 (accessed Dec. 14, 2014). [↑](#footnote-ref-46)
21. Ifrah, Georges, The Universal History of Computing (2001). [↑](#footnote-ref-47)
22. See “Slide Rule, 5-inch, Pickett N600-ES, Apollo 13,” Smithsonian National Air and Space Museum, http://airandspace.si.edu/collections/artifact.cfm?object=nasm\_A19840160000 (accessed Dec. 14, 2014). During the Apollo missions, astronaut crews carried a slide rule for routine calculations that were not performed by the on-board critical guidance and navigation computer. Id. See generally, The Slide Rule: A Computing Device That Put A Man On The Moon, NPR, http://www.npr.org/blogs/ed/2014/10/22/356937347/the-slide-rule-a-computing-device-that-put-a-man-on-the-moon (Oct. 22, 2014). [↑](#footnote-ref-48)
23. Georg Trogemann, Alexander Y. Nitussov, Wolfgang Ernst, Computing in Russia (2001). [↑](#footnote-ref-49)
24. It was noted that at the time that Babbage believed “he can, by his engine, form the product of two numbers, each containing twenty figures, in three minutes.” See “Sketch of the Analytical Engine Invented by Charles Babbage,” Charles Babbage on the Principles and Development of the Calculator at 242 (1989). Some calculations would have required supplying punch cards numbering in the tens of thousands. Id. Such calculations are handled internally by modern computer processors in less than a microsecond.[CITE] [↑](#footnote-ref-50)
25. Alan Turing, “On computable numbers, with an application to the Entscheidungsproblem”, Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265. Turing’s publication competed with a parallel development by Alfonso Church. Together, their unified thesis is often referred to as the Church-Turing principle. [↑](#footnote-ref-51)
26. Joseph Weisenbaum, Computer Power and Human Reason, 62 (1976). [↑](#footnote-ref-52)
27. See Joseph Weisenbaum, Computer Power and Human Reason, 64 (1976) (“Therefore, whenever we believe we understand a phenomenon in terms of knowing its behavioral rules, we ought to be able to express our understanding in the form of a computer program.”). [↑](#footnote-ref-53)
28. The Church-Turing-Deutsch principle states that a universal computing device can simulate every natural *physical* process, not just abstract mathematical ideas. See Deutsch, D. (1985). “Quantum theory, the Church-Turing principle and the universal quantum computer”. Proceedings of the Royal Society (London) (400): 97-117. [↑](#footnote-ref-54)
29. Joseph Weisenbaum, Computer Power and Human Reason, 63 (1976). [↑](#footnote-ref-55)
30. Stella: A multi-platform Atari 2600 VCS emulator, http://stella.sourceforge.net/docs/index.html. [↑](#footnote-ref-56)
31. A Comparison of Solutions for Running Windows or Linux Software on a Macintosh, http://www.macwindows.com/emulator.html. [↑](#footnote-ref-57)
32. A Comparison of Solutions for Running Windows or Linux Software on a Macintosh, http://www.macwindows.com/emulator.html. [↑](#footnote-ref-58)
33. John von Neumann, First Draft of a Report on the EDVAC (June 30, 1945) [↑](#footnote-ref-59)
34. For example, much of modern cryptography and encrypted communications rely on “one-way functions.” These are mathematical formulas that are quickly calculated to give a solution, but to undo the calculation (if one first had the solution in hand) requires relatively much longer time. Security relies solely upon the fact that the reversal of these formulas by a computer are *impractical* in time, even if possible. One can rest assured that a message will remain secret long after its secrecy is no longer important [↑](#footnote-ref-60)
35. B. Jack Copeland and Diane Proudfoot, “The Computer, Artificial Intelligence, and the Turing Test,” Alan Turing: Life and Legacy of a Great Thinker, 321 (Ed. Christof Teuscher 2004). [↑](#footnote-ref-61)
36. Edwin D. Reilly, Milestones in Computer Science and Information Technology 245 (2003). [↑](#footnote-ref-62)
37. John von Neumann, First Draft of a Report on the EDVAC (June 30, 1945). [↑](#footnote-ref-63)
38. John von Neumann, First Draft of a Report on the EDVAC (June 30, 1945). [↑](#footnote-ref-64)
39. Joseph Weisenbaum, Computer Power and Human Reason, 170 (1976). [↑](#footnote-ref-70)
40. In modern computer architecture, there are significantly many more permutations of memory structures involved that are employed based upon their various trade-offs of speed, energy, and reliability, including: internal computer registers, central cache, external ROM, external RAM, solid-state drives, magnetic discs, optical discs, magnetic tape, and even separate general-purpose computers programmed to store information received over a communications network (commonly referred to as “cloud” storage, provided by services Apple iCloud, Google Drive, or Dropbox). [↑](#footnote-ref-71)
41. U.S. Const. art. 1, sec. 8, cl. 8. [↑](#footnote-ref-74)
42. Patent Act of 1970, ch. 7 sec. 1, 1 Stat. 109, 110. [↑](#footnote-ref-75)
43. In re Bilski, 545 F. 3d 943, 1001 (Fed. Cir. 2008) (citing Karl B. Lutz, Patents and Science: A Clarification of the Patent Clause of the U.S. Constitution, 18 Geo. Wash. L.Rev. 50, 54 (1949) (“The term ‘useful arts’ as used in the Constitution…is best represented in modern language by the word technology.’”)). [↑](#footnote-ref-76)
44. James S. Sfekas, Controlling Business Method Patents: How the Japanese Standard for Patenting Software Could Bring Reasonable Limitations to Business Method Patents in the United States, 16 Pac. Rim. L. & Pol’y J. 197, 214 (2007). [↑](#footnote-ref-77)
45. Le Roy v. Tatham, 55 U.S. 156, 174-145 (1852). [↑](#footnote-ref-78)
46. Le Roy v. Tatham, 55 U.S. 156, 174-145 (1852). [↑](#footnote-ref-79)
47. 2 F. Cas. 914 (C.C.D. Mass. 1818) (No. 1,047). [↑](#footnote-ref-80)
48. 2 F. Cas. 914 (C.C.D. Mass. 1818) (No. 1,047). [↑](#footnote-ref-81)
49. *O’Reilly v. Morse*, 56 U.S. 62 (1853). [↑](#footnote-ref-82)
50. *See id*. at 68-69. [↑](#footnote-ref-83)
51. *Id*. [↑](#footnote-ref-84)
52. Need cite. [↑](#footnote-ref-85)
53. Need cite. [↑](#footnote-ref-86)
54. Need cite. [↑](#footnote-ref-87)
55. *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. 498 (1874) [↑](#footnote-ref-88)
56. *See id*. at 500. [↑](#footnote-ref-89)
57. *See id*. at 507. [↑](#footnote-ref-90)
58. Mackay Radio & Telegraph Co. v. Radio Corp. of America, 306 US 86 (1939). [↑](#footnote-ref-91)
59. The Telephone Cases, 126 US 1 (1888). [↑](#footnote-ref-92)
60. Diamond v. Diehr, 450 US 175, 185 (1981). [↑](#footnote-ref-93)
61. The Telephone Cases, 126 US 1 (1888). [↑](#footnote-ref-94)
62. *Gottschalk v. Benson*, 409 U.S. 63 (1972). [↑](#footnote-ref-96)
63. *See id*. at 73-74. [↑](#footnote-ref-97)
64. *See id*. at 68. [↑](#footnote-ref-98)
65. *See id*. at 71-72. [↑](#footnote-ref-99)
66. 409 U.S. at \_\_\_\_ (emphasis added). [↑](#footnote-ref-100)
67. A shift register, by itself, does not have universality computing power as a Turing machine would. It is simply a series of storage locations each holding a 0 or 1 bit of information, with the limited functionality of storing, masking, and shifting these bits left or right. [↑](#footnote-ref-101)
68. In re Benson. 441 F.2d 682, 169 USPQ 548 (1971). [↑](#footnote-ref-102)
69. Donald S. Chisum, Patenting Intangible Methods: Revisiting Benson (1972) After Bilksi (2010). Pre-1972, policy and practice in the Patent Office favored IBM’s no-patents-on-software position. Supporting this position was a recommendation in 1966 by a “President’s Commission on the Patent System.” *Id.* [↑](#footnote-ref-103)
70. *Parker v. Flook*, 437 U.S. 584, 585 (1978). [↑](#footnote-ref-104)
71. *See id*. at 586. [↑](#footnote-ref-105)
72. *See id*. at 589-90. [↑](#footnote-ref-106)
73. *See id*. at 593-95. [↑](#footnote-ref-107)
74. *Diamond v. Diehr*, 450 U.S. 175, 177 (1981). [↑](#footnote-ref-108)
75. *See id*. at 177-78. [↑](#footnote-ref-109)
76. *See id*. at 178. [↑](#footnote-ref-110)
77. *See id*. at 178-79. [↑](#footnote-ref-111)
78. *See id*. at 187, 191-92. [↑](#footnote-ref-112)
79. *Bilski v. Kappos*, 561 U.S. 593, 130 S.Ct. 3218, 3223-24 (2010). [↑](#footnote-ref-114)
80. *See id*. at 3225-26. [↑](#footnote-ref-115)
81. *See id*. [↑](#footnote-ref-116)
82. *See id*. at 3226. [↑](#footnote-ref-117)
83. *See id*. at 3231. [↑](#footnote-ref-118)
84. *See id*. at 3227. [↑](#footnote-ref-119)
85. *Bilski*, 130 S.Ct. at 3228-29. [↑](#footnote-ref-120)
86. *See id*. at 3231. [↑](#footnote-ref-121)
87. *See id*. [↑](#footnote-ref-122)
88. *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, \_ U.S. \_, 132 S.Ct. 1289, 1294 (2012). [↑](#footnote-ref-123)
89. *See id*. at 1294-95. [↑](#footnote-ref-124)
90. *See id*. at 1295. [↑](#footnote-ref-125)
91. *See id*. at 1294. [↑](#footnote-ref-126)
92. *See id*. at 1296. [↑](#footnote-ref-127)
93. *See id*. at 1294. [↑](#footnote-ref-128)
94. *Mayo*, 132. S.Ct. at 1297-98. [↑](#footnote-ref-129)
95. *See id*. at 1305. [↑](#footnote-ref-130)
96. *Id*. at 1298. [↑](#footnote-ref-131)
97. *Alice Corp. PTY v. CLS Bank Int-l*, \_ U.S. \_, 134 S.Ct. 2347, 2355 (2014). [↑](#footnote-ref-132)
98. *Id*. at 2356. [↑](#footnote-ref-133)
99. *See id*. at 2360. [↑](#footnote-ref-134)
100. *See id*. at 2356 (quotations omitted). [↑](#footnote-ref-135)
101. *See id*. at 2359-60. [↑](#footnote-ref-136)
102. *See id*. at 2354 (-We have repeatedly emphasized this concern that patent law not inhibit further discovery by improperly tying up the future use of these building blocks of human ingenuity.-) (quotations omitted). [↑](#footnote-ref-137)
103. *Alice*, 134 S.Ct. at 2360. [↑](#footnote-ref-138)
104. As discussed above in Section \_\_\_\_\_, only Claim 13 was a bare idea had no need for a specific implementation. Justice \_\_\_\_ apparently ignored that Claim 8, like Samuel Morse’s claims one through seven, was for a specific implementation of performing the math on a specific apparatus. [↑](#footnote-ref-140)