

13.1.2: Patent-Eligible Inventions

 Learning Objectives

After completing this section, you will be able to

- Identify the characteristics of a patentable invention.
- Understand what is not patentable and why.

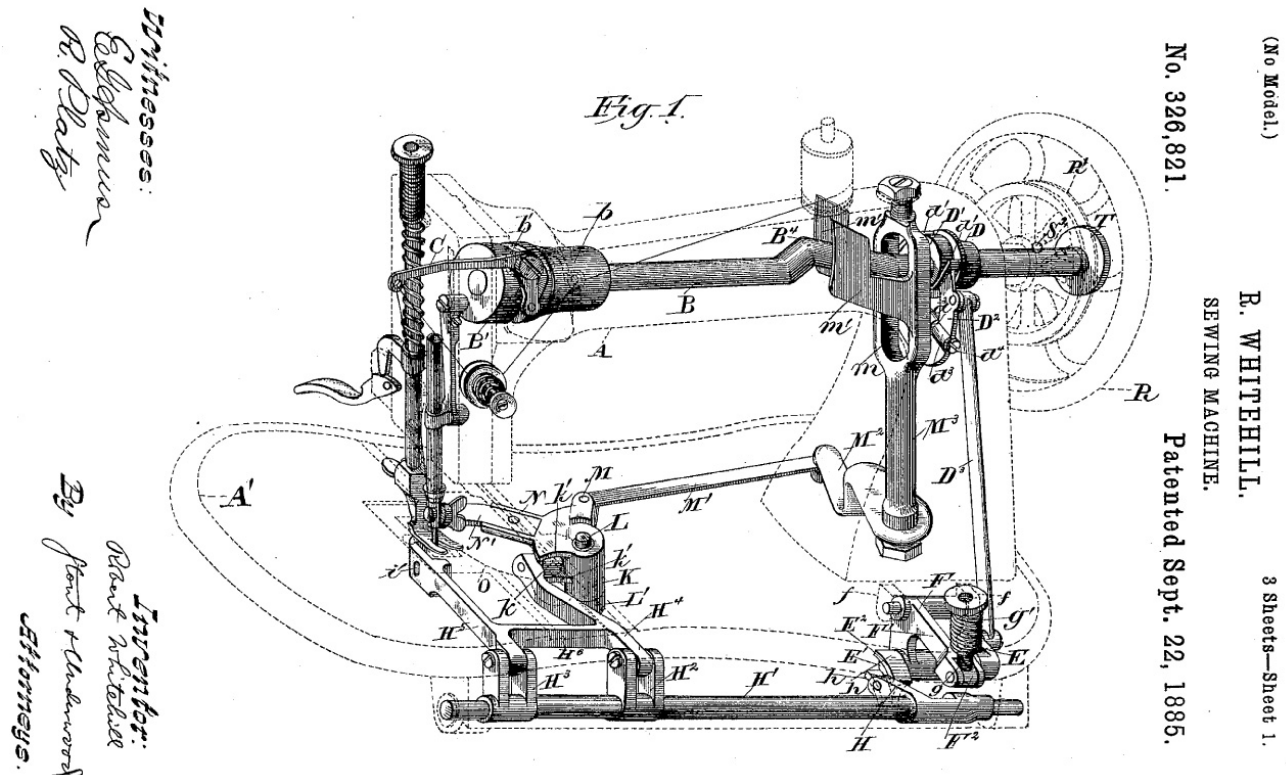


Figure 1.6.1 USPTO Patent No. 326821 for a sewing machine. (credit: USPTO via Wikimedia Commons / Public domain)

What, Exactly, Can You Patent?

Title 35 of the United States Code (the Patent Act) allows anyone who invents or discovers any new, nonobvious, and useful machine, manufacture, process, or composition of matter—or who makes an improvement of any of the above—to obtain a patent.^{[xljii](#)}

These four kinds of patentable inventions fall into one of two categories: They are products or processes.^{xliv}

Product or Process?

Products are physical things—whether they be machines (a new type of robotic welder), manufactures (an artificial knee made of titanium), or compositions of matter (a new chemical “superglue” for binding materials together). In this photo, Elon Musk observes the robotic arms in the Tesla Motors factory. Specialized manufacturing equipment like this are patentable products.

Processes (or methods), on the other hand, are a means to an end—either a means of doing something new (being able to pay for purchases directly from your smartphone), or a new way of doing something old (using “pinch, swipe, and zoom” gestures on a touchscreen, rather than clicking drop-down menus, to manipulate text, music, and images on a smartphone).

You Cannot Patent Ideas

All patented inventions fit into one of these four categories: machine, manufacture, process, or composition of matter. But not everything that fits in one of these four categories can be patented. And the most important reason why one thing is patentable and

another is not lies in the difference between ideas and applications.

You cannot patent an idea for a better mousetrap—not unless it can be developed into a new, non-obvious, and useful machine, manufacture, process, or composition of matter that can actually accomplish the task. You may have a genius idea for faster-than-light travel, but that will not get you a patent unless you can outline how to develop a tangible process or device for actually doing so, in which case you can seek to patent it.

Put another way, “talk is cheap” when it comes to securing a patent. The U.S. Patent and Trademark Office offers no judgment as to the wisdom or desirability of any particular invention—patent No. 2,882,858 for a bird diaper is certainly proof of that. But it absolutely will insist that every invention include a tangible device or process for achieving its intended purpose before it deems the invention worthy of a patent.

Mathematical Formulas Not Patentable

There are other discoveries that fall into the broad category of abstract ideas and are thus unpatentable. You cannot patent a mathematical formula. You cannot patent a law of nature, such as Einstein’s $E=MC^2$. And you cannot patent natural phenomena like electricity (discovered by William Gilbert in 1600) or the Higgs particle that gives all matter its mass (discovered by researchers at the Large Hadron Collider on July 4, 2012). These all exist independently of human intervention, whether we have discovered them or formulated their rules yet or not, and must be freely available to all of humanity for its understanding and betterment.

To restate the distinction, you cannot patent electromagnetism but you can patent a telegraph that uses electromagnetism to communicate rapidly over great distances, as Samuel Morse did in 1840 with patent no. 1647. And although you cannot patent light waves, you can patent a fiber optic wire that employs light waves to communicate even more rapidly and over greater distances, as Corning Glass researchers did in 1970 with patent no. 3,711,262.

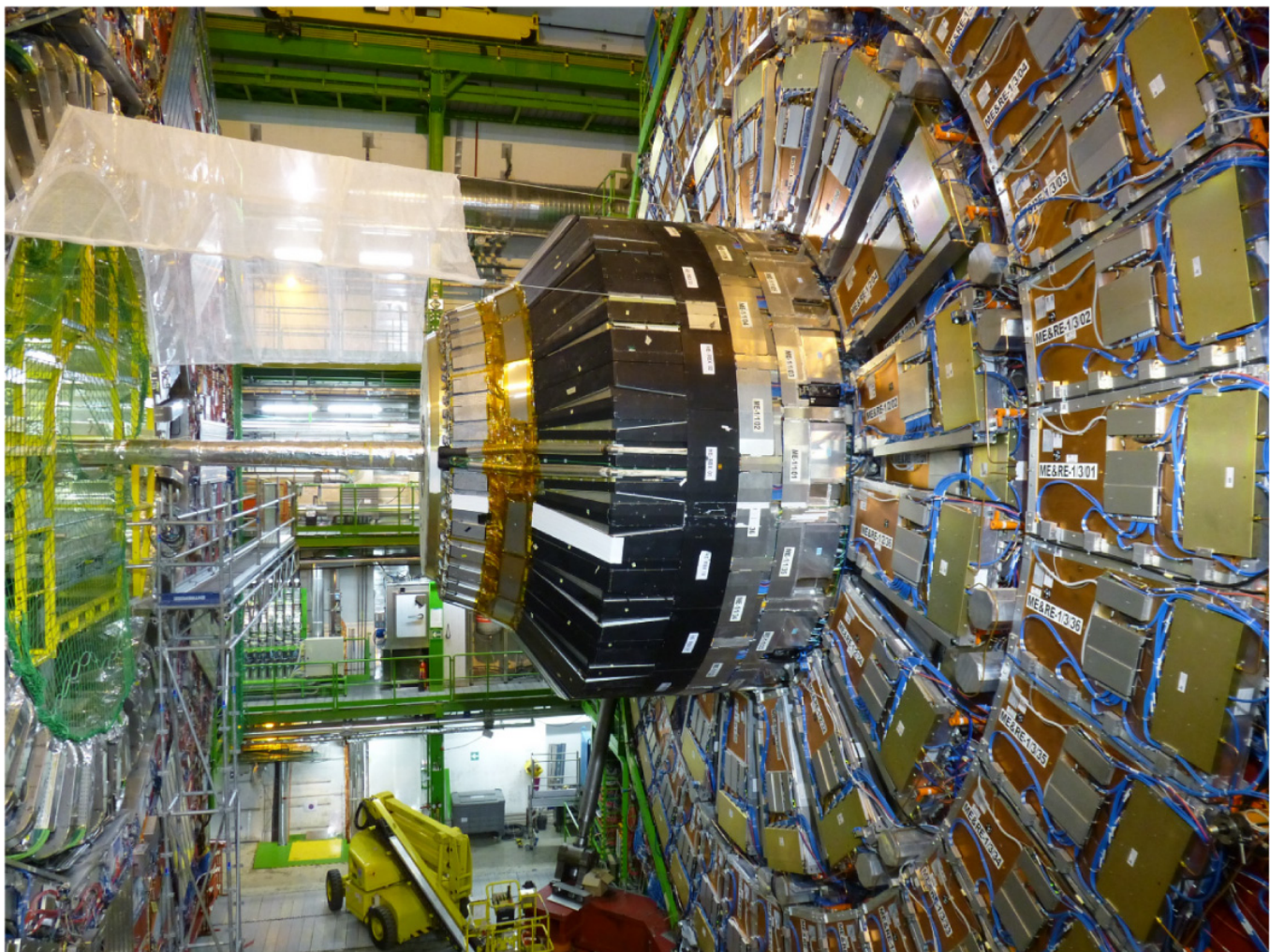


Figure 1.6.2 Portion of the Large Hadron Collider. (credit: Photograph by Luigi Selmi via flickr / CC BY 2.0)

Can You Patent Computer Software?

The boundary between ideas and applications might seem clear, but it has become blurred since the advent of computer technology 40 years ago, especially regarding the patentability of software.

To learn more, watch this video from [PBS Digital Studios](#) about the first software patent ever awarded and to learn a bit about the debate around software patentability.

Mixed Verdicts on Software Patentability

The federal courts and the U.S. Supreme Court have tried to clarify the limits of patentability in the computer age. Three Supreme Court cases in particular—often called the “patent-eligibility trilogy”—reveal the evolution of its thinking about software patentability. In 1972, the Supreme Court held in [Gottschalk v. Benson](#) that an algorithm in a computer program—in this case, a mathematical procedure executed electronically that was similar to long division with paper and pencil—was in and of itself not patentable. Phenomena of nature, mental processes, and abstract intellectual concepts are the basic tools of scientific and technical research, the court noted, and therefore could not be patented lest it foreclose others from using the algorithm and thereby stifle rather than promote technological progress. Granting a patent in this case, the court said, would be analogous to having granted Samuel Morse a patent covering all possible uses of electromagnetism in communications, rather than for the specific method and apparatus he actually invented.

The court made a point of saying, however, that its decision did not mean that computer software could not be patented—only that software whose only useful characteristic was an abstract algorithm could not be patented.

The Supreme Court further refined its thinking on software patentability in the 1978 case [Parker v. Flook](#). Unlike the attempt to patent all uses of an algorithm in the Benson case, here the use of a software algorithm was limited to a specific application—setting off an alarm during the catalytic chemical conversion of hydrocarbons. This was a specific and tangible use of an algorithm, but the Court still ruled the software unpatentable because it felt the application itself was not inventive.

But once again, the Court left the door open: “Even though a phenomenon of nature or mathematical formula may be well known, an inventive application ... may be patented.”

Three years later, the Supreme Court made its third attempt to define the patent-eligibility of software. In [Diamond v. Diehr](#), the Court ruled that although algorithms by themselves are not patentable, a software program that used algorithms to govern the molding of raw synthetic rubber into cured precision products was in fact patentable because it involved “transforming or reducing an article to a different state or thing.”

The Software Picture Blurs Even More

Taken together, the three rulings appeared for a time to arrive finally at a coherent definition of software patentability—namely, that although algorithms by themselves are abstract concepts and therefore unpatentable, software programs may be patented if they employ algorithms to produce a tangible and inventive or transformative result. This view was further augmented by a 1998 U.S. Court of Appeals for the Federal Circuit decision in [State Street Bank v. Signature Financial Group](#), which extended software patentability to software-enabled methods of doing business so long as these produced a “useful, concrete, and tangible result.”

But this definitional equilibrium was not to last. The “useful, concrete, and tangible result” test in [State Street Bank](#) was rejected ten years later by the same court in [In re Bilski](#), which upheld the USPTO’s denial of a patent for a method of hedging risk in commodities trading. The court instead offered a “machine or transformation” test, which allows a software program or business method to be patented only if it is implemented on a specific machine to achieve a special purpose that is novel, nonobvious, and useful; it transforms an article from one thing or state to another.

But in its review of [Bilski v. Kappos](#), the Supreme Court ruled that while the “machine or transformation” test was useful, it was not the only test for patentability. In addition to the “machine or transformation” test, the court decreed (rather vaguely) that any future test should be “grounded in the examples and concepts” expressed in its original “patent-eligibility trilogy” of opinions. They thus reaffirmed that business methods may indeed be patentable.

Finally, a “Pretty Clear” Message

While businesses and the courts were trying to figure out what the other tests for patentability might be, the Supreme Court provided further input with its March 2012 decision in [Mayo Collaborative Services v. Prometheus Laboratories](#). Here, the court ruled that a process enabling physicians to correlate blood test results with medication levels to achieve the most appropriate dosages was ineligible for patent protection.”

But then in June of 2014, The Supreme Court issued what may prove to be its most consequential decision on the patentability of software in the 33 years since [Diamond v. Diehr](#). In [Alice v. CLS Bank](#), the Court ruled that taking some activity that people have been doing for centuries—in this case, holding funds in escrow until a transaction is completed—and then merely “doing it through a computer” did not turn this age-old activity into a patentable new invention.

At first, many observers believed that the effects of the [Alice v. CLS Bank](#) ruling would be very limited. Only the patent in the suit was invalidated, after all, not all software patents. What’s more, the abstract reasoning of the court in its decision did not provide clarity on how the ruling may or may not apply to other kinds of software patents—for example, the sort of software used in manufacturing that was ruled patentable in the 1981 [Diamond v. Diehr](#) case.

But by October of 2014, a series of lower court decisions applying the new [Alice v. CLS Bank](#) standard had invalidated 13 additional software patents. As technology policy journalist Timothy B. Lee noted, “The courts are sending a pretty clear message: you can’t take a commonplace human activity, do it with a computer, and call that a patentable invention.”

“The courts are sending a pretty clear message: you can’t take a commonplace human activity, do it with a computer, and call that a patentable invention.”

-Timothy B. Lee, technology policy journalist

How far reaching will the impact of *Alice v. CLS Bank* be? “This doesn’t necessarily mean that all software patents are in danger,” Lee noted, because the patents involved “were particularly vulnerable to challenge under the new *Alice* precedent. But it does mean that the pendulum of patent law is clearly swinging in an anti-[software] patent direction.”

Overcoming the Alice Paradox

By late 2015, however, it was clear that the *Alice* ruling was having an impact not only on patent law, but also on the innovation process itself within corporate America.

As John Cronin, a former top inventor at IBM and now the CEO of the innovation-on-demand firm ipCreate, observes: “The highest-value products and services today—the ones that increasingly drive margins in business—involve cloud computing, Big Data, machine learning, connectivity, mobility and location-based services, and on-demand and anything-as-a-service software applications and business processes. But ironically, these high-value innovations are also the most difficult to patent nowadays as a result of the Supreme Court’s *Alice* decision.”

Cronin calls this the “*Alice* Paradox,” and it has left many in-house patent groups struggling for a solution. One thing is clear: To be patentable nowadays, software has to take a genuinely-inventive step and either trigger an action, employ a device, or in some other way produce a tangible transformative result.

In addition, smart companies are trying to address patentability issues involving software and business processes much earlier in the innovation cycle, before huge investments are made in R&D that may turn out to be not patentable.

Overall, addressing the “*Alice* Paradox” will be critical for many companies because patenting clearly adds value to a new product or service. In a groundbreaking joint study from Carnegie Mellon University, Georgia Institute of Technology, and Duke University entitled “R&D and the Patent Premium,” economists found that “the patent premium for innovations that were patented is substantial. Firms earn on average a 50% premium over the no patenting case, ranging from 60% in the health-related industries to 40% in electronics.”^{xlv}

The Debate on Software Continues

The debate over the dividing line between patentable versus unpatentable computer software-related inventions continues—in corporate R&D labs and in the courts.

However, there are some who don’t believe that software and business methods should be patented under any conditions. They argue first of all that software is different from other industries—more iterative and more incremental, with each advance building upon thousands of previous advances. Therefore, their thinking goes, software should not be entitled to patents that ought to be more properly reserved for truly breakthrough or revolutionary inventions.

There are two problems with this argument. First, as anyone in the semiconductor, chemical, or medical device industries can attest, innovation is no more iterative, incremental, or cumulative in software than is innovation in many other industries. Indeed, there are probably just as many or more patents for incremental semiconductor inventions that build modestly upon earlier work as there are patents for incremental software inventions that do the same.



Figure 1.6.3 An Apple Macintosh Mouse M0100 Beige from 1984 (left), and an Apple Magic Mouse from 2009 (right) (credit: Photograph by reneko via flickr / CC BY 2.0)

As patent scholar and veteran practitioner Paul Janicke of the University of Houston put it, “There really are no breakthrough inventions—at least not in the sense imagined by these critics. Everything moves one step at a time. In fact, every time I thought I encountered a large leap, it turned out that I didn’t know the full extent of the prior art.”

The second problem with their logic is that the Founding Fathers specifically designed the patent system to encourage precisely this kind of incremental invention, so that ordinary people—using only the basic technical skills possessed by most citizens—could participate in rapidly developing the economy from the ground up. This was a very different approach than that of elitist European patent systems of the day, and it produced results that the rest of the world very soon came to envy (see “Section 1.5: What the U.S. Patent System Wrought”).

As the October 21, 1876, issue of *Scientific American* noted, “In the aggregate the little things—which in England or the continent would not or could not be patented—probably add more to the wealth and wellbeing of the community ... than the great things do.” Or to quote Thomas Jefferson himself: “A smaller [invention], applicable to our daily concerns, is infinitely more valuable than the greatest which can only be used for great objects.”^{xlvi} Any uncertainty over the validity of incremental patenting was removed once and for all by the Patent Act of 1952. Consistent with the Founders’ intentions, U.S. law now explicitly holds that patent eligibility is not restricted solely to revolutionary inventions or “flash of genius” discoveries, but also includes more iterative advances in the state of the technological art so long as these meet the requisite novelty, nonobviousness, and utility criteria.

Evidence Shows Software Patents Don’t Hinder Innovation

Another argument made by critics of software patenting is that patents stifle innovation and foster monopolization in the software industry. Research, however, suggests that this is decidedly not the case.^{xlvi} If anything, in fact, the software industry has become even more innovative, more diversified, and more start-up friendly since patenting became common in the 1990s. You need only look at the huge proliferation of highly innovative start-ups in today’s social media and apps software fields to see just how erroneous the claim is that software patents stifle innovation. Or consider for a moment the fate of the Blackberry, once dominant but within the space of a couple of years superseded by more innovative competing smartphone makers.

Finally, some critics insist that the intangible nature of software ought to disqualify it from patentability. But as noted earlier, the Supreme Court has affirmed repeatedly that although abstract concepts cannot be patented, software that employs algorithms to produce a tangible and inventive result—e.g., software that governs the molding of raw synthetic rubber into cured precision products—may be patented.

It's also helpful to view this issue in a larger context. Forty years ago, 80 percent of the market value of all public companies resided in their tangible physical assets—their plant, equipment, and raw materials. In today's Knowledge Economy, however, it is intangible assets—intellectual property—that make up 80 percent of the market value of public companies.^{[xlvi](#)}

Indeed, the entire history of economic progress on our planet may be described as one long climb by humanity up the ladder of abstraction—from brute force to the subtle use of energy, from wealth derived from tangible resources and industrial machinery to wealth derived from ever-more ingenious ways to deploy that energy and those resources. It seems only logical, therefore, to expect that invention itself should follow a similar trajectory—from the realm of the tangible to the realm of the intangible.

Ironically, the debate over patents for software, business methods, and other intangible inventions is nowhere more heated than on the Internet, itself an intangible realm in which “virtual” businesses launched with little more than hope and electrons (e.g., Facebook) are creating real and substantial wealth in the form of new products, new services, new jobs, and new economic growth for society. Yet strangely, those who have no trouble accepting the Internet as the intangible fruit of information age invention seem to get stuck in industrial age conceptions of what should and should not be patentable.

Odd perhaps, but not surprising. The expansion of patentable subject matter into new and more intangible realms has always met with resistance. Patents involving the use of electricity were condemned 140 years ago, as were biotechnology patents 30 years ago, and of course software patents when they began to appear in large numbers 20 years ago. In each case, critics warned that these new kinds of patents would hold back further scientific discovery and innovation. Yet in each case, innovation and discovery actually intensified and their benefits to society multiplied.

When Gene-Related Inventions Are Patentable

A similar resistance is also developing toward gene-related patents. These began to be issued in significant numbers after the Supreme Court's 1980 ruling in *Diamond v. Chakrabarty*, which upheld the first patent on a newly-created living organism—a bacterium for digesting crude oil in oil spills. Since then, patents have been granted for isolated gene sequences, but so far only on those with known functions and not on naturally occurring genes in humans or other organisms. Patents have also been granted for gene sequences used in diagnostic testing, and on gene sequences that have been altered to make them more useful in a specific application.

In March of 2010, however, a federal district court judge ruled in the case of *Myriad Genetics* that even isolated DNA is fundamentally the same as naturally-occurring DNA and is therefore ineligible for patenting. His ruling was reversed by the U.S. Court of Appeals for the Federal Circuit in July, 2011. But the Supreme Court then set aside that decision and directed the appeals court to once again review the case in light of its March, 2012 *Prometheus* decision. On August 16, 2012, however, the U.S. Court of Appeals for the Federal Circuit once again reaffirmed *Myriad's* right to patent the isolated genes *BCRA1* and *BCRA2*, which are involved in most inherited forms of breast and ovarian cancer.

On June 13, 2013, however, the U.S. Supreme Court finally determined in a unanimous decision that a naturally occurring DNA segment is a product of nature and cannot be patented merely because it has been isolated, thereby invalidating *Myriad's* patents on the *BRCA1* and *BRCA2* genes. The Court did rule, however, that the manipulation of a gene to create something not found in nature—such as a strand of synthetically-produced complementary DNA (cDNA)—could still be eligible for patent protection.

To the average citizen—and perhaps to many patent lawyers as well—all this legal hairsplitting over the limits of patentability in the computer age must seem a bit like the debates in medieval times over how many angels can dance on the head of a pin. But two critical points must be borne in mind regarding these debates.

First, no matter what anyone thinks the limits of patentability in an ideal world ought to be, out in the real world where we actually live, software, business method, and gene patents are multi-billion-dollar facts of life that businesses ignore only at their peril.

Second, whatever confusion may exist today, the debates over patentability in the computer age will almost certainly be resolved eventually to most people's satisfaction, just as all previous debates over patentability have. For if nothing else, the two hundred year-plus history of the courts and the patent office demonstrate a remarkable ability on the part of these institutions to adapt to the challenges posed by new technologies and new economic conditions.

Footnotes

- [xliii](#) 35 U.S.C § 101.
- [xliv](#) Arthur R. Miller and Michael H. Davis, *Intellectual Property: Patents, Trademarks, and Copyright in a Nutshell*. (5th ed., p. 25). St. Paul MN: West Publishing Co., 2007.
- [xlv](#) Ashish Arora, Marco Ceccagnoli, and Wesley M. Cohen, “R&D and the Patent Premium,” Science Direct, International Journal of Industrial Organization Issue 26, 2008.
- [xlvi](#) From Jefferson’s letter to George Fleming in 1815, excerpt from *The Jefferson Cyclopedia*, courtesy of B. Zorina Khan.
- [xlvii](#) Robert Merges, “*Patents, Entry and Growth in The Software Industry*,” University of California at Berkeley School of Law, Berkeley, California, 2006. Retrieved from: http://papers.ssrn.com/sol3/papers.c...ract_id=926204
- [xlviii](#) Op. cit., Ocean Tomo.

This page titled [13.1.2: Patent-Eligible Inventions](#) is shared under a [CC BY 4.0](#) license and was authored, remixed, and/or curated by [OpenStax](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.