No. 08-964

In The
Supreme Court of the United States

BERNARD L. BILSKI and RAND A. WARSAW,

Petitioners,

vs.

JOHN J. DOLL, ACTING UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND ACTING DIRECTOR, PATENT AND TRADEMARK OFFICE,

Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The Federal Circuit

BRIEF OF ROBERT R. SACHS AND DANIEL R. BROWNSTONE AS *AMICI CURIAE* IN SUPPORT OF NEITHER PARTY

ROBERT R. SACHS
Pro Se
DANIEL R. BROWNSTONE
Pro Se
555 California Street
Suite 1200
San Francisco, CA 94104
(415) 875-2410

COCKLE LAW BRIEF PRINTING CO. (800) 225-6964
OR CALL COLLECT (402) 342-2831

QUESTIONS PRESENTED

1. Whether the Federal Circuit erred by holding that a "process" must be tied to a particular machine or apparatus, or transform a particular article into a different state or thing ("machine-or-transformation" test), to be eligible for patenting under 35 U.S.C. § 101, despite this Court's precedent declining to limit the broad statutory grant of patent eligibility for "any" new and useful process beyond excluding patents for "laws of nature, physical phenomena, and abstract ideas."

2. Whether the Federal Circuit's "machine-or-transformation" test for patent eligibility, which effectively forecloses meaningful patent protection to many business methods, contradicts the clear Congressional intent that patents protect "method[s] of doing or conducting business." 35 U.S.C. § 273.

TABLE OF CONTENTS

TABLE OF CONTENTS – Continued

TABLE OF AUTHORITIES

Page

CASES

TABLE OF AUTHORITIES – Continued

INTEREST OF *AMICI CURIAE*[1]

Robert R. Sachs[2] is a patent attorney, with seventeen years of experience prosecuting software patents for high technology software and companies in the Silicon Valley. Mr. Sachs has been responsible for crafting patent strategies for a wide range of software inventions, including search engines, user interfaces, databases, networking protocols, encryption, compression, video codecs, interactive television, predictive modeling systems, as well as inventions related to e-commerce, financial services, financial instruments, and financial analytics. Mr. Sachs is also an inventor on several patents. Mr. Sachs is a frequent writer and speaker on patent law issues pertaining to software and business methods. Finally, Mr. Sachs is the primary patent evaluator for a number of patent pools covering a variety of audio, video, interactive cable, telecommunications, and wide-area networking standards.

Daniel R. Brownstone is a patent attorney and inventor, working for the past decade with numerous

---

[1] No part of this Brief was authored in whole or in part by any of the parties. No monetary contributions to fund the preparation or submission of this Brief were made by anyone other than the *Amici*.

[2] In compliance with Sup. Ct. R. 37.2(a), timely notice of *Amici's* intent to file this Brief was given to counsel of record for both parties, and consent granted by both parties. This Brief is being filed with the consent of the parties. The parties have also filed with the Clerk of this Court general consents to the filing of *amicus* briefs.

companies in the software, communications and bioinformatics industries. Mr. Brownstone guides startups and public companies in developing and managing patent protection programs to identify and protect their core software and business technologies in this country and abroad. Mr. Brownstone is also an adjunct professor of law, and a frequent speaker on topics involving developments in patent law and practice.

By the nature of their concentration on software and financial clients,[3] *Amici* have deep experience with how inventors, patent examiners, and the courts approach the question of patent eligibility.

As students of the patent law and 35 U.S.C. § 101 in particular, *Amici's* interest in this case is two-fold: First, to illustrate the potential impact of the Federal Circuit's machine-or-transformation test on software-related inventions; and second, to provide a framework for analysis of patent eligibility that relies on the basic principles of patent law and a philosophically and scientifically sound approach to the nature of software innovations and software process claims.

[3] *Amici* submits this brief *pro se*, and the views expressed here are not necessarily those of their firm or clients.

INTRODUCTION AND SUMMARY OF ARGUMENT

This Court has long recognized that the patent eligibility statute, 35 U.S.C. § 101 ("§ 101"), should be read broadly, with limited exceptions. These limited exceptions include processes claiming laws of nature, scientific phenomena or abstract intellectual ideas. These exclusions apply to inventions and discoveries of the modern Information Age just as surely as they did to those of the Industrial Age. The challenge is to "establish rules that enable a conscientious patent lawyer to determine with a fair degree of accuracy which, if any, program-related inventions will be patentable." *Diamond v. Diehr*, 450 U.S. 175, 219 (1981) (Stevens, J., dissenting).

The Federal Circuit's machine-or-transformation test, derived from this Court's decisions, is certainly *a* test for identifying patent-eligible subject matter. But it cannot be, as the Federal Circuit held, the *only* test. Such a requirement is not only inconsistent with the precedent of this Court, but fails to recognize that a "one-test-fits-all" rule cannot be easily applied across the three categories of exclusion – laws of nature, scientific phenomena, and abstract ideas. Even within a single category like abstract ideas, a single rigid test is not easily conformed to the vast array of technologies from which innovation springs.

Software inventions are an important illustrative example of why the machine-or-transformation test should not be the definitive test for patent eligibility.

Software inventions have long been recognized as deserving of patent protection, and it is software innovation that drives much of the modern economy. But software, by its very nature, abstracts from the physical world, both in its design and operation. To the extent that there is a test for whether a software process claim is patent eligible, that test must be able to distinguish between claims for purely *abstract intellectual ideas*, which are not patentable, and claims that use *abstractions* to achieve results having meaningful "real world" applications, which are patent eligible.

Amici take no position on whether claim 1 of Bilski's application, or so called "business methods," are patent eligible. Rather, *Amici's* concern is that the machine-or-transformation test threatens to undermine patent protection for software. Even though the Federal Circuit declined "to adopt a broad exclusion over software," in the short time since their decision, the machine-or-transformation test has been applied to software inventions in contradictory and inconsistent manner at every level of the patent system: by patent examiners, by the Board of Patent Appeals and Interferences,[4] and by

[4] *Compare Ex parte Borenstein,* No. 2008-3475, 2009 WL 871128 (B.P.A.I. Mar. 30, 2009) (information stored in a database provides sufficient structure to meet machine prong); *Ex parte Greene,* No. 2008-4073, 2009 WL 1134839 (B.P.A.I. Apr. 24, 2009) (recitation of "vector processor" insufficient); *Ex parte Koo,* No. 2008-1344, 2008 WL 5054161 (B.P.A.I. Nov. 26, 2008)

(Continued on following page)

various District courts.[5] The Federal Circuit's new test disturbs the settled expectations of the software industry by calling into question the validity of tens of thousands of issued patents – protection that such inventions have enjoyed under this Court's broad understanding of patent-eligible subject matter.

The machine-or-transformation test should not be the only tool to test patent eligibility, because not every invention in every field will fit the same mold. Rather, the Court can reaffirm the law of patentable subject matter in its broad, open-ended form, recognizing the different technological contexts in which process claims appear. Applying a variety of balancing factors provides the flexibility necessary to leave "room for the revelations of the new, onrushing technology," *Gottschalk v. Benson*, 409 U.S. 63, 71 (1972).

(recitation of a "relational database management system" insufficient).

[5] *DealerTrack, Inc. v. Huber*, 2009 U.S. Dist. LEXIS 58125 (C.D. Cal. July 7, 2009); *Cybersource Corporation v. Retail Decisions, Inc.*, 2009 U.S. Dist. LEXIS 26056 (N.D. Cal. Mar. 27, 2009); *compare Versata Software, Inc. v. Sun Microsystems, Inc.,* 2009 WL 1084412 (E.D. Tex. Mar. 31, 2009).

ARGUMENT

I. THE RISK TO SOFTWARE INNOVATION FROM THE FEDERAL CIRCUIT'S MACHINE-OR-TRANSFORMATION TEST

The primary concern of this case is what appears to be an overly broad patent claim, potentially giving the patentee a greater exclusive right than he deserves. The Federal Circuit was attempting to determine whether Bernard Bilski's claim was nothing more than a claim for an "abstract intellectual concept." *In re Bilski*, 545 F.3d 943, 952 (Fed. Cir. 2008). That court attempted to find a single test that embodied this Court's prohibition against patents claiming laws of nature, natural phenomenon, and abstract ideas. But this Court has recognized the difficulty in resolving questions like this, and has cautioned against assuming that even its own multiple different approaches are dispositive: "We do not hold that no process patent could ever qualify if it did not meet the requirements of our prior precedents." *Gottschalk v. Benson,* 409 U.S. 63, 71 (1972); *see also Bilski*, 545 F.3d at 979 (Newman, J., dissenting) ("Nonetheless, the Federal Circuit now so holds").

The Federal Circuit unfortunately did not perceive the larger view that this Court has had. In answering the narrow question before it, the Federal Circuit turned this Court's identification of two possible *sufficient* conditions – that a process claim is patentable *if* it transforms its subject matter to a different state or thing or *if* it is machine implemented – into *necessary* conditions, that such a claim is

patentable *if and only if* it is either machine implemented or performs this transformation. The Federal Circuit took what this Court described in *Benson* as the "clue"[6] to patentability, *id.*, and turned it into a "definitive test." This legal legerdemain has been exhaustively analyzed by others, and that analysis will not be repeated here. *See In re Bilski*, 545 F.3d at 979 (Newman, J., dissenting); *id.* at 1012 (Rader, J., dissenting).

The machine-or-transformation test is not an incorrect test, but it is a limited one. If a process claim meets the machine-or-transformation test, then the claim is patent eligible. However, the converse is not true: if a claim fails the machine-or-transformation test, it should not be automatically deemed patent ineligible. Indeed, the test fails to encompass many software innovations that have traditionally been considered patent eligible, as illustrated below.

A. The Federal Circuit's Test Does Not Encompass Many Software Innovations

Software innovation covers such diverse fields as operating systems, memory management, computer programming languages, communications protocols, networking topologies, databases, information

[6] A clue is "the information or key that guides through an intricate procedure or a maze of difficulties" Webster's Third New International Dictionary (1986).

retrieval, graphics processing, video processing, color processing, fonts, animation, word processing, email applications, web browsers, navigation systems, and of course, graphical user interfaces. The U.S. Patent Classification System has over one thousand specific classes directed to software and computer related inventions. Any test for patent eligible process claims should take into account this diversity.

Significant problems in applying the machine-or-transformation test to software inventions arise from the transformation prong of the test. The Federal Circuit recognized that electronic signals and electronically-manipulated data are "the raw materials of many information-age processes" *In re Bilski*, 545 F.3d at 962, and stated that "transformation" could under at least some circumstances include transformation of data that is representative of physical and tangible objects. *Id.* at 962-63.[7]

But a narrow construction of the test to require that the data being transformed always represent macroscopic physical and tangible objects – objects that can be seen and touched – ignores the reality that many useful software inventions use and transform data that does not have clear tangible correlates. Rather, many software applications operate on

[7] In the court's example, the data was transformed into a visual depiction on a display. *Id.* The court did not clarify whether the visual depiction itself was required to meet its test, or whether transforming data representative of physical objects without subsequent display would have been sufficient.

data representing *information* instead of physical objects, and it is the transformation of that information that provides the ultimate practical application of the software – the practical application that this Court has identified time and again as a touchstone of patentability. Several examples illustrate this point.

1. Graphical User Interface Inventions

A "graphical user interface," or GUI, refers generally to the way in which people interact with modern personal computers and other electronic devices. The Microsoft Window XP and Linux operating systems are each examples of software that provide GUIs. Since the mid 1960s, the U.S. Patent and Trademark Office ("PTO") has granted over 22,000 patents in the GUI field.[8]

A GUI's underlying functionality is what makes it more than just pictures behind glass. Using a mouse to drag a file from one folder to another is a simple example of the relationship between a GUI's visual components and its underlying functionality. GUIs are typically protected with process claims that describe those functional operations. GUIs play a vital role in differentiating products in the

[8] GUIs are part of class 715 of the U.S. Patent Classification System. User interface inventions are also found in a variety of other classes.

marketplace, and patent protection for new and useful functionality provided by GUIs is thus important to software industry.

Under the machine-or-transformation test's transformation prong, data being transformed must be representative of physical objects. But it is unclear what data must be displayed in a GUI to be considered "representative," and which "physical tangible objects" have to be represented. For example, it is common now to describe files stored on a computer as being in different "folders." But clearly there are no physical folders inside a computer, nor could one locate a "trash can" or "recycle bin" on a hard drive. Each of these is an abstraction of the way computers store information; they are not themselves "physical tangible objects." Under the machine-or-transformation test, however, these abstractions may be fatal to patent eligibility.

An "abstraction" is not the same as an "abstract intellectual idea."[9] Software designers and programmers regularly use abstractions to represent the objects, functions and interactions that make up modern computing. The abstraction of a "folder" is useful to represent something that stores a number of distinct "files" or "documents." This does not, however, make a "folder" merely an abstract intellectual idea.

---

[9] *See* discussion *infra* Part II.C.

To be sure, a "document" shown on the screen can be printed and thus become a tangible piece of paper. But if a "document" is created electronically and never printed, it cannot represent a physical object, because the physical object does not and will never exist – and yet, surely, the document is more than an abstract intellectual idea.

Similarly, under the machine-or-transformation test, it is not clear what counts as a "transformation" of data in a user interface. To the user, being able to edit a document, construct a spreadsheet, design a web page, or even search the Internet for documents, all are useful operations that take a given state of the world, and upon user input, change that state in some degree, however small. The machine-or-transformation test imposes an unguided qualitative assessment of whether such a transformation is "good enough."

That process claims to user interfaces may be novel and useful, and yet excluded under the Federal Circuit's test, illustrates that test's unsuitability to software applications. Intended by the Federal Circuit to be applied generally to all technologies, it fails to appreciate the nuances of any.

2. Color Processing Inventions

Output devices, such as computer monitors, television screens, printers, and projectors cannot yet output (e.g., display, print, or project) the entire range of colors that humans can perceive. The portion of the

color space that a particular device can output is called its color gamut.

Digital images are represented by pixels, and each pixel has a set of data values that represent the "color" of the pixel. In many images, pixels have colors that are "outside" the gamut of the device the pixels are being shown on. To create the best possible output for a given image, it is necessary to determine how to change the original color values of the pixels into color values that the device can output. The particular process (algorithm[10]) selected for the transformation impacts the final quality of the output image. Solutions to this problem are useful in fields as diverse as medical imaging, digital cameras, and high-definition broadcasting of football games. There are over 500 patents that address various aspects of solving this problem.

A narrow construction of the machine-or-transformation test could find claims on innovative

---

[10] There is considerable case law about "algorithms." An algorithm is a specific sequence of operations or steps that accomplish a task. Not all algorithms are "mathematical algorithms," which are algorithms that are solely mathematical operations. *In re Freeman,* 573 F.2d 1237, 1246 (C.C.P.A. 1978) ("Because every process may be characterized as "a step-by-step procedure * * * for accomplishing some end," a refusal to recognize that Benson was concerned only with mathematical algorithms leads to the absurd view that the Court was reading the word "process" out of the statute.") (emphasis in original). Algorithms *per se* do not exist in nature, they are created by humans to achieve tasks.

color gamut mapping solutions unpatentable as neither tied to a particular machine, nor transformative of data representing physical objects. First, algorithms used to solve the color gamut problem can be executed on a variety of computers and processors, and thus need not be tied to a "particular machine" other than a general-purpose computer.

Second, *Bilski's* requirement that data represent physical tangible objects seems misplaced: the data here represents the "color" of pixels. The numerical representation of color is not what most people would consider "physical and tangible."[11]

Third, if the test requires a claim to recite that the resulting transformation is *displayed*, then this rule ignores two significant facts: that the algorithms can be used for conversion without display, and more importantly, for infringement purposes, that the entity that performs the color conversion may not be the entity that outputs the image. For example, a first company may perform a pre-processing service, which performs the appropriate conversion and stores it on disk. A second company later purchases the converted data and displays it. By requiring a claim to the conversion technology to include the displaying step, both parties described above would avoid any liability for infringement. *See BMC Resources, Inc. v.*

[11] Of course, scientifically, color in the sense of light exists as photons, and electromagnetic waves, which have measureable physical properties.

Paymentech, L.P., 498 F.3d 1373 (Fed. Cir. 2007) (no direct infringement where multiple parties individually performed only limited steps of the process claim).

3. Encryption and Compression Algorithms

Data compression algorithms are what make it possible to store 7,000 songs on a digital audio player, a full-length motion picture on a DVD, and 1,000 photographs on a memory card. Data compression algorithms generally operate by identifying statistical patterns that occur in the data being compressed. By replacing some of these patterns with shorter patterns, the original file is "transformed" into the compressed file, which uses up less space in memory or on disk.

Encryption algorithms secure everything from ATM machines transactions to military communications to Internet transactions, including e-commerce online banking, and securities trading. Encryption algorithms make data files unreadable by those without the means to undo the encryption.

Many of these compression and encryption algorithms are agnostic as to the "meaning" of the data they are transforming, and do not require that the data represent tangible objects. What the data represents is irrelevant: a simple compression algorithm will compress an image of an elephant, as well as the full text of James Joyce's *Ulysses*, or a

spreadsheet of random numbers. An encryption algorithm will secure the contents of a DVD just as effectively as it will protect a set of passwords. But under the Federal Circuit's rule, these algorithms would be patent eligible only if the underlying data represents physical tangible objects, and not where the data represents letters and numbers without a tangible analogue – even though the algorithm processes both sets of data in exactly the same manner. If, as here, the innovation is in the mechanism – or algorithm – for compressing or encrypting, then a test for eligibility that depends on what underlying data is being compressed or secured neither furthers the advancement of the technology, nor meaningfully tests whether the subject matter is directed only to an abstract idea.

With respect to the machine prong of the test, like the color management algorithms, these algorithms do not rely for their effectiveness on a specific physical device, but can be employed by general purpose computers. Decisions since *Bilski* have held that a general purpose computer is not sufficient to meet the machine prong of the Federal Circuit's test. *See, e.g.*, *DealerTrack, Inc. v. Huber*, 2009 U.S. Dist. LEXIS 58125, at *12-13 (C.D. Cal. July 7, 2009) ("Under *Bilski* and the recent decisions interpreting it, the central processor in this case cannot constitute a 'particular machine'").

Thus, there are many software innovations that are new, useful and deserving of patent protection,

but which would be excluded if the machine-or-transformation test were the sole basis for determining eligibility.

B. The Risk to Software Innovation from The Machine-or-Transformation Test

The above examples of software innovations are broadly representative of the wide range of software innovations in the following ways:

- They solve problems that only arise from human innovation in the first place: graphical user interfaces solve the problem of how to interact with a computer; color management solves the problem of how to output images on devices created by humans; encryption and compression solve the problem of how to store and secure digital data on human-created devices;
- They provide useful benefits even when the data they act on does not represent physical tangible objects;
- They do not require the visual depiction of data that represents physical objects in order to provide useful benefits.

Thus, if the sole test remains that a patent-eligible process must transform data representative of physical objects or be tied to a machine other than a general-purpose computer, many inventions already patented and many yet to come would be without

protection. The risk of invalidation of existing software disrupts the settled expectations of those inventors who pursued – and obtained – patent protection prior to the *Bilski* decision. To limit patentability so narrowly would discourage innovation in some of the fields most important to our modern economy.

II. A SYNTHETIC APPROACH TO SECTION 101 AND SOFTWARE PROCESS CLAIMS

A. Section 101 Is Not a Needle to Be Threaded

Section 101 implements the Constitutional invitation of Article I, Section 8, Clause 8, as an open call to all inventors to come forward with their discoveries in exchange for an exclusive right. Section 101 defines the scope of protection for what by definition is now unknown and cannot be foreseen – inventions that have not yet come to be.

While its limited exclusions turn away at the outset those claims to the scientific principles and abstract intellectual ideas that are free to all, the remainder must withstand the tests of novelty, non-obviousness, and enablement provided by other sections of the Act. It is there that overly broad claims are best identified and challenged.

Section 101 is too blunt an instrument with which to consistently differentiate between a patent claim that is properly broad, and one that is overly broad. A patent claim can be so broad that it reads on

the past, and hence is not novel.[12] Alternatively, a patent claim can be so broad that it reads on the unknown future, capturing something that the inventor has not yet invented – in which case the patentee has failed to fully enable his invention. The appropriate way to prevent these types of overly broad patents is application of the statutory requirements for novelty, non-obviousness, and enablement. Section 101 is also a poor tool for differentiation if only because humans cannot foresee – let alone judge well – what will be invented tomorrow. What we can do is make judgments based on evidence from the past and the present, and this is precisely what §§ 102, 103 and 112 allow us to do.

There will always be cases of patent applications seeking claims on subject matter that exists outside the reach of human ingenuity. How else will the boundary of knowledge and invention be advanced if it is not occasionally pushed? The *Benson* Court was sensitive to this conundrum, and made clear that it was leaving "room for the revelations of the new, onrushing technology." *Benson*, 409 U.S. at 71.

To balance these concerns – between overreaching rights of inventors and the need to keep the

[12] Indeed, *Benson* hinted at precisely that: "Here the 'process' claim is so abstract and sweeping *as to cover both known* and unknown uses of the BCD to pure-binary conversion." *Benson,* 409 U.S. at 69 (emphasis added).

door open to future innovations, this Court should maintain the framework that it has adopted over the past 100 years, using a mode of analysis based not on bright line rules and talismanic formulations, but guided by first principles of patent law.

B. A Single Test Cannot Be Used to Identify Unpatentable Claims to Laws of Nature, Natural Phenomena, and Abstract Intellectual Ideas

The difficulty of determining patent eligibility under § 101 and its predecessors is evidenced by the numerous doctrines that have been adopted by this Court, the Federal Circuit, and its predecessor, the Court of Customs and Patent Appeals, over the last 200 years. The multiple doctrines evince the need for a flexible framework that can accommodate the wide variety of different issues presented by patent claims at the boundary of ingenuity and the margins of invention.

Many courts have treated the exclusions from patent-eligible subject matter – laws of nature, natural phenomena, and abstract ideas – as one and the same, even though the case before them implicated only a single one of these categories. As Justice Breyer observed in *Lab. Corp. of Am. Holdings v. Metabolite Labs., Inc.,* these are three different categories, and difficult to define precisely. 548 U.S. 124, 134 (2006) (Breyer, J., joined by Stevens and Souter, JJ., dissenting from dismissal of writ of

certiorari). Logically, different categories of subject matter must be treated using different considerations. Indeed, apparatus claims are treated differently from process claims for purposes of § 101 analysis – so too must claims that implicate a law of nature, for example, be treated differently from those that implicate an abstract intellectual idea.

Even accepting Justice Breyer's caveat, some basic differences between the categories can be noted. "Law of nature" evokes a sense of immutable, universal, generalized rules of how the universe – or "Nature," or "reality" – works.[13] The laws and constants of physics, Maxwell's equations for electromagnetic fields, the speed of light, and pi are illustrative. By contrast, "natural phenomenon" connotes a naturally-occurring physical event or condition, not produced by humans. Lightning, tornadoes, the aurora borealis, the 17-year cycle of the cicada, solar eclipses and the "green flash" are all examples of natural phenomena. These two categories are quite different in kind: the former captures the eternal, the fundamental aspect of reality, while the latter captures events that occur in the physical world, and that are contingent upon the particulars of geography, biology, physics,

[13] A law of nature is "a generalized statement of natural processes; one of chief generalizations of science variously conceived as imposed upon nature by the Creator, as representing an intrinsic orderliness of nature or the necessary conformity of phenomena to reason and understanding." Webster's Third New International Dictionary (1986).

etc. Natural phenomena certainly comport with the laws of nature, but they are not themselves laws of nature.

The Federal Circuit incorrectly assumed that the three categories of exclusion could be treated as one, under the gloss of a "fundamental principle." *In re Bilski*, 545 F.3d at n.5 ("As used in this opinion, 'fundamental principle' means 'laws of nature, natural phenomena, and abstract ideas' "). That error was compounded by a misstatement of this Court's precedent. *Id.* at 954 ("The Supreme Court, however, has enunciated a definitive test to determine whether a process claim is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself").

The Federal Circuit's assumption ignores the very diverse manifestations of these categories and how they may arise in process claims. A test that perhaps captures a process claim for a law of nature itself – by identifying its inclusion of a fundamental construct of physics and nothing more – would hardly do well to identify a claim for nothing more than a transient event or a naturally occurring substance.

Similarly, an abstract idea is not inherently a law of nature or natural phenomenon, and so claims that seem too abstract must be treated with their own appropriate considerations. Of these three areas, the one that most directly impacts the patentability of software is the exclusion of abstract intellectual ideas.

C. Broad Software Process Claims Are Not Inherently Unpatentable Abstract Ideas

Claims by definition are *abstractions* and necessarily so. This has been long recognized:

> The difficulty which American courts . . . have had . . . goes back to the primitive thought that an "invention" upon which the patent gives protection is something tangible. The physical embodiment or disclosure, which, in itself is something tangible is confused with the definition or claim to the inventive novelty, and this definition or claim or monopoly, also sometimes called "invention" in one of that word's meanings is not something tangible, but is an abstraction. *Definitions are always abstractions*. This primitive confusion of "invention" in the sense of physical embodiment with "invention" in the sense of definition of the patentable amount of novelty, survives to the present day, not only in the courts, but among some of the examiners in the Patent Office.

E. Stringham, *Double Patenting,* Washington D.C., Pacot Publications (1933) (emphasis added).

There is a difference between an *abstract idea* and an *abstraction*: An *abstraction* is a generalization, it is a definition that identifies the principle aspects or features of the concept. The concept of "dog" is itself an abstraction from the details of any

specific dog. But a "dog" is clearly not an abstract *intellectual* idea.

Given that philosophers have debated the nature of abstract ideas for more than 2000 years – the problem of universals – it is not surprising that the Federal Circuit did not cut this Gordian Knot. Nor does *Amici* presume to set forth a definitive framework for analysis. Rather, *Amici* attempts to set forth a sufficient basis for aiding the Court in distinguishing between claims for abstract intellectual ideas and claims that are merely broad in scope.

All human language – and all patent claims – make use of concepts. Concepts such as "house," "dog," "red," and "father" are used to reference physical objects, their attributes, and relationships. We understand what the word dog means, because we generalize from our experience with individual dogs. In philosophical terms these concepts are called "concrete concepts." The other class of concepts involves abstract concepts. "Equality," "fairness," "justice," and "humility" are abstract intellectual ideas pertaining to humans and social relationships. Mathematics is one domain of abstract ideas, with prime numbers, groups and sets, and the Pythagorean Theorem, being simple examples. Thus, "dog" may be an abstraction, but it is certainly not an "intellectual" one. The concern of this Court has properly been on abstract *intellectual* ideas – ideas, the core meaning of which is not simply a generalization of the attributes of physical objects and experiences, but rather ideas

that do not have any physical representation in the world.

Software process claims – even when they do not recite real world-entities – are not claims to *abstract intellectual ideas*. Rather, software process claims describe an invention in the very same manner that computer programmers develop computer programs. Computer programs use abstractions to define the relevant characteristics and features of the data and the operations of the program. Software process claims also use abstractions – indeed sometimes the very same abstractions as in the computer program – to define the relevant steps of the invention. Just as a computer program itself will not be an abstract intellectual idea, so too a software process claim does not inherently describe an abstract intellectual idea.

There is a difference between a broad process claim and a claim that is for an abstract intellectual idea. A broad claim is acceptable, and may protect patentable subject matter, as long as it is definite and supported by the disclosure. Only where the subject matter of the claim *as a whole* is directed to intellectual ideas of the type described above would it fail to be patent eligible.

And this is the actual teaching of the historical patent law cases that were thought to invalidate claims as unpatentable subject matter. Instead, this Court's jurisprudence can be explained not as rejecting claims as nothing more than "abstract" *per se,* but rather as filtering out claims that were

abstract in the sense of being so overly broad that they were anticipated, indefinite or lacking enablement.

D. Selected Historical Cases: *Le Roy, O'Reilly, Corning,* and *Tilghman*

Several early Supreme Court cases that touch upon patent eligibility and are often cited in the modern case law are *Le Roy v. Tatham*, 55 U.S. 156 (1852); *O'Reilly v. Morse*, 56 U.S. 62 (1853); *Corning v. Burden*, 56 U.S. 252 (1853); and *Tilghman v. Proctor*, 102 U.S. 707 (1880). Although these cases touch on patenting abstract ideas, their actual holdings did not in fact turn on that issue.

Both *Benson* and *Diehr* cite *Le Roy* for the proposition that "A principle, in the abstract, is a fundamental truth; an original cause; a motive; and these cannot be patented, as no one can claim in either of them an exclusive right."[14] *Le Roy*, 55 U.S. at 175. The actual holding of *Le Roy* is that the trial

[14] This statement was dicta and expressly so, as the Court firmly stated that the issue of whether "The newly discovered principle, to wit, that lead could be forced, by extreme pressure, when in a set or solid state, to cohere and form a pipe, was not in the patent, and *the question whether it was or was not the subject of a patent was not in the case.*" *Le Roy,* 55 U.S. at 171 (emphasis added). The quoted language itself came from the trial court, which specifically admonished the jury that "The 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." *Id.* at 174.

court erred in instructing the jury that it was not to consider the novelty of Tatham's machinery when judging whether the patent was valid: "We think there was error in the above instruction, that the novelty of the combination of the machinery, specifically claimed by the patentees as their invention, was not a material fact for the jury." *Id.* at 177. The *Le Roy* Court did not hold Tatham's patent invalid for covering a "principle," rather it sent the patent back to the jury to determine whether the claim was invalid for want of novelty. Thus, the question of whether the patent claim was for an abstract principle did not arise in *Le Roy*.

The *O'Reilly* Court states that a patent on "principle" would be "void because the discovery of a principle in natural philosophy or *physical science,* is not patentable." 56 U.S. at 116 (emphasis added). The context makes clear that the use of "principle" is in reference to universal laws or "scientific principles," *id.* at 107, in other words, laws of nature, rather than "abstract ideas." The *O'Reilly* Court in fact decided the validity of the patent by what today would be understood as a failure of enablement under § 112:

> In fine, *he claims an exclusive right to use a manner and process which he has not described and indeed had not invented*, and therefore could not describe when he obtained his patent. The court is of opinion that the claim is too broad, and not warranted by law.

O'Reilly, 56 U.S. at 113 (emphasis added).

> Now in this case, *there is no description but one, of a process by which signs or letters may be printed at a distance*. . . . The words of the acts of Congress above quoted show that no patent can lawfully issue upon such a claim. For *he claims what he has not described in the manner required by law*.

Id. at 120 (emphasis added).

In contrast, in *Tilghman*, the process claim was patentable – yet the breadth of Tilghman's claim is astounding to the modern reader:

> I claim as of my invention, the manufacturing of fat acids and glycerine from fatty bodies by the action of water at a high temperature and pressure.

102 U.S. at 709.

This claim is no different in form from Morse's invalid claim in *O'Reilly*, as it describes the desired effect – manufacturing of fats and glycerine as compared to Morse's "making or printing intelligible characters" – by a generic mode of operation – the action of water at high temperature and pressure as compared to Morse's "motive power of the electric or galvanic current."

What saved Tilghman's claim from invalidity? The Court plainly explains: "It [Tilghman's process] is clearly pointed out in the specification, and one particular mode of applying it and carrying it into effect is described in detail. . . . The true construction of this claim is to be sought by comparing it, as have

already done, with the context of the specification; with the statement of the patentee." *Tilghman*, 102 U.S. at 729. Indeed, the Court quotes Tilghman's specification at considerable length to demonstrate the completeness of the disclosure. *See*, *id.* at 718-21.

In short, Morse's claim was not abstract, but overly broad – and invalid because he did not provide an enabling description commensurate with the breadth of the claim. Tilghman's equally broad claim was acceptable because he provided a specific description of the apparatus and mode of operation for his invention.

Finally, the problem in *Corning* was that that the claim was indefinite – not that it was abstract or for a natural principle: "It is true that the patentee, after describing his machine, has set forth his claim in rather ambiguous and equivocal terms which might be construed to mean either a process or machine." *Corning*, 56 U.S. at 269. The Court held that the trial court erred in instructing the jury that the claim was for a process. *Id.* at 270.

In sum, while many of this Court's early decisions discuss the problem of process claims, a careful reading of the cases suggests that they did not actually hold any process claim unpatentable for being directed to an abstract intellectual idea.

E. The Modern Cases: *Benson*, *Flook*, *Diehr* and the Preemption Doctrine

This Court's modern § 101 jurisprudence applies its earlier precedent to the modern computer age and the patentability of computer-related inventions. The cases evidence a similar concern with respect to overly broad claims, rather than claims to mere abstract ideas.

Benson was this Court's first assessment of patentable subject matter as applied to software and digital computers. It is also the first case to raise the issue of whether a claim would "pre-empt" all uses of a mathematical formula: "The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself." 409 U.S. at 71-72.

As this sentence makes absolutely clear, "pre-emption" is a conclusion, it is not an analysis itself. The preemption concern arose from the very specific problem before that Court, that "a scientific truth, or the mathematical expression of it, is not a patentable invention." *Id.* at 67 (citing *MacKay Radio & Telegraph Co. v. Radio Corp. of Am.*, 306 U.S. 86, 94 (1939)). For the purposes of its analysis, the *Benson* Court appears to have adopted Webster's very narrow definition of an algorithm: "A procedure for solving a given type of mathematical problem is known as an

'algorithm.'" 409 U.S. at 65. The Court correctly appreciated that claim 13, which recited purely mathematical steps, was in essence the "mathematical expression" of a "scientific truth," since the rules governing the conversion of BCD into binary are determined purely by number theory, not by human ingenuity. The Court's analysis thus equated Benson's particular "mathematical algorithm" with a "mathematical expression" of a scientific truth: "In *Benson*, . . . we concluded that such an algorithm, or mathematical formula, is like a law of nature, which cannot be the subject of a patent." *Diehr*, 450 U.S. at 186.

While that conclusion was appropriate given Benson's claim, its premise is not true as a general rule. Not all mathematical algorithms are "scientific truths" like the relationship between BCD and binary. Mathematical algorithms (or formulas) for routing of cell phone calls, scheduling airplanes or elevators, or ranking Internet search results are not "scientific truths."

The *Flook* decision also assumed, that like the algorithm for BCD to binary conversion, all mathematical algorithms expressed "scientific truths," and as such they are "one of the basic tools of scientific and technological work," *Parker v. Flook,* 437 U.S. 584, 591 (1987) (citing *Benson* 409 U.S. at 67) (internal quotation marks omitted). This then explains the basis of *Flook's* rule that the algorithm must be "treated as though it were a familiar part of the prior art." That rule is necessary only if all mathematical

algorithms in fact express scientific truths, i.e., something that was "true" before it was discovered, and thus part of the prior art. In his dissent in *Diehr*, Justice Stevens explained that the *Flook* Court's reasons for finding Flook's claim unpatentable were very basic: "The essence of the claimed discovery . . . was an algorithm that could be programmed on a digital computer." *Diehr*, 450 U.S. at 209 (Stevens, J., dissenting). Justice Stevens makes clear that using a digital computer to solve a problem is not the stuff of patentability, urging the majority to adopt "an unequivocal holding that no program-related invention is a patentable process under § 101 unless it makes a contribution to the art that is not dependent entirely on the utilization of a computer." *Id.* at 219. This view makes sense only on the assumption that mathematical algorithms implemented by computers always express scientific truths, an assumption that is not correct.

Flook's analysis need not be discarded however, because the decision is consistent with the fundamental concerns of *O'Reilly:* a failure of enablement and thus of the constitutional bargain. As in *O'Reilly*, Flook claimed more broadly than he enabled:

> The patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables. Nor does it purport to contain any disclosure relating to the chemical processes at work, the monitoring of process variables, or the means of setting off an alarm or adjusting an alarm system.

> All that it provides is a formula for computing an updated alarm limit.

Flook, 437 U.S. at 586.

The *Diehr* Court made this very same point:

> We were careful to note in Flook that the patent application did not purport to explain how the variables used in the formula were to be selected, nor did the application contain any disclosure relating to chemical processes at work or the means of setting off an alarm or adjusting the alarm limit. All the application provided was a "formula for computing an updated alarm limit."

Diehr, 450 U.S. at 193 n.14.

Thus *Flook*, too, lends support to a conclusion that software patent claims are not inherently abstract, but must be evaluated for their breadth in light of the disclosure.

Diehr also follows the preemption approach, but rather than dissecting the claim as in *Flook*, that Court considered the claim as a whole: "they do not seek to preempt the use of that equation. Rather, they seek only to foreclose from others the use of that equation in conjunction with all of the other steps in their claimed process" *Diehr,* 450 U.S. at 187. *Diehr* is consistent with the mode of analysis presented here, that the claim must be interpreted as a whole and with regard to what those of skill in the art would understand based upon the inventor's disclosure. Preemption is not itself an analysis of the claim, but a

conclusion to be drawn only after the claim is understood as a whole in view of the disclosure and knowledge of one of skill in the art.

F. A Multi-Factor Analysis for Section 101 and Software Process Claims

This Court has recognized in the obviousness context of § 103 that a "one-test-fits-all" approach to patentability is not appropriate. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398 (2007); *Graham v. Deere*, 383 U.S. 1 (1966). It stands to reason that a similarly flexible test would be appropriate when considering whether a claim is directed to excluded subject matter, given the diversity of technologies to which a claim may be related, and the differences in the types of exclusions under § 101.

A multi-factor approach to § 101 employs no one test to judge the patent eligibility of a process claim. Instead, the claim is evaluated by multiple balancing considerations, some of which may include:

- coverage of merely intellectual concepts versus coverage of concrete concepts and abstractions;
- preemption of applications of the claimed invention versus the available arena in which others can invent in the future;
- the breadth of the claim versus the definiteness and enablement provided by the specification;

- coverage of laws of nature versus coverage of applications of such laws for useful purposes; and
- coverage of natural phenomena per se versus the use of natural phenomena for particular ends.

As applied to software claims, these factors would likely lead to a conclusion that most are within the dominion of § 101. While a software process may have no practical application "except in connection with a digital computer," *Benson,* 409 U.S. 71, a modern analysis cannot stop there. Rather, the patent eligibility of a software process claim should be considered in view of the above factors.

First, even in the absence of explicit recitations of hardware, a software process claim describes the operations of a computer, as would be understood by anyone of skill in the art of computer engineering and programming. A computer cannot operate on "ideas," let alone "abstract" ones. The first balancing factor tilts heavily towards the conclusion that a process claim necessarily deals with concrete concepts and acceptable "abstractions."

Second, in the context of both the historical and modern cases, the concern over preemption as a *standalone* consideration is misplaced – rather preemption is a balancing factor. By definition a patent claim operates to "preempt" others from implementing a particular invention, for how else is the "exclusive right" obtained? It is not preemption per se

that is bad, but rather preemption that is unsupported by an equally broad and enabling disclosure of invention that makes clear the metes and bounds of the invention, and thereby improperly forecloses future innovations by others. On the other hand, a broad scope of preemption is justified by a clear and definite claim and broadly enabling patent disclosure. Thus, the horse of preemption must always be shackled to the cart of disclosure.

So long as a claim for a software process is described as operating on a computer, then a claim, even a broad one, is patent eligible. If someone invents a particular algorithm, and a process claim for that algorithm is limited to operation on a computer – either explicitly or implicitly – then the sheer breadth of the claim does not defeat patent eligibility. The breadth of the claim must be understood not in a vacuum, but rather entirely in the context of what one of skill in the art would understand *precisely* because it is one of skill in the art who must be enabled by the patent disclosure.

Once the difference between laws of nature, natural phenomenon, and abstract intellectual ideas is recognized, claims implicating the former two categories can be more precisely identified. Separate tests and considerations appropriate to these claims remain to be identified.

Finally, only if the claim is so broad as to cover every possible implementation – and this is a question of fact that requires more than cursory

analysis – then the claim likely would be unpatentable under § 102 or § 103 – but it would still be patentable subject matter under § 101. This is because such a claim would likely recite the desired result and without significant limitation as to the way of achieving that result. If the result is too broadly stated, then it is likely a result that has been obtained by others, and hence not novel.

CONCLUSION

Applying a variety of balancing factors, rather than a single, complicated test, provides the flexibility necessary to leave room for the unknown future of technology innovation, a future that begins with the first patent application filed tomorrow.

Respectfully submitted,

ROBERT R. SACHS
Pro Se
DANIEL R. BROWNSTONE
Pro Se
555 California Street
Suite 1200
San Francisco, CA 94104
(415) 875-2300

Amici Curiae