Washington University Law Review

Volume 90 | Issue 5

2013

Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents

Kevin Emerson Collins

Follow this and additional works at: http://digitalcommons.law.wustl.edu/lawreview

Part of the Intellectual Property Commons

Recommended Citation

Kevin Emerson Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 Wash. U. L. Rev. 1399 (2013).
Available at: http://digitalcommons.law.wustl.edu/lawreview/vol90/iss5/4

This work is brought to you free of charge by the Wash U Law Repository. To explore the repository, click here (http://digitalcommons.law.wustl.edu/). For more information, contact repository@wulaw.wustl.edu

PATENT LAW'S FUNCTIONALITY MALFUNCTION AND THE PROBLEM OF OVERBROAD, FUNCTIONAL SOFTWARE PATENTS

KEVIN EMERSON COLLINS*

ABSTRACT

Contemporary software patents are problematic because they are often overbroad. This Article offers a novel explanation of the root cause of this overbreadth. Patent law suffers from a functionality malfunction*: the conventional scope-curtailing doctrines of patent law break down and lose their ability to rein in overbroad claims whenever they are brought to bear on technologies, like software, in which inventions are purely functional entities.*

In addition to identifying the functionality malfunction in the software arts, this Article evaluates the merits of the most promising way of fixing it. Courts can identify algorithms as the metaphorical structure of software inventions and limit claim scope to particular algorithms for achieving a claimed function. However, framing algorithms as the metaphorical structure of software inventions cannot put the scope of software patents on par with the scope of patents in other arts. Most importantly, the recursive nature of algorithms and Gottschalk v. Benson *create to-date unappreciated problems.*

TABLE OF CONTENTS

I. INTRODUCTION 1400
II. THE INVENTION-STRUCTURE EQUATION 1407
A. *Structural and Functional Claim Limitations* 1408
B. *The Overbreadth of Functional Claims* 1411
1. *Reaching Beyond an Inventor's Contribution* 1414
2. *Reaching Toward Markets* 1417

---

* Professor of Law, Washington University School of Law. I thank the participants in the 2005 cyberprofs retreat hosted by the Berkman Center for Internet & Society and Berkeley Law for comments on a very early draft of this paper. I also thank my Spring 2012 Advanced Patent Law class at Washington University for a lively, extended, and productive discussion on the indefiniteness of means-plus-function limitations in software patents.

Wash U Law Repository

3. Plugging up Spillovers 1424
C. The Hidden, Unified Design Principle 1424
1. Patentable Subject Matter 1426
2. Functional Claim Limitations 1427
3. The Written Description Requirement 1430
D. The Limits of Enablement 1433
III. EXPLAINING SOFTWARE-PATENT OVERBREADTH 1440
A. Software Inventions Are Functional All the Way Down 1440
B. The Functionality Malfunction 1443
IV. ALGORITHMS: A VIABLE SOFTWARE-SPECIFIC PATCH? 1446
A. The Invention-Algorithm Equation 1447
B. The Federal Circuit's Tentative, Uncoordinated Steps 1450
1. Algorithms as Corresponding Structures 1451
2. Algorithms as Possessed Inventions 1455
3. Algorithms as Indicators of Particular Machines 1456
C. Are Algorithms a Patent Medicine? 1460
1. Uniformity 1461
2. Recursion 1463
3. Gottschalk v. Benson 1467
V. CONCLUSION 1470

I. INTRODUCTION

Software patents are overbroad. Compared to patents in other fields of endeavor, they routinely grant inventors rights that extend further beyond the technology that an inventor has actually invented and disclosed.[1] The

1. JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE: HOW JUDGES, BUREAUCRATS, AND LAWYERS PUT INNOVATORS AT RISK 199 (2008); DAN L. BURK & MARK A. LEMLEY, THE PATENT CRISIS AND HOW THE COURTS CAN SOLVE IT 159–60 (2009); Mark A. Lemley, *Software Patents and the Return of Functional Claiming* 2–4 (Stanford Pub. Law, Working Paper No. 2117302, 2012), *available at* http://ssrn.com/abstract=2117302; Brian J. Love, *Why Patentable Subject Matters for Software*, 81 GEO. WASH. L. REV. ARGUENDO 1 (2012); Richard R. Nelson, *Intellectual Property Protection for Cumulative Systems Technology*, 94 COLUM. L. REV. 2674 (1994). Overbreadth is not the only problematic aspect of software patents. *See* Lemley, *supra*, at 24–31 (surveying arguments that identify problematic aspects of software patents). Some empirical evidence suggests that software patents are not unusually likely to be "bad" patents that the PTO should not have issued. Stuart J.H. Graham & Saurabh Vishnubhakat, *Of Smart Phone Wars and Software Patents*, 27 J. ECON. PERSPECT. 67 (2013); David J. Kappos, *Investing in America's Future Through Innovation: How the Debate over the Smart Phone Patent Wars (Re)Raises Issues at the Foundation of Long-Term Incentive Systems*, 16 STAN. TECH. L. REV. 485 (2013). However, this evidence is irrelevant to the issue of overbreadth if, as this Article argues, software patents are overbroad because the patent doctrine that invalidates overbroad patents in most arts is ineffective in the software arts.

blame for the problem of software-patent overbreadth has at times been placed squarely on the shoulders of the Court of Appeals for the Federal Circuit. The Federal Circuit has dropped the ball, so the argument goes, in that it has failed to employ patent law's well-developed tools for curtailing permissible patent scope in the software arts.[2] Given this explanation of the problem, the solution seems self-evident and eminently achievable: the Federal Circuit must simply step up and use the tools that are already at its disposal to curb the scope of software patents.

This Article offers a different explanation for the overbreadth of contemporary software patents. The root of the problem is not that the Federal Circuit has failed to deploy in the software arts the well-developed doctrinal tools for curtailing patent scope that it already uses in other arts. To the contrary, the most important of these tools is not effective in the software arts. Software is intrinsically different from most other patentable subject matters in a way that matters. It lacks the metaphorical bolt onto which patent law's primary scope-restricting doctrinal tool can attach to gain purchase and ratchet in permissible claim scope. By demonstrating that the problem of software-patent overbreadth goes deeper than has previously been recognized, this Article does more than identify a new cause of a known problem. It also counsels greater skepticism toward the existence of a simple, judicially administered solution to the problem. The problems that plague software-patent scope cannot be remedied as easily as many suggest they can be, and the anomalously high costs of contemporary software patents are unlikely to ever be eliminated.

To explain why patent law's conventional scope-curtailing doctrines are ineffective in the software arts, this Article articulates and defends two premises. First, it identifies the principal mechanism through which patent doctrine allows judges and examiners to successfully curtail permissible claim scope in technological arts other than the software arts. There is a distinction between the *structure* and *function* of an invention—that is, a distinction between the physical form of an invention and the behaviors that an invention can exhibit, respectively. Building on this distinction, patent doctrine makes what this Article calls *the invention-structure equation*: it holds as an ontological matter that an invention "is" its

2. *See, e.g.*, Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?*, 17 BERKELEY TECH. L.J. 1155, 1204–05 (2002) (arguing that the Federal Circuit could rectify the problem of overbroad software claims by tweaking the enablement analysis); Lemley, *supra* note 1, at 4, 38–42 (arguing that courts can fix the problem of software overbreadth "in one fell swoop" by restricting functional software claims to algorithms for performing the claimed function under section 112(f)).

Wash U Law Repository

structure and that an invention's function is more peripherally only what an invention "does."[3] The invention-structure equation, in turn, gives rise to an effective tool for curtailing permissible claim scope. Patents that recite enough of the physical, structural properties of what an inventor has invented as claim limitations are valid because they are limited in scope to an inventor's invention. However, the flip side of the invention-structure equation is that functional claims—that is, claims that only recite the behaviors of an invention as limitations—are not valid. Construed literally, functional claims encompass all structures that are capable of performing the claimed behaviors, meaning that they reach beyond the structures conceived and disclosed by an inventor and thus beyond an inventor's invention. In sum, the invention-structure equation is a doctrinal tool that courts use to invalidate broad, functional claims and rein in permissible claim scope in most arts.

The second premise addresses the intrinsic nature of software as a technology: software inventions are pure functionality. Software is a powerful technology precisely because it has been engineered at a deep level to ensure that the specification of functional properties does not require the specification of any physical, structural properties. Software inventions therefore cannot be defined with reference to the physical, structural properties of either a tangible copy of the software on a storage medium or a computer programmed with the software. They can only be defined by their behavior or function. They are functional entities "all the way down" on all relevant levels of description.[4] It makes no sense to talk about the physical, structural properties of a software invention when identifying what a software inventor has invented.[5]

The juxtaposition of these two premises—namely, the invention-structure equation in patent law and the intrinsically functional nature of software as a technology—lays bare the root cause of the contemporary problem of software-patent overbreadth. Permissible claim scope is usually tethered to the structure of an invention, but purely functional

3. *In re* Swinehart, 439 F.2d 210, 212 (C.C.P.A. 1971).

4. STEPHEN HAWKING, A BRIEF HISTORY OF TIME 1 (updated & expanded 10th anniversary ed. 1998) (using an origin myth about a stack of turtles to raise the issue of infinite regress to find a ground for an argument).

5. Of course, every embodiment of a software program is a material entity. A software invention is functional all the way down in the sense that the properties that make a software program a material entity—that is, its physical, structural properties—are not relevant to the definition of a protectable software invention or the scope of the patent that a software inventor should obtain as an economic matter. It is only in this limited sense that software inventions are pure functionality: software is clearly a material entity, but the invention-structure equation cannot use its materiality as a post to which to tether permissible patent scope.

http://digitalcommons.law.wustl.edu/lawreview/vol90/iss5/4

technologies like software have no relevant structure to which claim scope can be tethered. The only way to fashion a viable claim to a software invention is to employ purely functional claim language, so the prohibition on functional claims that reins in claim breadth in other arts cannot be used to rein in claim breadth in the software arts, at least without eliminating effective patent protection for software altogether. This doctrinal failure is what this Article calls patent law's *functionality malfunction*: the invention-structure equation is ineffective whenever it is brought to bear on inventions, such as software, that are pure functionality. The invention-structure equation suggests that the three most important concepts to keep in mind when understanding how patent doctrine usually curtails patent overbreadth are "structure, structure, structure." In contrast, the three most important concepts to keep in mind to understand the nature of a software invention are "function, function, function." It is this mismatch that gives rise to the functionality malfunction and that leads to overbroad claims in the software arts.

Tracing the problem of software-patent overbreadth to the functionality malfunction is important in two ways. First, it breaks new ground by demonstrating that most of the conceptual heavy lifting required to identify the cause of software-patent overbreadth cannot be accomplished through greater scrutiny of software patents in isolation. Instead, what is needed is a better understanding of how patent law regulates permissible claim scope as a general matter in *other* arts. Conventional wisdom identifies enablement as the most effective tool for reining in overbroad patents.[6] This Article upends this conventional wisdom, arguing that the invention-structure equation is the most effective tool, at least when the source of the overbreadth is functional claim language.[7] In part, this inversion follows from understanding the shortcomings of the enablement doctrine as a means of curtailing the type of overbreadth created by functional claims.[8] In larger part, however, it follows from recognizing just how pervasive and fundamental the invention-structure equation is in contemporary patent law. The invention-structure equation undergirds at least three patent doctrines: the exclusion of claims to "principles" or "abstract ideas" from patentable subject matter under section 101,[9] the

6. *See infra* note 144.

7. Interestingly, although functional claiming is widely viewed as leading to overbroad patents, why functional claims are overbroad has not to date been explored. This Article therefore offers a first sketch of this analysis, which turns out to be more complex than one might initially expect. *See infra* Part II.B.

8. *See infra* Part II.D.

9. 35 U.S.C. § 101 (2006).

Wash U Law Repository

section 112(f) rules of means-plus-function claiming,[10] and the section 112(a) written description requirement.[11] These three doctrines are usually addressed within isolated analytical silos, perhaps because each has its own distinct linguistic formulation and statutory grounding. This compartmentalization has masked the importance of the invention-structure equation, allowing one of the fundamental design principles of patent law's regulation of permissible claim scope to pass below the radar of patent courts and commentators. (It is for this reason that this Article has to coin the name "invention-structure equation.") To understand what is wrong with the scope of software patents, it is initially necessary to reveal the hidden design principle that curtails permissible claim scope in other arts. Only then can the import of the principle's inefficacy in the software arts be fully appreciated. The peculiarities of the figure (patent protection in the software arts) come into clear focus only by focusing initially on the ground (patent protection in other arts).

Second, the functionality malfunction offers support for a new software-exceptionalist approach to patent protection. Arguments supporting exceptional treatment of software in the patent regime often rely on the economics of innovation in the software industry or examination difficulties at the PTO to demonstrate how software is different.[12] In contrast, the functionality malfunction suggests that it is the intrinsic, technical nature of software that makes software exceptional. The proliferation of overbroad claims in the software arts has not occurred because the Federal Circuit failed to port well-developed patent doctrine over to the software arts in an intelligent manner. The problem is rather that software is intrinsically different in a way that renders the well-developed doctrine ineffective.[13]

Of course, the functionality malfunction need not in and of itself doom software patents to a fate of anomalous overbreadth. A *sui generis*, software-specific solution could, in theory, bring the scope of software patents into line with the scope of patents in other technological arts. To

10. *Id.* § 112(f).

11. *Id.* § 112(a).

12. *See, e.g.*, BURK & LEMLEY, *supra* note 1, at 156–58; Glynn Lunney, *E-Obviousness*, 7 MICH. TELECOMM. & TECH. L. REV. 363 (2001).

13. It is possible that inventions in some other technologies in which patents are viewed as problematic—such as business methods, printed texts, mental processes, and perhaps even monoclonal antibodies—are also functional all the way down, or at least close thereto. If this is true, then the functionality malfunction also explains, in part at least, why patents in these areas are problematic. To offer a proof of the principle, this Article only addresses the functionality malfunction in the software arts.

borrow a term from the software arts, the functionality malfunction demonstrates that patent law needs to be updated with a software-specific "patch." Perhaps the most promising of such patches is to treat *algorithms* as the metaphorical structure of software inventions. In computer science, an algorithm specifies a step-by-step procedure that must be followed to perform any given functionally defined task.[14] Algorithms are still functional entities in that they, too, are composed of functionally defined steps. Yet, because algorithms describe a specific set of functions for achieving the general function recited as a claim limitation, courts can frame them as the metaphorical structure of a software invention. That is, courts can adopt an *invention-algorithm equation* in the software arts: they can root a protectable invention in the algorithms that an inventor actually employs to achieve a claimed function, and they can limit the scope of functional software claims to particular algorithms for achieving the claimed functions.

The idea of using algorithms to limit patent scope in the software arts is not new in either patent opinions or patent scholarship. The turn in the Court of Appeals for the Federal Circuit to algorithms to limit the scope of software claims has already been noted in cases involving means-plus-function limitations.[15] However, the full extent of the Federal Circuit's turn to algorithms has not to date been appreciated. The Federal Circuit has taken the first, tentative steps toward adopting an invention-algorithm equation not only in means-plus-function software cases but also in software cases addressing the patentable subject matter and written description doctrines.[16] Just as the unified nature of the invention-structure equation as a mechanism for curtailing permissible claim scope has remained hidden, the Federal Circuit's turn to algorithms in all three of the doctrines that employ the invention-algorithm equation in software cases has also gone unnoticed.

Yet, despite the Federal Circuit's recent, tentative embrace of algorithms, this Article argues that the invention-algorithm equation is far from a panacea for patent law's functionality malfunction in the software arts. In fact, the medicine (courts curtailing claim scope to an algorithm for achieving a functionally specified claim limitation) may turn out to be more harmful than the disease (routinely overbroad, functional software

14. *See infra* notes 212–15 and accompanying text.

15. *See, e.g.*, Stephen Winslow, *Means for Improving Modern Functional Patent Claiming*, 98 GEO. L.J. 1891 (2010); Sebastian Zimmeck, *Use of Functional Claim Elements for Patenting Computer Programs*, 12 J. HIGH TECH. L. 168 (2011); *see also* Lemley, *supra* note 1, at 38–43.

16. *See infra* Part IV.B.

Wash U Law Repository

claims). Most important, algorithms are infinitely recursive. An algorithm specifies a step-by-step procedure for achieving a functional goal, but each of the steps of the algorithm is in turn nothing more than a functionally specified goal. Once an algorithm has been identified for performing a task, a sub-algorithm can be identified for performing each of the steps of the algorithm.[17] In most arts, the physical, structural properties of an invention provide an intuitive bottom for the invention-structure equation. These structural properties are not functional properties at a more specific level of generality; they are qualitatively different types of properties. Once structural properties are recited as claim limitations, there is no need to dig deeper to find what the inventor has actually invented. In the software arts, however, there is no qualitative distinction between a "goal" and a step in the algorithm for achieving that goal or a "function" and a step in the process for implementing that function.[18] The distinction is only one of degree or height on a ladder of abstraction. To allow algorithms to play the scope-limiting role in software that physical structures play in other arts, the Federal Circuit cannot employ the invention-algorithm equation in a rote fashion. It cannot restrict the scope of every functional claim limitation to an algorithm for achieving that goal. This way forward leads down a rabbit hole to infinite regress and madness. Rather, the Federal Circuit must only curtail the reach of those functional claim limitations that are above a threshold level of generality. Whether identifying and administering this unified and normatively contingent threshold lies within the institutional competence of an Article III court is highly questionable.[19]

In sum, even though reliance on algorithms as metaphorical structure may represent the best hope for incremental, judicial reform of the problem of overbroad software claims, claim scope is likely to remain relatively less regulable in the software arts than in other technological arts. This Article cannot speak directly to the ultimate balance between the benefits of patent-induced incentives for innovation and the costs of patent litigation in the software arts.[20] Nor can it speak directly to whether categorically excluding software from patentable subject matter would increase innovation in the software and software-related industries.[21] More

17. *See infra* Part IV.C.2.

18. *But see* Lemley, *supra* note 1, at 3, 39 (building a proposal based on these distinctions).

19. In addition, the Supreme Court's prohibition on software claims to algorithms in the abstract throws a wrench into the invention-algorithm equation. *See infra* Part IV.C.3.

20. *Cf.* BESSEN & MEURER, *supra* note 1, at 95–146 (estimating the net cost or benefit of the patent protection generated by the contemporary patent regime in different industries).

21. James Bessen and Michael Meurer seek incremental reform of software patents, but they

narrowly, it only suggests that claim scope will always be more difficult to regulate in the software arts and that the costs of anomalous patent scope therefore need to be recognized as an entry that is permanently inked into the cost side of the ledger in the social-welfare analysis of software patents.[22] The open question is not whether such costs exist, but rather how good a patch for fixing the functionality malfunction in the software arts can be and thus how large the costs must be.

This Article proceeds in three substantive parts. Part II unveils and examines the invention-structure equation as a mechanism for curtailing permissible claim scope. Part III demonstrates that software inventions are functional entities all the way down, and it identifies the functionality malfunction that occurs when the invention-structure equation is brought to bear on purely functional technologies such as software. Part IV introduces the possibility of employing an invention-algorithm equation to fix the functionality malfunction in the software arts, and it addresses the difficulties that the Federal Circuit will eventually need to confront if it continues its initial, tentative steps down this path.

II. THE INVENTION-STRUCTURE EQUATION

To lay the groundwork for understanding what is unusual about the scope of contemporary software patents, this Part examines how patent law reins in the scope of functional claims in arts other than the software arts. Part II.A draws a distinction between structural and functional properties of a technology (and thus between structural and functional limitations in patent claims). Part II.B argues that claims with only functional limitations would, if permitted, be overbroad as an economic, normative matter. Part II.C introduces the invention-structure equation that patent law employs to avoid this potential overbreadth. It argues that three patent doctrines that have traditionally been analyzed within isolated silos are all manifestations of the same invention-structure equation. Part II.D turns to patent law's enablement doctrine and briefly explains why, contrary to conventional wisdom, enablement is not an effective, independent curb on the overbreadth of functional claims.

---

support the exclusion of software from patentable subject matter if those incremental reforms were to fail. *Id.* at 235–53. The functionality malfunction creates a pessimistic outlook for the success of incremental reform.

22. These costs may take the form of uncertainty, rather than overbreadth, if courts compensate for the functionality malfunction by invalidating overbroad software patents in an *ad hoc* manner. *See infra* note 210.

Wash U Law Repository

A. Structural and Functional Claim Limitations

Patent scope is the set of distinct technological things that constitute an inventor's patent interest,[23] and it is determined by the patent's claims.[24] In the contemporary patent regime, claims are short, textual descriptions of a set of technologies. They list the minimum group of properties that a technology must possess to be included in the claimed set.[25] It is well known that claim scope can be adjusted in either of two ways. First, claims grow narrower as the list of recited properties grows longer: the larger the number of distinct properties that a technology must possess to be included within the claim, the smaller the set of technologies that possess the full list of properties. For this reason, the properties recited in a claim are often called "limitations," as each additional property recited further limits the scope of the claim.[26] Second, assuming a constant number of limitations, any individual limitation can be a stronger or weaker curb on claim scope depending upon the generality of the language employed. A limitation can recite a targeted and specific property, leading to a relatively smaller set of described technologies and a narrower claim. Or, it can recite a sweepingly general property, leading to a relatively larger set of described technologies and a broader claim.[27]

---

23. For more nuanced definitions of patent scope, see Kevin Emerson Collins, *The Reach of Literal Claim Scope into After-Arising Technology: On Thing Construction and the Meaning of Meaning*, 41 CONN. L. REV. 493, 501–03 (2008) [hereinafter Collins, *The Reach of Literal Claim Scope*] (distinguishing thing-scope and meaning-scope).

24. 35 U.S.C. § 112, ¶ 2 (2006) (codifying the requirement for claims); *cf.* Giles S. Rich, *The Extent of the Protection and Interpretation of Claims-American Perspectives*, 21 INT'L REV. INDUS. PROP. & COPYRIGHT L. 497, 499 (1990) ("[T]he name of the game is the claim.").

25. Contemporary patent claims are often called "peripheral" claims because they attempt to put the public on notice *ex ante* of the outer boundaries or periphery of the claim. Peripheral claims can be juxtaposed with "central" claims that define patent interests with exemplars or prototypes of an invention. *See* Dan L. Burk & Mark A. Lemley, *Fence Posts or Sign Posts? Rethinking Patent Claim Construction*, 157 U. PA. L. REV. 1743 (2009); Jeanne C. Fromer, *Claiming Intellectual Property*, 76 U. CHI. L. REV. 719 (2009). Technically, only peripheral claims with an open transition list the minimum set of properties needed for inclusion in the claimed set. Norian Corp. v. Stryker Corp., 363 F.3d 1321 (Fed. Cir. 2004) (construing a closed transition). The doctrine of equivalents extends a patentee's rights beyond the literal scope of the claims, but, today, not by much. John R. Allison & Mark A. Lemley, *The (Unnoticed) Demise of the Doctrine of Equivalents*, 59 STAN. L. REV. 955 (2007); David L. Schwartz, *Explaining the Demise of the Doctrine of Equivalents*, 26 BERKELEY TECH. L.J. 1157 (2011).

26. Jeffrey A. Lefstin, *The Formal Structure of Patent Law and the Limits of Enablement*, 23 BERKELEY TECH. L.J. 1141, 1145 (2008) (describing the mechanics of limitations).

27. Claim construction often hinges on disputes over the generality of limitations. *See, e.g.*, Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005) (en banc) (addressing the generality of the meaning of the claim term "baffle").

The conventional, two-dimensional understanding of how to modulate claim scope is technically accurate, but it is incomplete. It omits an important distinction between two qualitatively distinct types of limitations. Descriptive language, and thus claim limitations, can refer to a technology in either of two ways: it can point out either the structural or the functional properties of a technology.[28] The structural properties of a technology include its physical, spatial, and chemical properties. For example, *being eleven inches long* is a structural property of a standard sheet of copy paper, *being round* is a structural property of a bicycle wheel, and *being made of water* is a structural property of an ice cube. In contrast, the functional properties of a technology are what an invention can do and the roles that an invention can play in a larger system. They include the behavioral capacities that a technology possesses and the tasks that a technology is capable of achieving.[29] For example, *being flexible* (i.e., *being capable of bending*) is a functional property of a rubber band, and *being capable of curing a disease* is a functional property of a drug.[30]

The structural / functional distinction is not between objects that are either functional or structural. Any token or concrete, particular instance of a technological object always possesses both structural and functional properties. It is therefore possible to describe any given instance of a technology with a claim that has either structural or functional

---

28. *See* Peter Kroes, *Technological Explanations: The Relation Between Structure and Function of Technological Objects*, 3 PHIL. & TECH. 18, 18 (1998) (discussing "two different modes of description, viz., a *structural* and a *functional* mode of description" for technological objects).

29. The philosophical literature on functional explanation draws a distinction between the broader set of behavioral capacities or dispositions that an object possesses and the narrower set of functions of an object, the latter being informed in part by what society understands the primary purpose of an object to be in any given system. Richard N. Manning, *Functional Explanation*, *in* 3 ROUTLEDGE ENCYCLOPEDIA OF PHILOSOPHY 802, 802–03 (Edward Craig ed., 1998). According to this distinction, a heart has the function of pumping blood, but it has the mere behavior of making a noise. *Id.* In patent law, the term "functional property" is used in a broad sense that is interchangeable with the philosophical concept of a behavioral capacity or a disposition. Robert Cummins, *Functional Analysis*, 72 J. PHIL. 741, 758 (1975) ("To attribute a disposition *d* to an object *a* is to assert that the behavior of *a* is subject to (exhibits or would exhibit) a certain law-like regularity: to say *a* has *d* is to say that *a* would manifest *d* (shatter, dissolve) were any of a certain range of events to occur (*a* is put in water, *a* is struck sharply).").

30. Some descriptors that at first glance may appear to be structural in fact "define [things in part] in terms of use or effect. For example, a 'door' is something used to close and open a passageway; a 'nail' is an object used to hold two pieces of material together; a 'black' material is one incapable of reflecting visible light." *In re* Swinehart, 439 F.2d 210, 215 (C.C.P.A. 1971) (Lane, J., concurring). Nonetheless, the structural / functional distinction offers a viable classification system in most instances. The use of the structural / functional distinction in areas of law other than patent law suggests that the distinction is relatively stable. For example, building codes rely on a parallel distinction when they discuss "design" standards (that specify structure) and "performance" standards (that specify behavioral capacity). *See, e.g.*, J.L. Simmons Co. v. United States, 412 F.2d 1360, 1362 (Ct. Cl. 1969).

Wash U Law Repository

limitations.[31] For example, the coffee sleeve attached to the carry-out cup of coffee on my desk could be described as either "a cardboard band encircling a paper cup" (a purely structural limitation or description of the technology) or "a device that provides at least a ten degree temperature reduction between a hot liquid and an outer graspable surface" (a largely functional limitation or description of the technology).

The structural and functional properties of an object are interrelated. According to the materialist worldview that predominates today, a technology possesses the functional properties that it does only because it possesses its structural properties.[32] That is, there is a one-way dependence of causality from structure to function, and the structural properties of a technology are what make the technology capable of exhibiting the behaviors that it does. For this reason, the structural properties of a technology are commonly viewed as an answer to the "how" question of technology: "how [a] system will be able to perform the required function" requires "an explanation . . . in terms of the physical structure of that [system]."[33] Yet, despite the causal dependence of function on structure, there is no one-to-one correspondence between structural and functional properties. Any given functional property may be caused by an array of distinct structural properties.[34] A bicycle chain is flexible because it has one set of structural properties, whereas an elastic band is flexible because it has a different set of structural properties. Inversely, any given structural property may allow a technology to exhibit multiple, distinct behaviors.[35] If something possesses the structural property *being made of sugar*, it possesses both the functional properties *being capable of being a food source for a bacterium* and *being capable of dissolving in water*.

Although any given object can be referred to by naming either its structural properties or its functional properties, the choice between structural and functional limitations in a claim has a significant implication for a patentee's rights: it determines claim scope. If a claim specifies only the structural attributes of a technology, the plain language

31. At a given point in time, only a functional or structural description may be possible because the discovery of one type of property may predates the discovery of the other.

32. *See generally* MATERIALISM AND THE MIND-BODY PROBLEM (David M. Rosenthal ed., 2d ed. 2000) (collecting significant historical and contemporary essays on materialism).

33. Kroes, *supra* note 28, at 20–21.

34. Wesley C. Salmon, *Four Decades of Scientific Explanation*, *in* SCIENTIFIC EXPLANATION 30 (Philip Kitcher & Wesley C. Salmon eds., 1989) (discussing "functional equivalents"); Kroes, *supra* note 28, at 22 (noting that "physical structure is [not] logically implied by functional requirements").

35. Kroes, *supra* note 28, at 23 (noting that "the same structure may perform many different kinds of functions" and thus that "[a] one-to-one relation between structure and function is not guaranteed").

of the claim encompasses any technology that possesses those attributes, regardless of the use, purpose, or function to which the technology is eventually put by a user.[36] For example, a claim to a chemical entity with only limitations referencing the entity's molecular structure will encompass molecules that have the specified structure, regardless of whether the molecules are being used as a combustible fuel, a lubricant, or a nutrition supplement. Inversely, the plain meaning of a claim to a chemical entity with only functional limitations will encompass any compound that possesses the requisite behavioral capacities, regardless of the molecular structure of the entity. Compounds with radically different structures all fall within the scope of a functional product claim construed according to its plain meaning, so long as the different structures all give rise to behavioral capacities that satisfy the claim's functional limitations.[37]

B. The Overbreadth of Functional Claims

A social welfare analysis of optimal claim scope involves a classic balancing analysis. All other things being equal, inventors prefer broader claims and their potential to generate larger private benefits.[38] In turn, larger private benefits can lead to greater social benefits in the form of a stronger incentive to innovate.[39] However, the inseparable flip side of broader claims is that they generate an array of larger social costs, too. Static costs increase as higher prices push more end consumers out of the market.[40] Greater dynamic costs follow from a more pronounced slow-down in future innovation.[41] To the extent that broader claims lead to more overlapping patent interests, broader claims can also generate larger

36. Claims with only structural limitations do not recite the use to which a technology is put as a limitation, so the use is an additional element in the allegedly infringing device that is irrelevant to literal infringement. *See* Stiftung v. Renishaw PLC, 945 F.2d 1173, 1178 (Fed. Cir. 1991) (noting that "comprising" claims read on devices with additional elements).

37. Of course, if the meaning of a claim is determined using a special rule that does not comport with the normal rules of claim construction, it could be much narrower. *See infra* text accompanying notes 123–25 (presenting the rules of means-plus-function claiming).

38. *See infra* notes 67–70 and accompanying text (discussing the relationship between claim scope and market power). The most important "other thing" that might not be equal is validity: broader claims are more likely to be invalid, and inventors presumably prefer valid claims.

39. STAFF OF S. COMM. ON PATENTS, TRADEMARKS, AND COPYRIGHTS, 85TH CONG., AN ECONOMIC REVIEW OF THE PATENT SYSTEM (Comm. Print 1958) (prepared by Fritz Machlup) (presenting a historical overview of several justifications of the patent regime, including the incentive-to-invent justification).

40. SUZANNE SCOTCHMER, INNOVATION AND INCENTIVES 36–37 (2004).

41. *Id.* at 125–59.

Wash U Law Repository

administrative costs as acquiring the patent rights needed to undertake a research agenda or manufacture a product becomes, at best, more costly and, at worst, not feasible.[42] These costs mean that additional increments of claim scope are likely to generate a net social cost at some point in the gradual expansion of claim scope.

To curtail claim scope, patent law employs a commensurability principle: the scope of an inventor's claim should remain proportional to the contribution that an inventor makes to technological progress.[43] More precisely, the commensurability principle mandates that the set of things within the scope of a claim must track the set of things that embody an inventor's innovative ideas.[44]

If they were to be sanctioned, purely functional claims would frequently grant inventors excessively broad claims and violate the commensurability principle.[45] This section offers three arguments to defend this assertion and demonstrate that functional claims are particularly problematic from a social welfare perspective.[46] First,

---

42. Michael A. Heller & Rebecca S. Eisenberg, *Can Patents Deter Innovation? The Anticommons in Biomedical Research*, 280 SCI. 698 (1998); Carl Shapiro, *Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting*, *in* 1 INNOVATION POLICY & THE ECONOMY 119 (Adam B. Jaffe et al. eds., 2001).

43. For a wide-ranging discussion about the value of proportionality in intellectual property that goes far beyond the targeted arguments about the commensurability of technological contribution and claim scope made here, see ROBERT P. MERGES, JUSTIFYING INTELLECTUAL PROPERTY 159–92 (2011). Some justifications of patent protection do not rely on the commensurability principle. For example, prospect theory—and the social benefit of coordinated, less wasteful follow-on improvement that it highlights—could be invoked to defend functional claims even if they are not commensurate with an inventor's contribution. *See* Edmund W. Kitch, *The Nature and Function of the Patent System*, 20 J.L. & ECON. 265 (1977). The importance of prospect theory has been questioned by many scholars. *See, e.g.*, Lemley, *supra* note 1, at 54 n.175; Mark A. Lemley, *Ex Ante Versus Ex Post Justifications for Intellectual Property*, 71 U. CHI. L. REV. 129 (2004); Robert P. Merges & Richard R. Nelson, *On the Complex Economics of Patent Scope*, 90 COLUM. L. REV. 839, 868–908 (1990). Furthermore, even if prospect theory can be used to justify patent rights, it is far from clear that identifying behavioral capacities represents the best way of divvying up innovation into distinct prospects.

44. Kevin Emerson Collins, *Getting into the "Spirit" of Innovative Things: Looking to Complementary and Substitute Properties to Shape Patent Protection for Improvements*, 26 BERKELEY TECH. L.J. 1217, 1273–79 (2011) [hereinafter Collins, *Getting into the "Spirit"*]. Patent rewards and the social value of inventions are self-aligning in the sense that patent rewards are determined in part by market demand for patented inventions. Giles S. Rich, *Principles of Patentability*, 28 GEO. WASH. L. REV. 393, 402 (1960) ("That is one of the beauties of the patent system. The reward is measured automatically by the popularity of the contribution."). However, the market aligns the private rewards of patents and the social values of inventions only if one assumes that patent doctrine has a good system for tracking the set of things that embody an inventor's ideas. *See* Tun-Jen Chiang, *The Levels of Abstraction Problem in Patent Law*, 105 NW. U. L. REV. 1097, 1101 (2011) (framing this tracking problem as a levels-of-generality problem).

45. This section assumes that the functional claim limitations recite a previously unrealized technological behavior and that the claim is novel and nonobvious in relation to the prior art.

46. Surprisingly, despite the well-known nature of the prohibition on functional claims in patent

functional claims are unusually likely to encompass later-developed technologies that owe no debt at all to the contribution to technological progress made by a patentee. Second, they are unusually likely to reach toward markets and give patent owners significant market power.[47] Third, their absence need never deprive a deserving inventor of patent protection. A prohibition on functional claims only trims back the scope of protection at the margin, increasing the public "spillover" of patent rights.[48] In sum, the overbreadth problem lies in a combination of the nature of the after-arising technology that such claims encompass and the availability of robust patent protection for meritorious inventions without resort to such claims.

Before addressing these arguments in detail, it is important to dismiss one line in the sand that is commonly, but incorrectly, drawn to identify a claim whose scope exceeds an inventor's contribution. It is commonly argued that a claim by definition reaches beyond an inventor's contribution if it literally encompasses after-arising or later-developed technology.[49] If this bright-line rule were correct, then functional claims would be *per se* overbroad. However, the commensurability principle should not be interpreted to forbid literal claim scope from growing over time to encompass after-arising technologies that inventors did not themselves make available to the public.[50] Innovation is often cumulative: the later-developed embodiment can be an improvement that results from the later inventor employing the earlier inventor's contribution as a

law, no comprehensive normative defense of this prohibition has to date been articulated. The arguments presented here represent a first sketch of a defense.

47. Another concern distinct from overbreadth is that functional claims are inherently more vague than structurally defined claims. While this is true under limited conditions, *see infra* note 245, those conditions are not common in the software arts.

48. For a wide-ranging discussion of the value of spillovers in intellectual property law, see Brett M. Frischmann & Mark A. Lemley, *Spillovers*, 107 COLUM. L. REV. 257 (2007).

49. *See, e.g.*, IPPV Enters. v. Echostar Commc'ns Corp., 106 F. Supp. 595, 606 (D. Del. 2000) ("[L]ater developed technologies may not fall within the literal scope of the patent."); Charles W. Adams, *Allocating Patent Rights Between Earlier and Later Inventions*, 54 ST. LOUIS U. L.J. 55 (2009); Christopher A. Cotropia, *"After-Arising" Technologies and Tailoring Patent Scope*, 61 N.Y.U. ANN. SURV. AM. L. 151 (2005); Timothy R. Holbrook, *Equivalency and Patent Law's Possession Paradox*, 23 HARV. J.L. & TECH. 1, 37–40 (2009); Timothy R. Holbrook, *Possession in Patent Law*, 59 SMU L. REV. 123 (2006).

50. Collins, *Getting into the "Spirit," supra* note 44, at 1260–68, 1296–1302 (discussing overlooked, "easy" improvement cases); *cf.* Kevin Emerson Collins, *Enabling After-Arising Technology*, 34 J. CORP. L. 1083, 1098–1124 (2009) [hereinafter Collins, *Enabling*] (identifying three conditions under which literal claims routinely encompass after-arising technology); Collins, *The Reach of Literal Claim Scope*, *supra* note 23, at 536–53 (explaining how the meaning of claim language can remain fixed even as the set of things encompassed within a claim grows over time).

Wash U Law Repository

platform.[51] In many improvement cases, the earlier inventor's ideas are still embodied—perhaps in full—in the later-developed technology, despite the fact that the later inventor's ideas are also embodied in the technology.[52] In these cases, commensurability between contribution and claim scope requires allowing the earlier inventor's claim to encompass the later-developed improvement.

1. Reaching Beyond an Inventor's Contribution

Assume that an earlier inventor is the first person to reduce to practice a device that is capable of performing a function, that this inventor obtains a functional claim to his invention, and that a later inventor invents a device that is also capable of performing the claimed function but that is structurally distinct from the device reduced to practice by the earlier inventor. In this situation, there are two possible relationships between the earlier and later inventors. First, the later inventor may owe no intellectual debt at all to the earlier inventor, making the normative argument that the functional claim over rewards the earlier inventor an easy one to mount. Second, the later inventor may owe an intellectual debt to the earlier inventor, making the normative defense of a prohibition on functional claims more complicated.

First, functional claims can reach entirely beyond an inventor's contribution and into after-arising technology that owes no debt at all to the inventor.[53] In order to obtain a functional claim, an inventor does not need to prove that he was the first person to imagine or think up the

---

51. Suzanne Scotchmer, *Standing on the Shoulders of Giants: Cumulative Research and the Patent Law*, 5 J. ECON. PERSP. 29 (1991).

52. The counterintuitive notion that an improvement fully embodies the contributions made by both an earlier inventor and a later inventor becomes easy to see if inventions are viewed as sets of properties of things, rather than as things or ideas. Collins, *Getting into the "Spirit," supra* note 44, at 1235–42, 1255–73.

53. When determining if a later inventor owes a debt to an earlier inventor, both actual and constructive debts must be considered. Patent infringement does not require proof of copying, and there is no independent invention defense in patent law. DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc., 567 F.3d 1314, 1328–30 (Fed. Cir. 2009). Therefore, someone who infringes a patent by making the exact technology disclosed in the patent's specification may be an independent inventor who did not actually learn anything from the patentee. However, the infringer is charged with a constructive debt to the patentee under the commensurability principle: had he known about and read the earlier patent, then he would have learned how to make and use the disclosed technology. Similarly, in determining whether the later inventor of an after-arising technology owes a debt to an earlier inventor and patentee, it does not matter whether the later inventor actually read the earlier-issued patent. He owes a constructive debt to the earlier patentee if, after reading the patent, he would have learned information that would have made his later invention easier to achieve.

behavioral capacities that define claim scope.[54] A claim defined by a long-desired behavioral capacity is novel and nonobvious in relation to the prior art even if an inventor is merely the first person to reduce to practice or enable a technology that possesses the behavioral capacity.[55] Here, the inventor's contribution is only one means of achieving the behavioral capacity. Yet, a functional claim would encompass all technologies that possess the behavioral capacity, including those later-developed technologies that employ entirely unrelated means for achieving it and that therefore owe no debt at all to the contribution of the earlier inventor and patentee. The patentee's contribution is only one answer to the "how" question of technology, and the inventor of the later-developed technology offers an unrelated answer to that question.[56]

Consider a historical example. In *O'Reilly v. Morse*, Samuel Morse's broad, functional claim to the telegraph infamously described any device that used "the motive power of the electric or galvanic current . . . for marking or printing intelligible characters, signs, or letters, at any distances."[57] It would be difficult to fathom that Morse himself was the first to think up the idea of a technology that possessed the claimed behavioral capacity or to recognize the value of a device with the claimed functional property.[58] Rather, the functional claim limitations describe a long-felt consumer need. Morse's contribution to technological progress was being the first inventor to successfully achieve the difficult task of reducing to practice a telegraph machine that exhibited the claimed behavioral capacity. In this situation, a subsequent inventor may develop a different means of achieving the claimed behavior that is independent of Morse's entire contribution to technological progress. The later inventor

54. In plain English, one might say that an inventor did not "conceive of" or "invent" the behavioral capacity. However, these terms are often used in patent discourse in a way that is wound up with the structural properties of things. *See infra* note 174. This Article therefore describes inventors as "imagining" or "thinking up" behavioral capacities.

55. The articulation of a long-desired property in a prior art publication does not anticipate a claim to a machine that possesses that property. A publication anticipates a claim only if it enables an embodiment within the scope of the claim. Paperless Accounting, Inc. v. Bay Area Rapid Transit Sys., 804 F.2d 659, 665 (Fed. Cir. 1986) (discussing anticipatory enablement).

56. *See supra* note 33 and accompanying text (defining the "how" question).

57. O'Reilly v. Morse, 56 U.S. (15 How.) 62, 112 (1853). The Supreme Court invalidated this claim due to overbreadth. *See infra* text accompanying notes 106–08.

58. *O'Reilly*, 56 U.S. at 107 (noting that the idea of the telegraph was a "conviction . . . general among men of science everywhere" at the time of Morse's invention); *see also* ROBERT PATRICK MERGES & JOHN FITZGERALD DUFFY, PATENT LAW AND POLICY: CASES AND MATERIALS 88–90 (4th ed. 2007).

Wash U Law Repository

may owe no debt at all to Morse's contribution to technological progress,[59] and he may not have benefited from standing on the shoulders of the earlier inventor. The later inventor would be in an identical position to generate the after-arising technology even if Morse had never made his invention and Morse's patent disclosure had never been publicized.

Second, functional claims granted to an earlier inventor can sometimes encompass after-arising technology that does owe a debt to the earlier inventor. Here, the earlier inventor is not simply the first person to reduce to practice a device that exhibits the claimed behavior. Rather, he is more of a pioneer in that he is the first person to imagine the claimed behavior itself. In this situation, there is concededly a stronger argument under the commensurability principle in favor of upholding a functional claim. The very idea of a technology that exhibits the claimed behavior can legitimately be said to be part of the earlier inventor's contribution to technological progress, and any technology that exhibits that behavior can be said to embody the earlier inventor's idea. Here, a later inventor may owe an important debt to the earlier inventor even if the structural properties of a later-developed device are wholly independent of the structural properties of the device invented and disclosed by the earlier inventor.

As a contrast to Morse's functional claim, consider the functional claim at issue in *Halliburton Oil Well Cementing Co. v. Walker*.[60] An inventor made an improvement in a device for recording the depth of an oil well.[61] More specifically, he realized that the echoes of sound waves that have bounced off the shoulders of pipes at pipe junctions could be a valuable calibration measurement to figure out just how fast sound was traveling in the well. The inventor sought, and received, a functional claim to a "means . . . for tuning [an echo-recording device] to the frequency of echoes [from shoulders of pipes] . . . to clearly distinguish the echoes . . . from each other."[62] In *Halliburton*, the inventor was the person who thought up the idea of a technology that possessed the claimed behavioral capacity. Devices for distinguishing the echoes from the shoulders of pipes in oil wells were not long-felt consumer needs.

Yet, even in this *Halliburton* situation that is most favorable to the inventor, functional patent claims are overbroad in relation to an

59. More precisely, the later inventor may not even owe a constructive debt to Morse. *See supra* note 53.

60. 329 U.S. 1 (1946).

61. *Id.* at 7–8.

62. *Id.* at 10.

inventor's contribution.[63] The argument to defend the invalidity of the *Halliburton* claim is more complex than the argument required to defend the invalidity of the *O'Reilly* claim: a behavioral capacity is not the type of contribution to technological progress to which the scope of a patent claim must remain commensurate. The following two subsections provide support for this argument.

2. *Reaching Toward Markets*

Markets are defined by consumer purchasing behavior: they are categories of goods from which consumers will not switch away upon a small, non-transitory increase in price.[64] This sticky purchasing behavior develops only if goods within a market are sufficiently close economic substitutes for other goods within the market but not for goods outside of the market.[65] In turn, market power is the ability of a producer to raise price in a non-transitory fashion without consumer defection.[66] In a perfectly competitive market, no producer has market power. Identical goods produced by two different producers are presumptively perfect substitutes for a consumer, and any price increase by a producer causes consumers to switch away from that producer's good to an identical good made by a different producer. In order for a producer to obtain market

63. The Supreme Court invalidated the *Halliburton* claim. *See infra* text accompanying notes 120–22. However, there are some historical cases in which inventors have thought up a behavioral capacity and courts have upheld largely functional claims. *See* MERGES & DUFFY, *supra* note 58, at 96 (arguing that Alexander Graham Bell's broad functional claim was commensurate with his technological contribution because Bell thought up the claimed behavioral capacity).

64. PHILLIP E. AREEDA & HERBERT HOVENKAMP, 2A ANTITRUST LAW 179–246 (2d ed. 2002). The traditional antitrust definition of a market is binary, but the relevant question in intellectual property is arguably about the degree of substitutability on a continuum. *See* Louis Kaplow, *Why (Ever) Define Markets?*, 124 HARV. L. REV. 437, 506–08 (2010).

65. In the everyday sense of the word, "substitutes" are goods that can replace or fill in for each other because they satisfy the same consumer need. THE AMERICAN HERITAGE COLLEGE DICTIONARY 1354 (3d ed. 2000) (defining a substitute as "one that takes the place of another; a replacement"). Nails and industrial strength glue for bonding wood are substitute goods in this common-sense way: I use either one, but probably not both, to join pieces of wood. To determine whether goods are substitutes as a technical, economic matter, economists measure the goods' cross-price elasticity of demand. Two goods are substitutes if a decrease in the price of one good results in a decrease in demand of the other good and, inversely, an increase in the price of one good results in an increase in demand for the other good. ROBERT S. PINDYCK & DANIEL L. RUBINFELD, MICROECONOMICS 36 (7th ed. 2008). This technical definition usually maps onto the common-sense definition of a substitute. If consumers are willing to use either one good or the other to fulfill their needs, then a decrease in the price of one will drive consumers toward that good and away from the other. The cheaper nails are, the less likely I am to buy an industrial-strength glue when either one or the other can be used to achieve the desired goal of attaching pieces of wood.

66. AREEDA & HOVENKAMP, *supra* note 64, at 89.

Wash U Law Repository

power, there must be some barrier to entry that prevents other producers from making goods that consumers view as sufficiently close substitutes.

Patents can be the needed barrier to entry. If a good is patented, producers other than the patent owner cannot legally produce the good, or at least cannot do so without paying for a license. However, most patents do not generate significant market power.[67] Market power is contingent on the relationship between the scope of a patent and the scope of a market. If the scope of a patent and a market are coextensive, or the market is a subset of the patent, a patent grants its owner significant market power. However, the scope of a patent rarely encompasses an entire market. Sometimes, there are goods beyond the scope of a patent that are perfect economic substitutes for the patented goods, meaning that the patent does not grant its owner any market power at all.[68] In between these extremes, a patent owner may have market power that falls anywhere on a continuum from strong to weak, depending upon just how close the economic substitutes beyond the scope of the patent are.[69] The more distant the closest, unclaimed substitute technology is, the greater the rent that the inventor can expect from a patent.[70] More extensive patent scope is therefore imperfectly correlated with greater market power. Broader claims are more likely to encompass the full set of reasonable economic substitutes and to leave only more imperfect substitutes as the unpatented competition.

A functional claim is unusually likely to map onto an entire market, or, at least, to reach toward a market and create significant market power.[71] The reason why is that markets, too, are often defined by the very functional properties of goods that are used to mark the outer boundary of a functional claim. Consumer preferences determine whether goods are economic substitutes, and consumer preferences are usually keyed to what a technology can do, i.e., to its behavioral capacities. Inversely stated,

---

67. Ill. Tool Works Inc. v. Indep. Ink, Inc., 547 U.S. 28, 43 n.4 (2006) (agreeing with "the vast majority of academic literature" that market power should not be inferred from the existence of a patent). Of course, one reason why patents rarely generate market power today is the doctrinal prohibition within patent law on functional claims—the very prohibition that this section seeks to justify.

68. WILLIAM M. LANDES & RICHARD A. POSNER, THE ECONOMIC STRUCTURE OF INTELLECTUAL PROPERTY LAW 374 (2003).

69. *Id.* at 376 (noting the degree of market power depends on "how elastic . . . the demand of a given good" is).

70. SCOTCHMER, *supra* note 40, at 103–07.

71. *See* Mark A. Lemley & Mark P. McKenna, *Is Pepsi Really a Substitute for Coke? Market Definition in Antitrust and IP*, 100 GEO. L.J. 2055, 2112 (2012) (noting that when "patentees claim any device that performs a particular function" the claim "will often, though not always, coincide with market power").

consumers are often indifferent to the structural properties of a technology in and of themselves, so long, of course, as different structural properties do not give rise to different functional properties. A consumer wants a coffee sleeve that prevents his fingers from being singed by the heat dissipating through a disposable coffee cup; he is indifferent to whether that heat-dissipating function is performed by a substance that has the structure of corrugated cardboard or cork. A consumer who has an itch in the middle of her back does not say to herself, "I want a long stick with pointy claws on the end, and nothing else." Rather, she says, "I want something that will scratch this hard-to-reach itch, and nothing else." Given the primary importance of a technology's function and the secondary importance of its structure in consumers' utility functions, markets are commonly composed of a genus of goods that all perform the same, or similar, functions for their end users.[72] In sum, goods that have the same behavioral capacities are frequently economic substitutes in the same market, and functional claims that encompass all goods that have a behavioral capacity readily map onto markets.

Because they reach toward markets, functional claims generate unusually significant static, dynamic, and administrative costs. These elevated costs are most notable when they are compared to the costs of claims that include structural limitations and that are thus limited to one answer to the "how" question of technology.[73]

Functional claims that reach toward markets entail larger static costs: more market power for the patent owners means that more consumers are priced out of the market. In contrast, if a claim is limited to one structural means of achieving a functionally specified goal, it is more likely that there will be non-infringing economic substitutes and less market power. There is some consensus that patent protection provides sufficient incentives for innovation when it does not encompass entire markets but rather facilitates monopolistic competition in which producers have exclusive rights to products that are reasonable, but imperfect, substitutes.[74] Functional claims that reach toward markets are unlikely to strike this balance.

---

72. This is clearly an oversimplification. For example, branding also influences consumer preferences. Some people want an iPod, not a generic portable digital music player, and this consumer preference can impact market power. *Id.* at 2080–91.

73. *See supra* note 33 and accompanying text.

74. Herbert J. Hovenkamp, *Response: Markets in IP and Antitrust*, 100 GEO. L.J. 2133, 2148 ("[W]e protect most IP not because we expect that it will create monopoly, but rather because it will create sufficient product differentiation to justify short-run returns above marginal cost that are sufficient to incentivize the significant fixed cost investment that innovation requires."); Edmund W.

Wash U Law Repository

Functional claims that encompass markets also entail unusually large dynamic costs. When claim scope is limited only by a behavioral capacity that defines a market, competitors are likely to recognize in advance of even trying that they cannot design around the claim. They foresee that they will be unlikely to be able to invent a non-infringing technology that, at the same time, serves as a reasonable economic substitute for consumers and yet does not have the behavioral capacity that defines claim scope. In turn, the *ex ante* realization that design-around will not be successful significantly dampens entrepreneurial interest in follow-on innovation.[75] In the words of the Supreme Court, it is possible that "inventive genius may evolve many more devices" in addition to the device disclosed by an inventor "to accomplish the same [claimed] purpose" or function, but the "inventive genius" may choose not to invest the resources needed to invent the new devices because he is "frightened from the course of experimentation by broad functional claims."[76] In contrast, when the structural properties of goods that embody one answer to the "how" question of technology are included as claim limitations, competitors are likely to be more optimistic about the likelihood of successful design

Kitch, *Property Rights in Inventions, Writings, and Marks*, 13 HARV. J.L. & PUB. POL'Y 119, 122–23 (1990) ("The claims of most issued patents are so narrow that competitors can devise many ways of achieving the same thing as the subject matter of the claim."); Lemley & McKenna, *supra* note 71, at 2088 ("Many product patents do little more than create relatively minor enhancements in a product that make it distinctive to one group of customers, and competitors in that product are likely to have their own offsetting patented enhancements. As a result, the markets for automobiles, vacuum cleaners, cleansers, and pharmaceuticals are characterized by numerous patents, most of which suffice to make their products somewhat distinctive in a product differentiated market.").

75. Courts describe the incentive to design around a patent as a socially beneficial incentive. *See, e.g.*, State Indus., Inc. v. A.O. Smith Corp., 751 F.2d 1226, 1236 (Fed. Cir. 1985). However, some scholars argue that designing around patents is social waste that should be deterred. *Compare* James Bessen & Eric Maskin, *Sequential Innovation, Patents, and Imitation*, 40 RAND J. ECON. 611, 613 (2009) ("[A]n important role of patents is to encourage innovative activity on the part of others who would otherwise be inclined merely to imitate."), *with* Michael Abramowicz, *The Uneasy Case for Patent Races over Auctions*, 60 STAN. L. REV. 803, 807 (2007) (emphasizing the duplicative, socially wasteful efforts entailed in designing around a patent). Subsequent inventors confronted with a functional claim still have some incentive to engage in follow-on innovation to the extent that they can obtain a subservient, blocking patent on an improved machine for achieving the claimed function. However, the incentive is reduced because subsequent inventors know that, even if they can obtain a license to make the improvement, they will have to channel part of their profit back to the earlier patentee. Scotchmer, *supra* note 51, at 33–34.

76. Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 12 (1946). The dynamic costs of functional claims also create feedback that leads to static costs over a larger fraction of a patent's term. Economists often assume that successful design around reduces the effective term of a patent and thus the static costs of patent protection. *See* Jerry R. Green & Suzanne Scotchmer, *On the Division of Profit in Sequential Innovation*, 26 RAND J. ECON. 20, 22–25 (1995); Ted O'Donoghue et al., *Patent Breadth, Patent Life, and the Pace of Technological Progress*, 7 J. ECON. & MGMT. STRATEGY 1, 20 (1998). By reducing design-around activities, functional claims increase static costs.

around. They can find another answer to the "how" question more readily than they can shift consumers' utility functions and make consumers desire a technology that does something different.

Functional claims that reach toward markets may also lead to unusually significant administrative costs in the form of overlapping patent rights. Patent term lasts twenty years from the date a patent application is filed as a legal matter.[77] However, technological obsolescence means that the technology described by a patent claim may lose its commercial value before the legal term of a patent ends, making the effective term of a patent less than its legal term.[78] When a functional claim reaches toward a market and patent scope is limited only by the behavioral capacity of a technology, the patented technology is less likely to lose its commercial value before the end of the patent's legal term. Consumers' functional needs are more stable over time than the structural answers to the "how" question of technology. As a result, functional claims remain commercially valuable for a longer period of time, leading to a greater number of overlapping patents on any given technological good.

The costs generated by the ability of functional claims to reach toward and encompass markets provide an economic argument against functional claims that is concededly both under- and over-exclusive. It is under-exclusive in that claims defined partially or even solely by structural limitations sometimes give a patent owner significant market power. If a structural claim describes a bottleneck technology for which there are no reasonable economic substitutes, then it, too, may grant an inventor significant market power.[79] More significantly, the argument is over-exclusive in that not all functional claims will be broad enough to encompass or even reach toward markets. Like other limitations, functional limitations exist on a wide range of levels of generality.[80] Only the more general of the functional claims are likely to map onto markets and generate unusually significant social costs.

On the general end of the spectrum, patent limitations may recite a behavior that is an *end-user preference*—that is, a behavior that is directly valued by an end consumer and that may even define the consumer's need itself. Because markets are strongly influenced by consumer preferences,

---

77. 35 U.S.C. § 154(a)(2) (2006).

78. *See supra* note 76.

79. *Cf.* John R. Allison et al., *Valuable Patents*, 92 GEO. L.J. 435, 440 (2004) (noting that claims with narrow scope can still be economically valuable). Of course, it is often possible to design around a structural claim by inventing economic substitutes with different structural properties but the same functional properties. *Cf. supra* notes 75–76 and accompanying text.

80. *See supra* text accompanying note 27.

Wash U Law Repository

claims with limitations that describe end-user preferences are particularly likely to map onto markets.[81] For example, Morse's claim describes a set of technologies by describing an end-user preference: consumers want a device capable of communicating at a distance.[82]

On the specific end of the spectrum, functional limitations may not define an end-user preference. They may instead describe only one of several, alternative ways in which an end-user preference can be fulfilled, and they may thus be functional analogs of the answers to the "how" question of technology that are usually provided by a recitation of a technology's structure.[83] These specific behaviors can be called *how-functions.* A functional claim reciting a how-function is less likely to encompass a market, as there are non-infringing technologies that satisfy the end-user preference and that are economic substitutes for the claimed technology.[84] For example, the claim to a device for measuring the depth of oil wells in *Halliburton* arguably recites a how-function.[85] Consumers are not likely motivated to purchase a means for measuring echoes from the shoulders of pipe joints in an oil well *per se*; they want to purchase a means of measuring the depth of oil wells, however that goal is accomplished. It is possible that methods of measuring the depth of oil wells that did not involve measuring echoes from the pipe junctions—or that did not involve measuring echoes at all—already existed or could be developed, leaving reasonable substitutes beyond the scope of the claim.

To sum up, a prohibition on functional claims is over-exclusive because it invalidates not only functional claims that describe highly general end-user preferences but also functional claims that describe narrowly drawn how-functions. Yet, despite this over-exclusiveness, a prohibition on purely functional claims is a reasonable proxy for a prohibition on claims that reach toward markets.[86] The reason is that neither a straight-up market analysis nor a more narrowly tailored proxy is administratively feasible.

In theory, one could imagine a patent regime that incorporated a straight-up market analysis as a validity requirement. Patent claims that grant excessive market power could be narrowed or invalidated, regardless

81. *See supra* notes 71–72 and accompanying text.

82. *See supra* text accompanying note 57.

83. *See supra* notes 32–33 and accompanying text.

84. A functional claim could in theory recite such an exacting set of performance requirements that only a single structural device infringes.

85. *See supra* text accompanying notes 60–62.

86. Patent law's prohibition on functional claims is thus a largely unexplored aspect of the antitrust-patent interface that is embedded within patent law's validity rules.

of whether they are functionally or structurally defined. However, such a regime would not be feasible as a practical matter. Contemporary patent law does not attempt to define markets in their technical sense, and with good reason.[87] If claim validity is to remain stable, the relevant question in the assessment of permissible claim scope cannot be based on a conventional static market analysis, namely whether there are currently available substitutes beyond the scope of a patent. Rather, the question would have to focus on whether follow-on innovation is likely to produce non-infringing substitute products in the near future. Given the complex and speculative nature of the identification of innovation markets in an antitrust analysis,[88] answering a similar question as a routine part of the doctrinal analysis of permissible claim scope would be an extremely unwieldy exercise.

Alternatively, one could imagine a patent regime with a prohibition on functional claims that is more narrowly tailored to the functional claims that are unusually likely to reach toward or encompass markets. For example, perhaps only claims with functional limitations reciting end-user preferences could be invalidated, and claims with functional limitations reciting how-functions could be sanctioned.[89] Yet, while this distinction accurately identifies two ideal types, it is untenable as a legal distinction for determining claim validity. The distinction would require judges and examiners to differentiate the more fundamental needs that motivate consumer decisions from the mere means through which consumers choose to satisfy those needs. Defining consumer preferences directly in the minds of consumers, rather than inferring them from purchasing behavior, is not a task that any economist or lawyer should undertake lightly. The only administrable proxy for claims to end-user preferences is therefore an over-exclusive prohibition on all functional claims.[90]

---

87. Even though understanding markets is essential for understanding the private and social welfare effects of intellectual property, intellectual property doctrine shies away from the formal identification of markets. *See* Lemley & McKenna, *supra* note 71, at 2059–77.

88. Hovenkamp, *supra* note 74, at 2135 ("[M]arket power assessment will probably never do a good job of taking innovation into account because innovation is so badly behaved . . . ."); Herbert Hovenkamp, *Restraints on Innovation*, 29 CARDOZO L. REV. 247, 254–60 (2007).

89. *See supra* text accompanying notes 81–85.

90. The enablement doctrine cannot create a more closely tailored proxy, either. *See infra* Part II.D.

Wash U Law Repository

3. *Plugging up Spillovers*

A final argument in support of prohibiting purely functional claims considers the alternative types of claims that are available for providing an incentive to innovate. If functional claims were to be prohibited, what protection would be left for inventors? Functional claims can be deemed to be impermissibly overbroad in part because the patent protection that is available in the absence of such claims is robust.[91] No inventor who could obtain a purely functional claim if such claims were permitted would be denied patent protection if such claims were not permitted. Every inventor who seeks a novel and nonobvious claim defined solely by one or more behavioral capacities can also seek a narrower claim that recites some of the structural properties of the technology that the inventor has actually produced as limitations as well. A prohibition on functional claims only reduces the scope of the protection to which the deserving inventor is entitled. It increases the "spillover" of patent rights when inventors think up new behavioral capacities, allowing more of the benefit of innovation to flow to the public and less to be internalized by an inventor.[92] Even absent purely functional claims, the patent regime still provides incentives to think up new behavioral capacities for technologies.

C. *The Hidden, Unified Design Principle*

To address the overbreadth of functionally delineated claims, patent law employs what this Article terms the *invention-structure equation*. This doctrinal mechanism is simple in theory, even if it is often messy in the details of its application. Courts first make an ontological assumption about the nature of an invention that merits patent protection: the identity or defining quality of an invention—i.e., what makes an invention the invention that it is—resides in some subset of the physical, structural properties of the technology produced by an inventor.[93] Ontological

91. Debates over the optimal scope of patent protection are necessarily relative; the merits of any given position must be measured in relation to the merits of other possible positions. Imperfect solutions are frequently embraced when there is no better solution.

92. There is nothing inherently inefficient about positive externalities that remain uninternalized. *See, e.g.*, John P. Dawson, *The Self-Serving Intermeddler*, 87 HARV. L. REV. 1409, 1412 (1974) ("Uncompensated gains are pervasive and universal; our well-being and survival depend on them."); Frischmann & Lemley, *supra* note 48, at 258–84; Wendy J. Gordon, *On Owning Information: Intellectual Property and the Restitutionary Impulse*, 78 VA. L. REV. 149, 167–69 (1992) ("Culture *is* interdependence.").

93. For a broader argument explaining why the definition of what an inventor has invented should look to inventive properties of embodiments, rather than to either inventive ideas or inventive

rhetoric that characterizes the "identity" of a protectable invention and that posits what a protectable invention "is" litters patent opinions employing the invention-structure equation.[94] With this ontology in place, the courts have a metric for determining whether claims are overbroad with respect to an inventor's contribution to technological progress.[95] Courts categorically invalidate any claim delineated solely by a functional description of a technology as *per se* overbroad, and they uphold claims with functional limitations only if the claims also have limitations that refer to the physical, structural properties that define the invention. The first part of this inquiry is a bright-line rule, whereas the latter involves a detailed, and often controversial and underdetermined, inquiry into the set of physical, structural properties that define the protectable invention.[96]

There is no single patent doctrine that operationalizes the invention-structure equation. Rather, the invention-structure equation undergirds three patent doctrines that are conventionally viewed as largely unrelated: the section 101 doctrine of patentable subject matter,[97] the Supreme Court's prohibition on functional claims (and the congressional response codified in section 112(f)),[98] and the section 112(a) doctrine of written description.[99] To date, these three doctrines have been studied by scholars

---

embodiments as wholes, see Collins, *Getting into the "Spirit," supra* note 44.

94. *See, e.g.*, Regents of the Univ. of Cal. v. Eli Lilly & Co., 119 F.3d 1559, 1568 (Fed. Cir. 1997) ("A definition by function . . . does not suffice to define the [invented] genus [of technologies] because it is only an indication of what the [claimed invention] does, rather than what it is."); Amgen, Inc. v. Chugai Pharm. Co., 927 F.2d 1200, 1206 (Fed. Cir. 1991) ("It is not sufficient to define [a chemical] solely by its principal biological [i.e., functional] property . . . because an alleged conception having no more specificity than that is simply a wish to know the identity of any material with that biological property."); *In re* Swinehart, 439 F.2d 210, 212 (C.C.P.A. 1971) (describing the structural properties of an invention as what the invention "is" while describing the invention's functional properties as merely what the invention "does"); *see also* MPEP § 2173.05(g) (8th ed. Rev. 8, July 2010). This patent-law ontology is not far removed from the common-sense ontology of things that follows from a materialist world view: the functional properties of a technology can be explained with reference to the technology's (presumptively more fundamental) structural properties. *See supra* notes 32–33 and accompanying text.

95. *See supra* notes 43–44 and accompanying text (discussing the commensurability principle).

96. One difficult issue that is inherent in the administration of the invention-structure equation is identifying the subset of the structural properties of a disclosed technology that defines the invention. This issue has yet to be addressed by the courts in a satisfying manner, and it is one on which the distinct doctrinal manifestations of the invention-structure equation may differ. For example, some cases employing the invention-structure equation suggest that the structural properties that differentiate an invention from the prior art (i.e., the structural points of novelty) define the invention. *See, e.g.*, *infra* note 119. In contrast, the rules of means-plus-function claiming require all functional limitations, whether at the point of novelty or not, to be limited to the corresponding structures disclosed in the specification. *See infra* notes 123–25 and accompanying text.

97. 35 U.S.C. § 101 (2012).

98. *Id.* § 112(f).

99. *Id.* § 112(a). Courts may also sometimes implement the invention-structure equation through

Wash U Law Repository

and applied by courts only within distinct analytical silos, likely because the path-dependent, historical evolution of patent law has grounded them in different statutes and expressed them in distinct rhetorical formulations. This compartmentalization has hidden the importance of the invention-structure equation as the court's primary tool for curtailing the overbreadth of functional claims.[100] The cross-cutting examination of the doctrines that follows reveals a hidden, unified design principle of the patent regime. Patent doctrine is, at least in one respect, less complex and fragmented than it initially appears.[101]

1. Patentable Subject Matter

Throughout the nineteenth century, courts invalidated claims that described "principles" or "principles in the abstract."[102] Today, these cases are usually treated as cases rooted in the section 101 doctrine of patentable subject matter.[103] Although the rhetoric of these opinions does not expressly identify functional claim language as the problem, they all involved claims that were defined solely by what the technology could do. Furthermore, in all of these cases, the courts implied that the claims would have been valid if, counterfactually, some of the features of the machines invented by the inventors—i.e., some of the structural properties of the claimed technology—had been added as limitations on claim scope.

For example, in *Wyeth v. Stone*, Justice Story invalidated a claim for "cutting ice by means of any power, other than human power."[104] This was

---

the default (i.e., not 112(f)) rules of claim construction. *See* Phillips v. AWH Corp., 415 F.3d 1303, 1311–24 (Fed. Cir. 2005) (en banc). They may curtail permissible claim scope by construing functional claim language so that it refers only to technologies that possess the structural properties that constitute an invention. *See, e.g.*, Symbol Techs., Inc. v. Lemelson Med., Educ. & Research Found., 301 F. Supp. 2d 1147, 1164 (D. Nev. 2004) (construing the term "scanning apparatus" narrowly so that it includes some structural limitations).

100. For example, Dan Burk and Mark Lemley do not mention the invention-structure equation in any of its doctrinal guises in their otherwise extensive catalog of actual and potential "policy levers" for optimizing patent scope. *See* BURK & LEMLEY, *supra* note 1, at 122–28, 135–37.

101. One difference that exists within this unified design principle is the difference between invalidity rules and rules of claim construction. In all three areas, the courts initially implemented the invention-structure equation through an invalidity rule: overbroad claims are invalid. When Congress enacted what is now section 112(f) in 1952, it transformed the invention-structure equation into a rule of claim construction: overbroad claims are narrowed and upheld. *See infra* text accompanying notes 123–25.

102. *See, e.g.*, O'Reilly v. Morse, 56 U.S. (15 How.) 62 (1853); Le Roy v. Tatham, 55 U.S. (14 How.) 156 (1852); Wyeth v. Stone, 30 F. Cas. 723 (C.C.D. Mass. 1840) (Story, J.); Whittemore v. Cutter, 29 F. Cas. 1123, 1124 (C.C.D. Mass. 1813) (Story, J.).

103. *See, e.g.*, Bilski v. Kappos, 130 S. Ct. 3218, 3253 (2010).

104. *Wyeth*, 30 F. Cas. at 727.

a functional claim: it described the behavior of which the device was capable without describing any of the structural properties of the device. The claim was thus "broader than the actual invention" and impermissibly encompassed "a principle in the abstract" because it did not recite "any particular . . . machinery, by which ice is to be cut" as a limitation.[105]

Perhaps most famously, the Supreme Court in *O'Reilly v. Morse* invalidated one of Samuel Morse's claims to the telegraph for overbreadth: "the use of the motive power of the electric or galvanic current . . . for marking or printing intelligible characters, signs, or letters, at any distances."[106] Again, this is a functional claim: it describes the behavioral capacity of the device. The Court identified the problem with this claim as the lack of any limitation that recited the structural properties of the technology that Morse actually invented: the functional claim was overbroad because "it matter[ed] not by what process or machinery the result [was] accomplished."[107] Today, *O'Reilly* is often cited as a case that establishes a prohibition on the patenting of "a natural law," i.e., some aspect of nature that existed before man discovered it.[108] This characterization of these cases, however, is misleading. Telegraphs are no more natural than run-of-the-mill patentable inventions. The problem in these cases was the use of functionally defined limitations to delineate claim scope, and these cases therefore represent the early origins of the invention-structure equation as a mechanism for curtailing permissible claim scope.

2. *Functional Claim Limitations*

In the first half of the twentieth century, the Supreme Court made explicit its objection to functional claims. The Court invalidated a series of patent claims on the grounds that the claims were overbroad because they only recited the functional properties of technology that an inventor had produced and thus were not limited to an inventor's actual invention.[109] As

105. *Id.*

106. *O'Reilly*, 56 U.S. at 112.

107. *Id.* at 113; *see also id.* at 117 (noting that an effect as a claim limitation "must be combined with, and passed through, and operate upon, certain complicated and delicate machinery" in order to yield a patentable claim).

108. *See* Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1293–94, 1301 (2012).

109. *See, e.g.*, Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1 (1946); Gen. Elec. Co. v. Wabash Appliance Corp., 304 U.S. 364 (1938); Holland Furniture Co. v. Perkins Glue Co., 277 U.S. 245 (1928). For more in-depth treatments of these cases, see Paul M. Janicke, *The Crisis in Patent Coverage: Defining Scope of an Invention by Function*, 8 HARV. J.L. & TECH. 155 (1994); Mark D.

Wash U Law Repository

a mid-twentieth-century patent treatise noted: "It is possible that a claim for all means of arriving at a desired result would be broad enough to cover later discovered means wholly independent of the first means for arriving at the same final result. In that case, the inventor would be over-protected."[110] In this passage, "means" is code for a technology with particular structural properties, and "a desired result" is a functional property.

In *Holland Furniture Co. v. Perkins Glue Co.*,[111] an inventor developed a starch-based glue that was useful for wood veneering because, among its properties, it had the low water content that previously had only been attainable in glue made from animal substances.[112] The inventor obtained a functional claim to any starch-based glue "having substantially the properties of animal glue."[113] Claim scope was thus delineated "not in terms of [the invention's] own physical characteristics or chemical properties . . . but wholly in terms of the manner of use of the product."[114] The Supreme Court invalidated the claim because it employed only functional limitations. "A claim so broad, if allowed, would operate to enable the inventor who has discovered that a defined type of starch answers the required purpose to exclude others from all other types of starch" with different chemical compositions.[115]

In *General Electric Co. v. Wabash Appliance Corp.*,[116] an inventor obtained a functional claim that encompassed tungsten filaments "made up mainly of a number of comparatively large grains of such size and contour as to prevent substantial sagging and offsetting during a normal or commercially useful life for . . . a lamp or other device."[117] The Supreme Court held the claim "invalid on its face" because of its functional language.[118] "The claim . . . falls within the condemnation of the doctrine that a patentee may not broaden his product claims by describing the

---

Janis, *Who's Afraid of Functional Claims? Reforming the Patent Law's § 112*, ¶ 6 *Jurisprudence*, 15 SANTA CLARA COMPUTER & HIGH TECH. L.J. 231 (1999); Lemley, *supra* note 1, at 5–15. Congress reacted to these cases by enacting what is now section 112(f). *See infra* text accompanying notes 123–25.

110. RISDALE ELLIS, PATENT CLAIMS 338–39 (1949).
111. 277 U.S. 245 (1928).
112. *Id.* at 247.
113. *Id.* at 250.
114. *Id.* at 256.
115. *Id.* at 257.
116. 304 U.S. 364 (1938).
117. *Id.* at 368.
118. *Id.*

product in terms of function" and "vividly illustrates the vice of a description in terms of function" in a claim.[119]

Similarly, in *Halliburton Oil Well Cementing Co. v. Walker*, an inventor made an improvement in a device that records the echoes of sound waves sent into an oil well in order to measure well depth.[120] The inventor limited the scope of his claim with reference to a "means . . . for tuning [an echo-recording device] to the frequency of echoes . . . to clearly distinguish the echoes . . . from . . . each other."[121] The Court invalidated the claim, noting that the claim described the invention "in terms of what it will do rather than in terms of its own physical characteristics" and opining that the claim illustrated the "overhanging threat of the functional claim."[122]

The Supreme Court cases barring purely functional claims because of overbreadth remain good law today, but Congress has softened their impact. Describing the full scope of an invention without using functional limitations is sometimes difficult, and invalidating any claim with a functional limitation might leave inventors without effective patent protection. Therefore, Congress responded to *Halliburton* by enacting what is now section 112(f) as part of the 1952 Patent Act:

---

119. *Id.* at 371. The Court further noted that "the vice of a functional claim exists not only when a claim is 'wholly' functional, if that is ever true, but also when the inventor is painstaking when he recites what has already been seen, and then uses conveniently functional language at the exact point of novelty." *Id.*; *see also* Davis Sewing Mach. Co. v. New Departure Mfg. Co., 217 F. 775 (6th Cir. 1914) (holding that only functional claiming at the point of novelty is prohibited).

120. 329 U.S. 1 (1946).

121. *Id.* at 8–9.

122. *Id.* at 9, 12. To a contemporary ear, some language in *Halliburton* can be taken to imply that functional claims are problematic because they are indefinite—that is, their boundaries are impossible to discern. For example, *Halliburton* states that functional claims "fail adequately to describe the alleged invention." *Id.* at 14. An interpretation of *Halliburton* that focuses on the indefiniteness of functional claim language shows up in recent patent cases. *See, e.g.*, Laitram Corp. v. Rexnord, Inc., 939 F.2d 1533, 1536 (Fed. Cir. 1991); *In re* Swinehart, 439 F.2d 210, 214 (C.C.P.A. 1971); Supplementary Examination Guidelines for Determining Compliance with 35 U.S.C. 112 and for Treatment of Related Issues in Patent Applications, 76 Fed. Reg. 7162, 7165 (Feb. 9, 2011) [hereinafter PTO 112 Guidelines] (citing *Halliburton* for the proposition that "when claims merely recite a description of a problem to be solved or a function or result achieved by the invention, the boundaries of the claim scope may be unclear"). However, functional language is not inherently more ambiguous or vague than structural language. *See infra* text accompanying notes 242–44. *But cf. infra* note 245 (discussing the limited conditions under which functional claims are likely to be indefinite); *infra* note 210 (noting that the validity of functional software claims may be uncertain if courts invalidate them for overbreadth on an *ad hoc* basis). When *Halliburton* is viewed in light of the invention-structure equation, the statement that functional claim language fails to describe the invention means that the inventor's protectable invention resides in the structural properties of the technology that he produces and that the claim is overbroad because it does not recite those properties as limitations.

Wash U Law Repository

> An element in a claim . . . may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.[123]

This provision embodies a compromise. It allows functional claim limitations, but it requires that the scope of those limitations be determined using a special scope-restricting rule of claim construction: the functional language refers only to devices that have the "corresponding structure" or "material" that the patentee discloses in the specification, as well as its equivalents.[124] Because functional claim limitations are often drafted as a "means for" performing a function, limitations construed according to the rule set forth in section 112(f) are called means-plus-function limitations.[125]

In sum, the statutory rule restricts the scope of functional claim limitations to things that possess the structural properties that an inventor discloses in the specification. It turns the invention-structure equation into a rule of claim construction, eliminating the problem of overbroad claims before it arises and obviating the need to invalidate overbroad, functional claims.

3. The Written Description Requirement

Within the last two decades, the Federal Circuit has given birth to yet another doctrine that employs the invention-structure equation as a mechanism for curtailing permissible claim scope: the written description requirement.[126] The written description requirement mandates that the set of claimed technologies must remain commensurate with the set of technologies that the inventor "invented" or "possessed" at the time of

---

123. 35 U.S.C. § 112(f) (2012).

124. *Id.* Means-plus-function limitations employ a hybrid of central and peripheral techniques for delimiting claim scope. *Cf. supra* note 25. A series of limitations defines a periphery, but each of those limitations is defined with reference to the structural exemplar disclosed in the specification.

125. *See, e.g.*, *In re* Henatsch, 298 F.2d 954 (C.C.P.A. 1962).

126. *See* 35 U.S.C. § 112(a) (2012). The courts for several decades had been using the written description requirement to curtail the scope of claims amended during prosecution. *See In re* Ruschig, 379 F.2d 990 (C.C.P.A. 1967). The recent development is the application of the written description requirement to claims filed with the original patent application. *See, e.g.*, Ariad Pharm., Inc. v. Eli Lilly & Co., 598 F.3d 1336 (Fed. Cir. 2010) (en banc); Univ. of Rochester v. G.D. Searle & Co., 358 F.3d 916 (Fed. Cir. 2004); Enzo Biochem, Inc. v. Gen-Probe Inc., 323 F.3d 956 (Fed. Cir. 2002); Regents of the Univ. of Cal. v. Eli Lilly & Co., 119 F.3d 1559 (Fed. Cir. 1997).

filing.[127] As the cases discussed below demonstrate, the set of invented or possessed technologies is legalistic code for the set of technologies that possess the structural properties that constitute an invention under the invention-structure equation. The written description doctrine is therefore a variation on the theme of the Supreme Court cases from the nineteenth and early twentieth centuries that prohibit purely functional claiming.[128] It is another tool for invalidating excessively functional claims for overbreadth.[129]

In *Regents of the University of California v. Eli Lilly and Co.*,[130] the Federal Circuit invalidated a number of claims to DNA molecules for failure to satisfy the written description requirement.[131] The claims defined sets of DNA molecules with only a functional limitation—the claimed molecules had to encode for insulin, i.e., they had to possess the property

127. *Ariad Pharm*., 598 F.3d at 1351.

128. *See supra* Part II.C.1. Viewing the written description requirement as the suspenders in a belt-and-suspenders implementation of the invention-structure equation casts one commonly noted peculiarity of the written description doctrine in a new light. Most written description invalidations occur in the biochemical arts, and this empirical fact is conventionally interpreted to suggest that the Federal Circuit subjects claim scope to greater scrutiny, and allows only narrower claims, in the biochemical arts. *See* Janice M. Mueller, *The Evolving Application of the Written Description Requirement to Biotechnological Inventions*, 13 BERKELEY TECH. L.J. 615 (1998); Arti K. Rai, *Intellectual Property Rights in Biotechnology: Addressing New Technology*, 34 WAKE FOREST L. REV. 827, 831–38 (1999). Interestingly, the rules of means-plus-function claiming exhibit the inverse pattern. Functional claims in the biochemical arts are rarely, if ever, subject to the scope-narrowing rule of claim construction articulated in section 112(f). (This outcome may be explained in part by the formalistic nature of the threshold test the Federal Circuit employs to determine whether the rules of means-plus-function claiming apply. *See infra* notes 298–302 and accompanying text.) When both 112(f) and written description are understood to be manifestations of the invention-structure equation, it becomes clear that the Federal Circuit uses the written description doctrine to do the exact same work of curtailing the overbreadth of functional claims in the biochemical arts that it uses the rules of means-plus-function claiming to do in other arts. The written description doctrine does not make claims in the biochemical arts unusually narrow. Rather, it levels the playing field. It brings the invention-structure equation that already governs claims in other arts to bear in the biochemical arts. Kevin Emerson Collins, *An Initial Comment on Ariad: Written Description and the Baseline of Patent Protection for After-Arising Technology*, 2010 PATENTLY-O PAT. L.J. 60.

129. The primary function of written description is commonly identified as a prohibition on claims that are filed too early in time, before an inventor understands the structure of any of the embodiments that he is claiming. *See, e.g.*, BURK & LEMLEY, *supra* note 1, at 118. However, the not-too-early concern is just a limit condition of the not-too-broad concern. If an inventor has not disclosed the structure of any embodiment within the scope of the claims, the set of claimed technologies is never commensurate with the set of technologies that the inventor invented or possessed at the time of filing (because the set invented or possessed is a null set). The unified nature of the invention-structure equation can also be seen in the fact that the not-to-early written description cases are, as a matter of policy, indistinguishable from the means-plus-function cases in which the inventor fails to disclose any corresponding structure in the specification. *See infra* Part IV.B.1.

130. 119 F.3d 1559 (Fed. Cir. 1997).

131. *Id.* at 1566–69.

Wash U Law Repository

of being able to cause a cell to produce insulin.[132] The Federal Circuit invalidated the functional claims for overbreadth with respect to the technologies that were disclosed in the patent specification. It held that, in order to demonstrate "invention" or "possession" of a claimed genus of molecules, the specification had to reveal to the person having ordinary skill in the art (the "PHOSITA") the "structural or physical characteristics"—i.e., the physical, structural properties—that are shared by members of the genus.[133] The functional description of the claimed genus provided by the claims themselves "does not define any structural features commonly possessed by members of the genus," and it therefore does not allow the PHOSITA to "recognize the identity of the members of the genus."[134] Being able to "visualize" a technology lies at the heart of being in possession of an invention, and visualization, in turn, requires understanding physical structure.[135] In sum, *Eli Lilly* uses the invention-structure equation to invalidate overbroad, functional claims.

Two more recent written description cases also demonstrate that the written description doctrine implements the invention-structure equation. In *University of Rochester v. G.D. Searle & Co.*,[136] the Federal Circuit invalidated a claim to a method of "administering a non-steroidal compound that selectively inhibits activity of" a particular protein.[137] The claim recited only a functional property of the compound, and the patent did not disclose—let alone recite as a limitation on claim scope—the structural properties of any molecule capable of achieving the desired function.[138] Most recently, in *Ariad Pharmaceuticals, Inc. v. Eli Lilly & Co.*,[139] the court invalidated claims to methods of reducing the binding of a transcription factor to a family of genes.[140] The claims were purely functional—they "encompass[ed] the use of all substances that achieve the desired result"—and they were insufficiently limited by the structural

132. Technically, the claims described plasmids and microorganisms containing nucleotide sequences that encoded for insulin. *Id.* at 1563–64. The specification disclosed the DNA sequence of the insulin gene in rats, and the claims described nested classes of the insulin gene: the insulin gene in vertebrates, mammals, and both rats and humans. *Id.*

133. *Id.* at 1567; *see also id*. at 1569 (noting that written description "requires a kind of specificity usually achieved by means of the recitation of the sequence of nucleotides that [made] up" the DNA).

134. *Id.* at 1568. "Identity" is legal code for the structural properties of an invention under the invention-structure equation. *See supra* note 94.

135. *Eli Lilly*, 598 F.3d at 1568.

136. 358 F.3d 916 (Fed. Cir. 2004).

137. *Id.* at 918.

138. *Id.* at 927.

139. 598 F.3d 1336 (Fed. Cir. 2010) (en banc).

140. *Id.* at 1340.

properties of any molecule that could achieve that result.[141] As the Federal Circuit noted in *Ariad*, the written description requirement "is especially acute with genus claims that use functional language to define the boundaries of a claimed genus. In such a case, the functional claim may simply claim a desired result, and may do so without describing [the structures of the] species that achieve that result."[142]

D. The Limits of Enablement

The status of the invention-structure equation as the courts' principal tool for curtailing the overbreadth of functional claims has gone largely unappreciated. In part, this oversight flows from the heretofore hidden nature of the invention-structure equation as a deep design principle of patent law.[143] In part, however, it also follows from an overestimation of the efficacy of another patent doctrine that reins in permissible claim scope: enablement. The conventional wisdom in patent law is that enablement is the courts' best tool for curtailing claims that reach too far into yet-to-be developed technologies.[144] This belief, however, is highly suspect. Although enablement does curb some types of claim overbreadth, it is neither tasked with reining in the overbreadth of functional claims in most arts today nor well-suited for taking on that job tomorrow.

---

141. *Id.* at 1341, 1350.

142. *Id.* at 1349. The written description requirement diverges from the earlier Supreme Court cases administering the invention-structure equation in that the earlier cases invalidate a functionally drawn claim on its face whereas the written description requirement only invalidates a functionally drawn claim if there are insufficient structurally defined species disclosed in the specification. *See id.* ("[T]he specification must demonstrate that the applicant has made a generic invention that achieves the claimed result and do so by showing that that applicant has invented species sufficient to support a claim to the functionally-defined genus."). Additionally, if the PHOSITA is aware of a known correlation between function and structure, a claim with only functional limitations may be upheld. *Id.* at 1350.

143. *See supra* notes 97–101 and accompanying text.

144. For enablement's dominance in pioneering work on patent scope, see Merges & Nelson, *supra* note 43, at 845–52. More recently, enablement's dominance has surfaced in wide-spread arguments suggesting that the application of the written description doctrine to the claims originally filed with a patent application is superfluous in light of the enablement doctrine. *See, e.g.*, Enzo Biochem, Inc. v. Gen-Probe Inc., 323 F.3d 956, 976–83 (Fed. Cir. 2002) (Rader, J., dissenting); Mark D. Janis, *On Courts Herding Cats: Contending with the "Written Description" Requirement (and Other Unruly Patent Disclosure Doctrines)*, 2 WASH. U. J.L. & POL'Y 55 (2000); Mueller, *supra* note 128. The dominance of enablement in general discussions about how to use patent law's validity doctrines to best tailor patent scope extends into discussions about how to tailor patent scope in the software arts in particular. Robert P. Merges, *Software and Patent Scope: A Report from the Middle Innings*, 85 TEX. L. REV. 1627 (2007). *But see* Lemley, *supra* note 1 (arguing that section 112(f) is the key to curtailing the scope of software claims).

Wash U Law Repository

Like the written description requirement, enablement curbs permissible claim scope by tethering claims to the disclosure that an inventor makes in the specification.[145] However, it uses a different metric—that is, it looks to a different type of disclosed information—to ensure commensurability between an inventor's contribution to progress and permissible claim scope.[146] It looks to a patent specification's teachings concerning how to make and use an invention. More specifically, enablement requires that the disclosure teach the PHOSITA how to make and use a set of technologies without undue experimentation that is reasonably commensurate with the claimed set of technologies at the time of filing.[147] If a claim is overbroad with respect to the technologies that a patent disclosure teaches the PHOSITA to make and use, the claim is invalid for lack of enablement.

As a descriptive matter, the enablement doctrine is not the principal bulwark that holds back the type of claim overbreadth generated by functional claims. The rules of means-plus-function claiming enforce the invention-structure equation as part of claim construction.[148] Claim construction is the process through which courts determine the meaning of claim language, and it ordinarily occurs before any validity doctrine, including enablement, is considered.[149] The rules of means-plus-function claiming therefore deal with the problem of the overbreadth of functional claims before enablement even enters into the picture. Functional claims are usually construed so that they have limitations reciting some of the structural properties of what an inventor has invented, meaning that enablement does not today have to deal with the full extent of the potential overbreadth problem of functional claims.[150] Any argument that justifies the use of enablement as the best means of curtailing functional claims by extrapolating from the role that enablement plays in the contemporary patent regime misses the mark. Enablement operates only as a second line of defense against the overbreadth of functional claims.

---

145. Enablement and written description derive from intertwined words in the same statutory provision. *See* 35 U.S.C. § 112(a) (2012).

146. *But cf. infra* text accompanying notes 175–79 (noting that the enablement doctrine can become a manifestation of the invention-structure equation to restrict the scope of functional claims).

147. Nat'l Recovery Techs., Inc. v. Magnetic Separation Sys., Inc., 166 F.3d 1190, 1195–96 (Fed. Cir. 1999).

148. *See supra* text accompanying notes 123–25.

149. The meaning of claim language affects the breadth of claim scope, and the breadth of claim scope is often a critical fact in the enablement analysis. AK Steel Corp. v. Sollac, 344 F.3d 1234, 1241 (Fed. Cir. 2003).

150. Interestingly, even the doctrines that employ the invention-structure equation to invalidate overbroad claims have their greatest impact today in arts in which the rules of means-plus-function claiming are not often used. *See supra* note 128.

Of course, the status quo could be altered. The enablement doctrine could take on greater importance, and it could be positioned as the first, or perhaps only, line of defense against the overbreadth of functional claims. As a practical matter, however, enablement would be ill-suited to serving this role. Enablement has a blind spot. The enablement analysis is rooted in time on the date on which a patent application is filed,[151] and there is a type of overbreadth that cannot be seen by this time-bound PHOSITA: claimed embodiments that are unforeseeable or "unknown concepts" on that date.[152] This is precisely the type of overbreadth that functional claims create, making enablement a poor tool to use to curtail the overbreadth of functional claims.

Enablement challenges all involve allegations of gap technologies—that is, technologies that fall within the broader scope of a claim but not within the narrower set of technologies enabled by the specification. Most successful enablement challenges that invalidate claims involve gap technologies that were known concepts on the filing date, but that were not yet reduced to practice at that time. In this situation, the PHOSITA can readily imagine, conceptualize, or point to the gap technologies on the date of filing, allowing the lack of commensurability to be proven on that date. For example, in *Plant Genetic Systems v. DeKalb Genetics Corp.*, a patent broadly claimed "a plant cell" that had been transformed with a gene that endowed the cells with herbicide resistance.[153] The patent specification taught the PHOSITA how to make transformed dicot cells but not transformed monocot cells, making the transformed monocot cells the gap embodiments.[154] The Federal Circuit invalidated the cell claims for lack of enablement because these gap embodiments were "not an unknown concept" on the date of filing but were rather "specifically desired but difficult to obtain" at that time.[155] Similarly, in *Liebel-Flarsheim Co. v.*

151. Chiron Corp. v. Genentech, Inc., 363 F.3d 1247, 1254 (Fed. Cir. 2004).

152. Plant Genetic Sys. v. DeKalb Genetics Corp., 315 F.3d 1335, 1340 (Fed. Cir. 2003). For a lengthier exposition of the role of unforeseeability in enablement, see Collins, *Enabling*, *supra* note 50, at 1098–105. Enablement's blind spot results from what Rob Merges has famously called the temporal paradox of enablement and claim construction: enablement is performed on the date of filing, whereas claim construction is performed on the date of infringement. MERGES & DUFFY, *supra* note 58, at 295–97; Robert P. Merges, *Rent Control in the Patent District: Observations on the Grady-Alexander Thesis*, 78 VA. L. REV. 359, 379 n.73 (1992). However, because black-letter patent law in fact grounds both enablement and claim construction on the date of filing, the paradox underlying enablement's blind spot is more accurately described a meaning: enablement pays heed to denotational meaning, whereas claim construction fixes ideational meaning. Collins, *Enabling*, *supra* note 50, at 1099–100.

153. *Plant Genetic Sys.*, 315 F.3d at 1335.

154. *Id.* at 1338.

155. *Id.* at 1340. The transformed monocot cells were foreseeable after-arising technology, and

Wash U Law Repository

Medrad, Inc.,[156] the Federal Circuit invalidated a claim to a method of using a "high pressure power injector" for lack of enablement.[157] The gap embodiments were power injectors without pressure jackets, and they were undeniably known concepts at the time the claim was filed because the originally filed claims contained a pressure-jacket limitation that was removed during prosecution.[158] The enablement problem was thus not that the PHOSITA could not imagine or conceptualize the gap embodiments on the date of filing. Rather, the enablement problem was that, having readily imagined the gap embodiments, the PHOSITA did not know how to make them without undue experimentation.[159]

Inversely, if the gap technologies are after-arising in the stronger sense that they are unknown concepts to the PHOSITA on the date of filing, then enablement challenges routinely fail.[160] In this situation, enablement's temporally rooted PHOSITA cannot identify the unenabled, gap technologies, so he cannot marshal the evidence required to prove a lack of enablement. The classic case involving unknown-concept gap embodiments is *In re Hogan*.[161] In *Hogan*, a patent applicant claimed "a normally solid homopolymer" of a specified subunit.[162] The specification taught the PHOSITA how to make a normally solid homopolymer that had a low molecular weight and that was crystalline in nature. On a later date, another inventor made a normally solid homopolymer that had a high molecular weight and that was amorphous in nature.[163] The amorphous homopolymer was a gap embodiment: it was not enabled as a factual

thus known concepts on the date of filing, because the distinction between monocots and dicots was well known in the art and the value of transformed monocots had been recognized. *Id.* at 1338.

156. 481 F.3d 1371 (Fed. Cir. 2007).

157. *Id.* at 1373.

158. *Id.* at 1374. Additionally, the specification explicitly discussed the difficulties of power injectors without pressure jackets. *Id.* at 1379–80. For another case in which the specification demonstrated that the gap embodiments in an enablement analysis were known concepts on the date of filing because it taught away from those embodiments, see *AK Steel Corp. v. Sollac*, 344 F.3d 1234 (Fed. Cir. 2003).

159. Another common scenario in which enablement challenges are successful involves claims that recite a large or open-ended range and specifications that teach how to make embodiments satisfying only a small portion of the claimed range. *See, e.g.*, MagSil Corp. v. Hitachi Global Storage Techs., Inc., 687 F.3d 1377 (Fed. Cir. 2012); *In re* Fisher, 427 F.2d 833 (C.C.P.A. 1970). Here, again, the gap embodiments are known concepts on the date of filing. If a claim describes a purity range from one to ten (of some arbitrary unit), then a compound with purity of nine is a known concept even if there is no embodiment with a purity of nine that is enabled by the specification.

160. *But see infra* text accompanying notes 175–79 (noting that the enablement doctrine can restrict the scope of functional claims by becoming a manifestation of the invention-structure equation).

161. 559 F.2d 595 (C.C.P.A. 1977).

162. *Id.* at 597.

163. *Id.* at 601.

matter by the specification, but it was within the scope of the claim.[164] However, the court exempted the gap embodiment from enablement's commensurability analysis because it was a "later state of the art," i.e., it was an unknown concept on the date the patent was filed.[165] At the time of filing, the claim language "had a meaning to one skilled in the art that was coextensive with the species," suggesting that the PHOSITA could not imagine or conceptualize the amorphous homopolymer.[166] The concept of a high molecular weight, amorphous homopolymer of the specified subunit was not a within the grasp of the PHOSITA on the date the claim was filed, so the PHOSITA could not mount a successful invalidity argument that sounded in enablement.[167]

At first glance, enablement's blind spot may seem odd as a policy matter in that it paradoxically invalidates claims that encompass less radical improvements and upholds claims that encompass more radical improvements.[168] However, to the contrary, enablement's blind spot in fact serves an important policy function: the blind spot places an obligation on a patent applicant to draft narrower claims and avoid claim overbreadth only when the applicant can reasonably be held to be aware of the overbreadth at the time of filing and thus to have the capacity to draft narrower claims.[169] When claims encompass gap embodiments that were known concepts on the date of filing, the patent drafter should have known the claims were overbroad, and he should have drafted narrower claims excluding the gap embodiments.[170] Thus, when claims are held invalid for

164. The *Hogan* concurrence argued that the claim was not broad enough to encompass the amorphous homopolymer. *Id.* at 609–11 (Miller, J., concurring). However, the patent that the PTO eventually issued was, after litigation, found to be broad enough to encompass after-arising homopolymers. U.S. Steel Corp. v. Phillips Petroleum Co., 865 F.2d 1247 (Fed. Cir. 1989). Furthermore, it would be poor policy to categorically exclude later-developed technology from earlier-issued claims. *See supra* notes 49–52 and accompanying text.

165. *Hogan*, 559 F.2d at 605–07; *see also* Plant Genetic Sys. v. DeKalb Genetics Corp., 315 F.3d 1335, 1340 (Fed. Cir. 2003) ("In *Hogan*, amorphous propylene, on the record before the court, was not known . . . when the application was filed.").

166. *Hogan*, 559 F.2d at 610 (Miller, J., concurring).

167. For a more recent application of *Hogan* in a case involving a gap embodiment that was an unknown concept on the date of filing, see *Chiron Corp. v. Genentech, Inc.*, 363 F.3d 1247 (Fed. Cir. 2004).

168. Robin Feldman, *Rethinking Rights in Biospace*, 79 S. CAL. L. REV. 1, 17 (2005); Ellen P. Winner, *Enablement in Rapidly Developing Arts—Biotechnology*, 70 J. PAT & TRADEMARK OFF. SOC'Y 608, 624 (1988). The paradox assumes that the process of thinking up a previously unknown concept is a more radical advance over the prior art than the enablement of a known concept, but this assumption is not always a good one.

169. Collins, *Enabling*, *supra* note 50, at 1102–03.

170. The eventual invalidation during litigation of an overbroad claim that encompasses known-concept after-arising technology should therefore leave an inventor with narrower protection for the technology he actually did invent. Patents are conventionally drafted with a series of telescoping

Wash U Law Repository

lack of enablement because they encompass foreseeable gap embodiments, the patentee's "difficulty in enabling the . . . claims is a problem of its own making."[171] In contrast, when claims encompass gap embodiments that were not known concepts on the date of filing, there is no reason to suspect that the patent drafter knew that the claims were overbroad. Without access to the requisite concepts, the patent applicant cannot be expected to draft narrower claims that exclude the gap embodiments.[172] Here, the difficulty in enabling the claims is not of the patentee's own making. Because the patent applicant could not be expected to have filed a narrower claim, a claim that encompasses after-arising technology that is an unknown concept on the date of filing should not be invalidated for lack of enablement.

Once enablement's blind spot is recognized, it is easy to see why enablement is an ineffective curb on the overbreadth of functional claims. Functional claims are problematic because the type of overbreadth at issue is precisely the type that falls within enablement's blind spot: the gap technologies are unknown concepts. Consider again the claim in *Wyeth v. Stone* to a means of cutting ice other than by human power.[173] This is a clear example of an overbroad, functional claim. Yet, at the time of filing, the PHOSITA likely could not imagine or conceive any particular gap embodiment; the PHOSITA could not describe an unenabled means of achieving the claimed functional goal.[174] Thus, while the PHOSITA may strongly suspect that means of cutting ice other than the one disclosed in

claims at nested levels of generality. If the broader claims are invalidated, the narrower claims still give an inventor some patent protection. When after-arising technology is a known concept on the date of filing, the patent drafter was perfectly capable of drafting the narrower claims excluding the after-arising technology and encompassing only the technology that the inventor actually did invent.

171. MagSil Corp. v. Hitachi Global Storage Techs., Inc., 687 F.3d 1377, 1384 (Fed. Cir. 2012).

172. Courts sometimes justify enablement's blind spot by discussing what information a reasonable applicant can be expected to disclose rather than by focusing on what claims a reasonable applicant can draft. "The law does not expect an applicant to disclose knowledge invented or developed after the filing date. Such disclosure would be impossible." *Chiron*, 363 F.3d at 1254. A focus on what an applicant is capable of disclosing misses the point because it cannot differentiate between *Hogan* and *Plant Genetic Systems*. In cases in which the after-arising technology is specifically desired but difficult to obtain at the time of filing, it is impossible to disclose the after-arising technology at the time of filing, but a claim that encompasses the after-arising technology should be invalid under the enablement doctrine. It is in part because an enabling disclosure is impossible at the time of filing that the *Plant Genetics Systems* claim should not be upheld.

173. *See supra* text accompanying notes 104–05.

174. Conception is wound up with understanding the structural properties of an invention: "Conception requires . . . the idea of the structure of the chemical compound." Oka v. Youssefyeh, 849 F.2d 581, 583 (Fed. Cir. 1988). It is for this reason that the claims that fail the written description requirement are also often claims that had not yet been conceived on the date of filing. Fiers v. Revel, 984 F.2d 1164 (Fed. Cir. 1993).

the specification would eventually be developed, the PHOSITA cannot marshal the evidence needed to prove the overbreadth on the date of filing.

Despite the difficulties of proving the overbreadth of a functional claim under the enablement doctrine, courts do use enablement to invalidate functional claims from time to time. To reach this result, however, the courts must selectively bracket *Hogan*, ignore enablement's blind spot, and transform enablement into yet one more vehicle for administering the invention-structure equation. For example, consider the Federal Circuit's enablement holding in *Amgen, Inc. v. Chugai Pharmaceutical Co.*[175] *Amgen* involved a functional claim to a set of DNA molecules: all DNA molecules with sequences that encode for polypeptides that are sufficiently duplicative of erythropoietin (EPO) to have the behavioral capacity of increasing the production of red blood cells.[176] The specification disclosed a DNA molecule with a sequence that encoded for EPO itself, but it gave no indication of the portion of that sequence (or the portion of the polypeptide that constitutes EPO) that needed to be conserved for the DNA molecule to possess the claimed function.[177] The Federal Circuit invalidated the claim for lack of commensurability under the enablement doctrine because "[t]here *may* be many other genetic sequences" beyond those disclosed "that code for EPO-type products," i.e., that possess the claimed functional property.[178] The Federal Circuit did not yet know even on the date of writing an opinion in an infringement case if there was another, non-disclosed genetic sequence that possessed the claimed functional property, but it invalidated the claim for lack of enablement nonetheless. The reason for the invalidation is clear: the claim failed to recite a sufficient quantum of structural properties as limitations. Enablement may thus in exceptional cases be employed as a policy lever that allows courts to invalidate functional claims. But, when it is used in this manner, enablement is indistinguishable from written description. It becomes yet one more doctrinal manifestation of the invention-structure equation.[179]

---

175. 927 F.2d 1200 (Fed. Cir. 1991). *Amgen* was decided before the advent of the modern written description requirement, so it might be expressly framed as a written description case today.

176. *Id.* at 1204.

177. *Id.* at 1212–14.

178. *Id.* at 1213 (emphasis added).

179. Thus, for the Federal Circuit to use enablement to curb the reach of functional claims in the software arts, it must identify the metaphorical structure of a software invention—the same task that it must undertake to bring the invention-structure equation to bear on software claims. The merger of enablement and the invention-structure equation in software claims surfaced in *LizardTech, Inc. v. Earth Resource Mapping, Inc.* when the Federal Circuit used the invention-structure equation to invalidate a software claim under both enablement and written description. *See infra* Part IV.B.2.

Wash U Law Repository

III. EXPLAINING SOFTWARE-PATENT OVERBREADTH

This Part identifies the root cause of the problem of software-patent overbreadth. Part III.A argues that software is a purely functional technology in the sense that the structural properties of the software actually generated by an inventor should not define a protectable software invention. Part III.B identifies patent law's functionality malfunction: the invention-structure equation breaks down when it is brought to bear on technologies, like software, that are purely functional.

A. Software Inventions Are Functional All the Way Down

Software is commonly described as exceptional or different from other patentable inventions, and one of the factual premises frequently put forward to justify software exceptionalism is that software is "intangible."[180] Early courts questioning the patentability of software inventions analogized software to the supposedly intangible mental processes that occur within human minds,[181] and the *Freeman-Walter-Abele* test for patentable subject matter labeled software-executed processes as non-physical.[182] More recently, the Federal Circuit used the assumption that software is intangible to question the patentability of software inventions under the machine-or-transformation test.[183]

However, the embodiments of software programs that are capable of infringing software patents are clearly material, worldly entities.[184]

---

180. *See, e.g.*, Richard S. Gruner, *Intangible Inventions: Patentable Subject Matter for an Information Age*, 35 LOY. L.A. L. REV. 355, 357 (2002) ("New designs for software and computer-based business practices . . . . resemble the sorts of intangible ideas and thought processes that have traditionally fallen outside of patent protections.").

181. *See, e.g.*, *In re* Bernhart, 417 F.2d 1395 (C.C.P.A. 1969). The analogy between computer and mental processes has reemerged in the Federal Circuit's recent cases addressing the conditions under which software claims constitute patentable subject matter. *See, e.g.*, Bancorp Servs. v. Sun Life Assurance Co., 687 F.3d 1266, 1277–79 (Fed. Cir. 2012).

182. *See, e.g.*, *In re* Grams, 888 F.2d 835 (Fed. Cir. 1989).

183. *See, e.g.*, *In re* Bilski, 545 F.3d 943 (Fed. Cir. 2008), *aff'd on other grounds sub nom* Bilski v. Kappos, 130 S. Ct. 3218 (2010). Software is also a human-readable text. *See* Arti Rai & James Boyle, *Synthetic Biology: Caught Between Property Rights, the Public Domain, and the Commons*, 5 PLOS BIOLOGY 58 (2007) (describing software as "a machine made of words"). Human-readable texts, in turn, are commonly described as intangible entities because the knowledge that such texts convey has its principal locus in the human mind. *See generally* Kevin Emerson Collins, *Semiotics 101: Taking the Printed Matter Doctrine Seriously*, 85 IND. L.J. 1379 (2010) (casting the printed matter doctrine as a prohibition on the patenting of the human mind).

184. *See* Microsoft Corp. v. AT&T Corp., 550 U.S. 437 (2007) (discussing the physicality of a copy of a software program in the course of assessing whether software can be a "component" under § 271(f)); Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform*, 7 UCLA J.L. & TECH. 1, 8–12 (2003). The definition of what constitutes a

Software does not violate the materialist worldview: it is the physical structure of software loaded onto a computer that endows software with its behavioral capacities.[185] Software exists as electrons or charges on a hard drive or in a computer's memory; a computer implements a software program only because a particular set of gates in the processor is open or closed.[186] While likely intended metaphorically, the statement that "[p]rogram text is, thus, like steel and plastic, a medium in which other works can be created" is literally true in that an operable embodiment of a software program exists only because it is crafted from a tangible medium.[187]

Yet, despite the materiality of embodiments of software inventions, there is a grain of truth buried in the assertion that software is intangible in a way that makes software anomalous among patentable subject matters. Software programs may not lack materiality, but their materiality is irrelevant to identifying, delineating, or defining protectable software inventions. Assume a computer programmer has just invented a new software program. How can the inventor describe or define what he has invented? The programmer would be hard pressed to convey the gist of what he had invented by referring to *any* of the physical, structural properties of the software.

> [F]or all practical purposes the programmer and others who think about and describe the program have no practical choice but to conceive of and describe it in terms of its logical structure [or function]. . . . It is far from clear that it would even be possible for the human mind to appreciate the physical structure of all but the simplest programs or to explain them in terms of their physical structures.[188]

---

software invention is an intangible conceptual type. *Cf. Microsoft*, 500 U.S. at 447–52 (discussing software "in the abstract"). However, the definitions of inventions in all technologies are conceptual types, so this does not make software exceptional.

185. *See supra* note 32 and accompanying text (discussing materialism).

186. Plotkin, *supra* note 184, at 38–39. In fact, patent law has unflinchingly recognized the tangibility of software embodiments in some of its doctrines even as it has denied the tangibility of software in others. For example, a computer programmed with a new software program has long been treated as a new machine under the novelty doctrine. WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1348 (Fed. Cir. 1999); *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc); *In re* Freeman, 573 F.2d 1237, 1247 (C.C.P.A. 1978); *In re* Prater, 415 F.2d 1393, 1403 n.29 (C.C.P.A. 1969).

187. Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2323 (1994).

188. Plotkin, *supra* note 184, at 46 & n.126; *see also id.* at 26 ("The process of computer programming enables a programmer to create a machine that has a particular novel physical structure

Wash U Law Repository

The irrelevance of the physical, structural properties of a software embodiment to the definition of a software program has been engineered into the very nature of software itself at the most fundamental of levels. The core value of software lies in the fact that the design of software that possesses any given set of logical, functional properties need not involve any consideration of the physical properties of the hardware or the distribution of electrical charges therein: "Computers are understandable because you can focus on what is happening at one [functional] level of the hierarchy without worrying about the details of what goes on at the lower [structural] levels."[189] A software program can be implemented in entirely different code in the same programming language or in an entirely different language,[190] and there is no common thread of physical, structural properties that runs through these distinct codings. Thanks to interpreters and compilers, any given program can be implemented on a wide array of different computers, each possessing a different internal architecture and requiring the software to take on different physical, structural properties.[191] In fact, "[p]resent-day computers" on which software programs are executed "are built of transistors and wires, but they could just as well be built, according to the same principles, from valves and water pipes, or from sticks and strings."[192] Furthermore, hardware and software implementations of any given program are functionally interchangeable despite their radically different structural properties.[193]

The behavioral capacities of a computer program—that is, "the actions that a computer can perform by executing program instructions"—are central to the definition of a computer program.[194] Standing alone, however, the importance of the functional properties does not differentiate

for performing a particular function without requiring the programmer to design the novel features of the machine in physical terms."); *id.* at 36 ("[O]ne of computer science's express goals is to ensure that programmers can do their work in complete ignorance of the physical structure of . . . hardware . . . ."); *id.* at 44–45 ("[A] programmer who modifies the physical structure of a computer by providing source code to the computer need not even know that the computer's memory is being physically modified at all, much less understand or appreciate the nature of those physical modifications.") (citations omitted).

189. *See* W. DANIEL HILLIS, THE PATTERN ON THE STONE, at ix (1998).

190. Samuelson et al., *supra* note 187, at 2317.

191. HILLIS, *supra* note 189, at 56–58 (discussing interpreters and compilers); *cf.* Microsoft Corp. v. AT&T Corp., 550 U.S. 437, 450 (2007) ("Software . . . is a stand-alone product developed and marketed 'for use on many different types of computer hardware . . . .'").

192. HILLIS, *supra* note 189, at viii. Software programs running on after-arising hardware always infringe earlier-filed software patents precisely because the radically new structure that the software must take on to run on the after-arising hardware is irrelevant to the definition of what constitutes a software invention. Collins, *Getting into the "Spirit," supra* note 44, at 1265–67.

193. *In re* Alappat, 33 F.3d 1526, 1583 (Fed. Cir. 1994) (Rader, J., concurring).

194. Samuelson et al., *supra* note 187, at 2316–20.

software from other patentable technologies. A pharmaceutical drug is valued by a consumer more directly for what it does (cure a disease) than for its molecular structure; a patient cares that a drug has a particular molecular structure only because that structure has a metabolic function.[195] What is unique about software is not the significance of functionality *per se* but rather the insignificance of physical structure: it is practically impossible to refer to a set of structural characteristics shared by the embodiments of a software invention.[196] It is for this reason that "[b]ehavior is not a secondary by-product of a program, but rather an essential part of what programs are."[197]

In short, software is exceptional. The key to software exceptionalism lies in a weak form of the software-is-intangible argument. Although they exist, the physical, material properties of an embodiment of a software invention are not relevant to what constitutes a protectable software invention or thus to the optimal scope of a software claim. Viewed from the opposite perspective, software is a purely functional technology on all relevant levels of definition.[198] The ontology of invention is thus reversed in the software arts. In most arts, an invention "is" its structure, not its function.[199] In the software arts, however, an invention "is" its function, not its structure. A software invention is function "all the way down."[200]

B. The Functionality Malfunction

There is a fundamental mismatch between the invention-structure equation that patent law has traditionally employed to curtail the scope of functional claims and the purely functional nature of software inventions. The invention-structure equation enables courts to rein in the overbreadth of functional claims. Because protectable inventions are defined by some subset of the structural properties of the technology that an inventor has

195. *See supra* text accompanying note 71–72 (noting the importance of functional properties in market definition).

196. Unique may be an overstatement. *See supra* note 13.

197. Samuelson et al., *supra* note 187, at 2317.

198. A recent commentator arrives at a similar conclusion about the exceptional, functional nature of software inventions. Note, *Everlasting Software*, 125 HARV. L. REV. 1454, 1456 (2012) ("[S]oftware . . . is defined . . . by function itself."). However, the explanation for the functional nature of software is not that "software does not have physical characteristics." *Id.* Rather, the explanation is that software as a technology has been engineered at a deep level to make the physical, structural characteristics of software irrelevant to the definition of a software program. *See supra* text accompanying notes 188–93.

199. *See supra* notes 93–94 and accompanying text.

200. HAWKING, *supra* note 4, at 1.

Wash U Law Repository

generated, purely functional claims are overbroad and invalid.[201] Software inventions, however, are purely functional entities. The material, structural properties of a software program are not relevant to the definition of a protectable software invention or the optimal scope of a software claim.[202] Software inventions therefore lack the metaphorical bolt onto which patent law's conventional scope-restraining doctrinal tools (or "policy levers"[203]) can attach to gain purchase and curtail permissible claim scope in a systematic manner. This is patent law's *functionality malfunction*: the patent doctrines that have traditionally curtailed claim overbreadth break down when they are brought to bear on purely functional technologies. The invention-structure equation simply cannot get a grip on the problem of the overbreadth of functional claims in the software arts as there are no relevant physical, structural properties to grab onto and require as claim limitations.

The functionality malfunction places judges and examiners between a rock (too little protection) and a hard place (too much protection) in the software arts. Enforcing the invention-structure equation by requiring the recitation of physical, structural properties as claim limitations would yield absurdly narrow, economically irrelevant claims. Unclaimed, perfect economic substitutes would abound,[204] and the private value of a software patent to its owner would most likely not be worth the private cost of obtaining the patent. Software patents would wither on the vine. Alternatively, if the courts were to exempt software from the invention-structure equation and sanction purely functional software claims, then the very functional claims that are deemed to be overbroad and invalidated in other arts would be an expected feature of software patents.[205] As early commentators on the patentability of software inventions noted,

> [I]f the Patent and Trademark Office were willing to issue a patent with claims for *any* means of achieving a particular set of results, such a patent would issue at a high level of generality and would inhibit competition in development of useful program behaviors out

201. *See supra* Part II.C.
202. *See supra* Part III.A.
203. BURK & LEMLEY, *supra* note 1, at 109–65.
204. *See supra* notes 64–70 and accompanying text (discussing the relationship between claim scope and market power).
205. *See supra* Parts II.B & C.

> of proportion to the innovation actually contributed by the claimant.[206]

Confronted with this choice between patent protection that is either too hard or too soft, the courts have clearly chosen too soft. Inventors routinely seek, and courts routinely sanction, purely functional software claims. For an anecdotal illustration of the purely functional nature of most contemporary software claims, consider claim 22 of patent number 5,231,670.[207] Claim 22 describes a voice-recognition technology that allows a user to create a text file by uttering both the words that are the substance of the text ("spoken input") and commands that trigger functional responses from the software that one could otherwise trigger by using pull-down menus or keyboard shortcuts ("spoken commands").[208] In its entirety, claim 22 reads:

> 22. A method for editing displayed textual data generated in response to an audio input signal representing spoken input text, said method comprising the steps of:
>
> editing said generated textual data by
>
> receiving audio signals representing spoken input textual data and spoken commands through an input device;
>
> analyzing said audio signals to determine whether said audio signals represent spoken input to be provided to an application or a spoken command for editing textual data.[209]

This language delineates the outer boundary of a software claim by reciting two behaviors—the receiving and analyzing steps—that allow the software to perform the function of editing textual data. This type of functional claim would routinely be invalidated for overbreadth in other

---

206. Samuelson et al., *supra* note 187, at 2345.

207. U.S. Patent No. 5,231,670 (filed Mar. 19, 1992). This claim is used as an example because it is both simple to explain and representative of software patents broadly writ, not because it is unusual or well known. For an example of a high-profile, litigated functional software claim, consider Amazon's infamous patent on a one-click method of checking out during an internet retail transaction. Amazon.com, Inc. v. Barnesandnoble.com, Inc., 239 F.3d 1343 (Fed. Cir. 2001); *see also* Lemley, *supra* note 1, at 16–18 (listing examples of functional claim limitations in litigated software cases).

208. '670 Patent, col. 2 ll. 50–55 ("[T]he system and method . . . process both simple spoken words as well as spoken commands and . . . provide the necessary text generation in response to the spoken words or . . . execute an appropriate function in response to a command.").

209. *Id.* at col. 16 ll. 18–27.

Wash U Law Repository

arts, but the functionality malfunction has made it the norm in the software arts.[210]

The functionality malfunction, and thus the problem of software-patent overbreadth, results from a clash between the legal importance of the structural properties of an invention in patent doctrine and the technological irrelevance of those properties in the software arts. The problem is not simply that courts have failed to bring readily available policy levers to bear on these claims in the economically appropriate manner.[211] Rather, the problem is that the exceptional nature of software as a technology neutralizes the policy lever that is conventionally used to adjust the permissible scope of functional claims.

IV. ALGORITHMS: A VIABLE SOFTWARE-SPECIFIC PATCH?

The functionality malfunction in the software arts explains why the traditional policy lever for curtailing the scope of functional claims, namely the invention-structure equation, has not to date been widely used in the software arts. Yet the inefficacy of the traditional doctrinal tool need not in and of itself doom software patents to a fate of anomalous overbreadth. A *sui generis*, software-specific patch for the functionality malfunction could bring the scope of software patents into line with the scope of patents in other technological arts. This Part considers the most promising of such patches: the adoption of an invention-algorithm equation to curtail the scope of functional claims in the software arts. Part IV.A defines an algorithm as the term is used in computer science. It argues that an invention-algorithm equation offers a technology-specific way of bringing the invention-structure equation to bear on functional claims in the software arts. Part IV.B demonstrates that the Federal Circuit has already taken the first, tentative steps toward recognizing algorithms as the metaphorical structures of software inventions in all three of the doctrines that employ the invention-structure equation. Part IV.C sounds three notes of caution about the difficulties that the Federal Circuit will

210. James Bessen and Michael Meurer argue that the principal effect of the functional nature of software claims may not be claim overbreadth. They argue that it may instead be uncertainty concerning validity if courts act in an expedient, ad hoc manner to narrow or invalidate overbroad software claims. BESSEN & MEURER, *supra* note 1, at 211–12; *cf.* Request for Comments and Notice of Roundtable Events for Partnership for Enhancement of Quality of Software-Related Patents, 78 Fed. Reg. 292, 294 (Jan. 3, 2013) (framing the principal problem created by functional software claims as a problem of indefiniteness and uncertainty). Even if Bessen and Meurer are correct, the functionality malfunction remains the root cause of the costly problem that plagues software patents, but the problem is uncertainty about claim validity rather than overbreadth.

211. *See supra* note 2 and accompanying text.

have to address if the invention-algorithm equation is to be developed into a tool that can curtail the scope of functional software claims in a rigorous, systematic manner.

A. The Invention-Algorithm Equation

In computer science, an algorithm is defined as a "systematic and precise, step-by-step procedure . . . for solving certain kinds of problems or accomplishing a task."[212] Simply put, "the essence of algorithms" is "what to do to perform a task."[213] This is an extremely flexible definition.[214] All that is needed is, first, for a problem or task to be identified and, second, for a step-by-step procedure for solving the problem or accomplishing the task to be specified. There is no single format in which an algorithm must be communicated. Mathematical formulae, prose, and flow charts can all express algorithms.[215]

Although the invention-structure equation curbs permissible claim scope and invalidates purely functional claims in most arts,[216] it cannot curtail functional claiming in the software arts because there are no physical, structural properties that contribute to the identity of a software invention.[217] However, courts could port the invention-structure equation over to the software arts by framing algorithms as the metaphorical structures of software inventions and adopting a technology-specific *invention-algorithm equation*.

Just as the invention-structure equation grounds the identity of the protectable invention in a subset of the structural properties of the technology that an inventor actually produces and discloses, the invention-algorithm equation defines a protectable software invention with reference to some of the algorithmic properties of the software that an inventor

212. DICTIONARY OF COMPUTER SCIENCE, ENGINEERING, AND TECHNOLOGY 13 (Phillip A. Laplante ed., 2000); *see also* MICROSOFT COMPUTER DICTIONARY (5th ed. 2002) ("a finite sequence of steps for solving a logical or mathematical problem or performing a task"); LESLEY ANNE ROBERTSON ET AL., SIMPLE PROGRAM DESIGN: A STEP-BY-STEP APPROACH 271 (3d ed. 2000) ("[Algorithms are a] set of detailed, unambiguous and ordered instructions developed to describe the processes necessary to produce the desired output from given input."); Allen Newell, *Response: The Models Are Broken, The Models Are Broken*, 47 U. PITT. L. REV. 1023, 1024 (1986) ("An algorithm is [a] . . . sequence of steps or operations for solving a class of problems."). The computer scientist's definition of an algorithm is likely more general than the definition that the Supreme Court employed in its software cases. *See infra* Part IV.C.3.

213. Newell, *supra* note 212, at 1026.

214. In fact, its flexibility may be its downfall as a tool for patent law. *See infra* Part IV.C.2.

215. Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1340 (Fed. Cir. 2008).

216. *See supra* Part II.C.

217. *See supra* Part III.A.

Wash U Law Repository

actually produces and discloses. Each functional limitation in a software claim states a problem that needs to be solved or a task that needs to be performed. Any step-by-step procedure that leads to the solution of the problem or the accomplishment of the task is an algorithm. For example, consider again claim 22 of the '670 patent.[218] This claim recites two functional limitations: the "receiving" limitation and the "analyzing" limitation. Each of these limitations specifies a task to be accomplished or, if you are a software engineer who is instructed to write code that can perform these tasks, a problem to be solved. An algorithm—whether in the form of a flow chart or a textual description—lists a series of more specific steps that, if performed in the proper sequence, accomplishes the more general task that is recited as the claim limitation. The invention-algorithm equation would invalidate claims like claim 22 under the commensurability principle.[219] Claim 22 encompasses all software programs that accomplish the claimed tasks of receiving and analyzing. However, under the invention-algorithm equation, the inventor's protectable invention is only a set of algorithms for performing the tasks of receiving and analyzing, rendering claim 22 impermissibly overbroad for the same reason that purely functional claims are overbroad in other arts. The invention-algorithm equation upholds the validity of functional software claims only if those claims are limited in scope to particular algorithms for accomplishing the claimed functions.[220]

In most arts, the physical, structural properties of the technology that an inventor has produced are the answer to "how" question of technology: How does a technology work, or how does it achieve the functional utility that it possesses?[221] In the software arts, a computer scientist would answer this same "how" question not by reciting the physical, structural properties of an embodiment of software but rather by stating an algorithm. Algorithms are still functional entities in that they, too, are composed of functionally defined steps.[222] Yet, because algorithms describe a more specific set of functions for achieving the more general function recited as a claim limitation, courts can frame them as the

218. *See supra* notes 207–09 and accompanying text.

219. *See supra* notes 43–44 and accompanying text (discussing the commensurability principle).

220. The invention-algorithm equation could employ either of the two mechanisms through which the invention-structure equation already operates. It could invalidate overbroad claims, or it could construe them narrowly. *See supra* note 101.

221. *See supra* text accompanying notes 32–33.

222. Functional software claims are unavoidable given that software is function all the way down. *See supra* Part III.A.

metaphorical structure of a software invention,[223] and courts can use them to curtail the scope of functional claims.[224]Additionally, algorithms possess the same many-to-one relationship to functions that physical structures have to functions in other technologies.[225] There are many different physical designs for mousetraps that can catch mice, and a variety of different molecular structures for drugs that can lower cholesterol. Similarly, there is usually an array of algorithms for achieving any given functionally specified task. The many-to-one relationship between algorithms and functionally specified tasks means that later software inventors can aspire to invent-around earlier software claims that are limited by the invention-algorithm equation, just as later inventors in other arts can aspire to invent-around claims that are limited in scope by the invention-structure equation.[226] Other answers to the "how" question often remain available for later innovators to discover, invent, and commercialize without running afoul of the earlier inventor's rights.

In sum, algorithms can be the software analog for the physical, structural properties that define a protectable invention in other arts. They give the invention-structure equation something to latch onto in the purely functional realm of software. When incorporated into the invention-algorithm equation, they offer one route forward for remedying patent law's functionality malfunction in the software arts, resolving the problem of software-patent overbreadth that it has generated and bringing the scope of software patents into line with the scope of patents in other technologies.

One potential objection to the invention-algorithm equation as a mechanism for curtailing claim scope in the software arts is that software inventors deserve anomalously broad, functional claims because a software inventor's "*actual invention* is functionality" whereas other inventors' actual inventions are only structurally defined means of achieving that functionality.[227] However, this distinction does not justify the issuance of anomalously broad claims in the software arts. It is true that a software inventor who thinks up a new behavior for a software program makes a different kind of contribution to technological progress than the inventor of the first drug that has the long-desired behavior of

223. They are "logical structure" rather than physical structure. Plotkin, *supra* note 184, at 26–29.

224. *But see infra* Part IV.C.2 (discussing the difficulties that the functional nature of an algorithm creates for the invention-algorithm equation).

225. *See supra* text accompanying note 34.

226. *See supra* text accompanying note 76.

227. *Everlasting Software*, *supra* note 198, at 1474.

Wash U Law Repository

being able to cure a well-known medical condition does.[228] However, inventors in the non-software arts do sometimes think up or imagine the functional properties that they employ to limit claim scope, and there are sound policy reasons to explain why even these relatively more meritorious inventors cannot today obtain functional claims under the invention-structure equation.[229]

B. The Federal Circuit's Tentative, Uncoordinated Steps

The idea of framing algorithms as the metaphorical structures of software inventions in order to curtail the permissible scope of functional software claims is not novel. As explored below in Part IV.B.1, the Federal Circuit has recently invalidated a significant number of software claims drafted in means-plus-function format because the specifications failed to disclose algorithms for performing the claimed functions.[230] However, what has escaped general notice is that the Federal Circuit has also repeatedly used the invention-algorithm equation to curtail the permissible scope of software patents in cases not involving means-plus-function claims. In a tentative and uncoordinated manner, the Federal Circuit has employed the invention-algorithm equation through all three of the patent doctrines in which the invention-structure equation manifests itself in other arts.[231] Part IV.B.2 considers the Federal Circuit's use of algorithms in the written description requirement, and Part IV.B.3 looks at algorithms in the doctrine of patentable subject matter. Superficially, the developments in these three doctrines may appear to be distinct, as the doctrine addressed in each section employs unrelated terminology and arises from different statutory provisions. Substantively, however, all three

228. *See supra* Part II.B.1.

229. *See supra* Parts II.B.2 & 3 (discussing the normative concerns about claims that map onto markets and the preservation of spillovers). Another possible objection to allowing the invention-algorithm equation to curtail the scope of broad, functional claims in the software arts sounds in enablement. The objection is that "[a]n inventor's description of one object for accomplishing a function in the real world does not enable . . . all objects that accomplish the function" whereas "[o]nce the function [of a software invention] is disclosed" all software embodiments that perform the function are enabled. *Everlasting Software*, *supra* note 198, at 1462, 1464–65. This objection misses the mark because the disclosure of a desired behavior does not enable all means of achieving that behavior, either beyond or within the software arts. Later mechanical inventors will devise nonobvious structures that exhibit an earlier-identified behavior, and later software inventors will devise nonobvious algorithms, programming languages, and computer hardwares that exhibit an earlier-identified behavior.

230. These cases have already been addressed in patent commentary. *See supra* note 15.

231. *See supra* Part II.C (discussing the three distinct doctrinal manifestations of the invention-structure equation).

demonstrate that the Federal Circuit has taken the first, tentative steps toward porting the invention-structure equation over to the software arts in the form of an invention-algorithm equation. Just as the unified nature of the invention-structure equation as a mechanism for curtailing permissible claim scope has to date remained a hidden principle of the deep structure of the patent regime,[232] the wide-spread nature of the Federal Circuit's turn to algorithms in all three of the doctrines that employ the invention-algorithm equation in software cases has also gone unnoticed.

1. Algorithms as Corresponding Structures

Section 112(f) states that claim limitations drafted in means-plus-function format are to be construed in a narrow fashion.[233] The limitations do not encompass all structures capable of achieving the claimed functions. Rather, they encompass only the corresponding structures for achieving those functions disclosed in the specification, as well as the equivalents of those corresponding structures.

Software claims are commonly drafted in means-plus-function format,[234] and the Federal Circuit has recently begun to invalidate means-plus-function software claims for indefiniteness if the patent specification fails to disclose an algorithm for achieving the claimed function.[235] These cases are doctrinally complex, but the take-home lesson is simple. The Federal Circuit has adopted the invention-algorithm equation to limit permissible claim scope. It treats an algorithm as the metaphorical

232. *See supra* notes 97–101 and accompanying text.

233. 35 U.S.C. § 112(f) (2012); *see also supra* notes 123–25 and accompanying text (discussing means-plus-function limitations).

234. In fact, means-plus-function claims were the preferred style of claiming software for many years when the status of software inventions as patentable subject matter under section 101 had not yet been clearly established. *See* LEMLEY ET AL., SOFTWARE AND INTERNET LAW 217–19 (3d ed. 2006); Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CALIF. L. REV. 1, 9–10 (2001). *But cf. infra* note 302 (noting a more recent decline in the use of "means for" language in patent applications).

235. *See* Function Media, L.L.C. v. Google Inc., No. 2012-1020, 2013 WL 516366 (Fed. Cir. Feb. 13, 2013); ePlus, Inc. v. Lawson Software, Inc., 700 F.3d 509 (Fed. Cir. 2012); Noah Sys., Inc., v. Intuit Inc., 675 F.3d 1302 (Fed. Cir. 2012); HTC Corp. v. IPCom GmbH & Co., 667 F.3d 1270 (Fed. Cir. 2012); Dealertrack, Inc. v. Huber, 674 F.3d 1315 (Fed. Cir. 2012); *In re* Aoyama, 656 F.3d 1293 (Fed. Cir. 2011); Rembrandt Data Techs. v. AOL, 641 F.3d 1331 (Fed. Cir. 2011); *In re* Katz Interactive Call Processing Patent Litig., 639 F.3d 1303 (Fed. Cir. 2011); Brown v. Baylor Healthcare Sys., No. 2009-1530, 2010 WL 1838921 (Fed. Cir. May 7, 2010); Net MoneyIN, Inc. v. VeriSign, Inc., 545 F.3d 1359, 1366–67 (Fed. Cir. 2009); Encyclopaedia Britannica, Inc. v. Alpine Elecs., Inc., No. 2009-1087, 2009 WL 4458527 (Fed. Cir. Dec. 4, 2009); Blackboard, Inc. v. Desire2Learn Inc., 574 F.3d 1371, 1383–85 (Fed. Cir. 2008); Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323 (Fed. Cir. 2008); Aristocrat Techs. Austl. Pty. Ltd. v. Int'l Game Tech., 521 F.3d 1328 (Fed. Cir. 2008); *cf.* Keithley v. Homestore.com, Inc., 636 F. Supp. 2d 978, 988–95 (N.D. Cal. 2008).

Wash U Law Repository

structure of a software invention in order to bring the invention-structure equation to bear on software claims.

When the question of what constitutes the corresponding structure of a means-plus-function limitation in a software claim arose, patent owners sought a definition at an extremely high level of generality: a general-purpose computer programmed with software—any software.[236] If corresponding structure could be defined at this high level of generality, then purely functional software claims would be a de facto reality even for claims drafted in means-plus-function format. At the opposite extreme, if the corresponding structure were to be defined at the level of the physical, structural properties of a software embodiment, then software patents would be worthless as a practical matter.[237] In *WMS Gaming, Inc. v. International Game Technologies*, the Federal Circuit eschewed these two extremes and settled on a middle ground: it identified algorithms as the corresponding structures for means-plus-function limitations in software claims.[238] A means-plus-function limitation in a software claim can only encompass software programs that employ the algorithms for achieving the claimed function disclosed by the inventor in his patent specification, as well as equivalent algorithms.

Means-plus-function limitations must be limited to the corresponding structure disclosed in the patent specification (and its equivalents), but what if the patent specification fails to disclose any structure that is capable of performing the claimed function? In this situation, the Federal Circuit holds that the means-plus-function claims are invalid for indefiniteness.[239] The indefiniteness doctrine of section 112(b) holds that claims that employ limitations that have no discernible meaning are invalid.[240] Indefiniteness is a common-sense rule. The scope of a claim to a "thingamajig" cannot be ascertained, so there are many instances in which neither the validity nor infringement analyses can proceed. In the context of means-plus-function limitations, section 112(f) states that the meaning of the claim language is the corresponding structure in the disclosure and its equivalents. If there is no corresponding structure in the disclosure to be found, the means-plus-function limitation has no discernible meaning and is thus invalid for indefiniteness.[241]

---

236. WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1347–50 (Fed. Cir. 1999).
237. *See supra* text accompanying note 204.
238. *WMS Gaming*, 184 F.3d at 1349; *see also* Harris Corp. v. Ericsson Inc., 417 F.3d 1241, 1253 (Fed. Cir. 2005).
239. *In re* Donaldson Co., 16 F.3d 1189, 1195 (Fed. Cir. 1994) (en banc).
240. 35 U.S.C. § 112(b) (2012).
241. *Donaldson*, 16 F.3d at 1195.

Despite the technical nature of the rejection as an indefiniteness rejection, it is important to recognize that the root of the problem when there is no corresponding structure remains overbreadth. There is nothing inherently unclear about a claim defined with functional language. For example, the rules of means-plus-function claiming aside, a claim to any device that uses electromagnetism to communicate intelligible characters at a distance is not like a claim to a thingamajig. In fact, as the Supreme Court flatly stated in *O'Reilly v. Morse*: "It is impossible to misunderstand the extent of this claim."[242] The functional language in *Morse* has a clearly discernible meaning: it describes, and thus encompasses, every means for performing the specified function. One can readily determine whether any particular machine falls within or beyond its scope. The indefiniteness problem arises only because of the statutorily specified rules of claim construction for means-plus-function limitations, and these rules, in turn, exist only because of the Supreme Court's cases from the early twentieth century that invalidate purely functional claims for overbreadth.[243] Thus, it is misleading to say, as patent commentators often do, that "[t]he purpose of section 112(f)—and thus the purpose of requiring an algorithm to support functional software claims—is to ensure 'adequate defining structure to render the *bounds of the claim* understandable to one of ordinary skill in the art.'"[244] The purpose of 112(f) is to reign in permissible claim scope, not to remedy some (nonexistent) indefiniteness problem that is inherent in all functional language.[245] Indefiniteness

242. 56 U.S. (15 How.) 62, 112 (1853).

243. *See supra* Part II.C.2.

244. Comments of Google Inc., In re Request for Comments and Notice of Roundtable Events for Partnership for Enhancement of Quality of Software-Related Patents, at *17 (Apr. 15, 2013), http://www.uspto.gov/patents/law/comments/sw-e_google_20130415.pdf.

245. The functional nature of claim language does sometimes create an unusual indefiniteness problem. However, the root of the problem in these situations is not that the meaning of the claim language is unclear. Rather, the problem is that testing embodiments of a technology to determine whether they possess the claimed behavior is extremely laborious. For example, in *Enzo Biochem, Inc. v. Applera Corp.*, 599 F.3d 1325 (Fed. Cir. 2010), the Federal Circuit considered, but rejected, a defendant's argument that a functional claim limitation was indefinite. The claim described a linkage group in a DNA strand that did not interfere substantially with the ability of the DNA strand to hybridize with other DNA strands. The defendant argued that the hybridization limitation was indefinite due to the need to test each and every combination of a linkage group and a strand to see if it meets the limitation: "Because even a minor alteration of a single nucleotide may have profound effects on the ability of a DNA strand to hybridize, depending on the length and sequence of the strand, [the defendant] argues that identical linkage groups may cause interference in some strands but not in others . . . ." *Id.* at 1332. Consider also a claim to a drug-coated stent with the following functional limitation: "'wherein said [stent] provides an in-stent late loss in diameter at 12 months following implantation in a human of less than about 0.5 mm.'" Bos. Scientific Corp. v. Johnson & Johnson, 647 F.3d 1353, 1368 (Fed. Cir. 2011). The meaning of the functional claim language is perfectly clear, but potential defendants cannot know whether a stent infringes the claim unless they

Wash U Law Repository

problems in 112(f) claims are second-order problems. They are artifacts of patent applicants' failure to provide the data needed to construe a functional claim narrowly according to the rules of 112(f). But for the need to rein in claim scope, functional claims would not have routine indefiniteness problems.

In its 2008 opinion in *Aristocrat Technologies Australia v. International Game Technology*,[246] the Federal Circuit invalidated a software claim drafted with a means-plus-function limitation for indefiniteness because there was no algorithm disclosed in the patent specification.[247] *Aristocrat Technologies* involved a patent on "an electronic slot machine that allows a player to select winning combinations of symbol positions."[248] The patent claimed the electronic guts of the slot machine, with a limitation reciting a "game control means" that performed the functions of controlling the images displayed to the player, paying a prize when the predetermined combination of symbols shows up, and defining the pay lines.[249] The Federal Circuit held that the "game control means" limitation was a means-plus-function limitation and that the specification only described "pictorial and mathematical ways of describing the claimed function of the game control means," not an algorithm that specified a step-by-step procedure for accomplishing the claimed function.[250] The court invalidated the claim "to avoid pure functional claiming . . . 'unbounded by any reference to structure in the specification.'"[251]

---

implant the stent into a human, wait twelve months, and examine the device. In these situations, functional claims create a public notice problem that is not caused by a lack of clarity in the meaning of functional language.

246. 521 F.3d 1328 (Fed. Cir. 2008).

247. *Id.* at 1332–38. The intersection of indefiniteness and means-plus-function claims for software inventions had been raised in earlier cases, but the Federal Circuit had previously applied a lax standard and upheld software claims with means-plus-function limitations even when the disclosure did not specify an algorithm for performing the claimed function. *See, e.g.*, *In re* Dossel, 115 F.3d 942, 946 (Fed. Cir. 1997) ("While the written description does not disclose exactly what mathematical algorithm will be used to compute the end result, it does state that 'known algorithms' can be used to solve standard equations which are known in the art.").

248. *Aristocrat Techs.*, 521 F.3d at 1330.

249. *Id.* at 1331.

250. *Id.* at 1334–35; *see also id.* at 1334 (noting that an equation disclosed in the specification "is not an algorithm that describes how the function is performed, but is merely a mathematical expression that describes the outcome of performing the function"). The argument that the specification simply restates the claimed functions is common in the Federal Circuit's algorithm cases. *See, e.g.*, Encyclopaedia Britannica, Inc. v. Alpine Elecs., Inc., No. 2009-1087, 2009 WL 4458527, at * 5 (Fed. Cir. Dec. 4, 2009) (addressing a "one-step algorithm" that "is simply a recitation of the claimed function").

251. *Aristocrat Techs.*, 521 F.3d at 1333 (quoting Med. Instrumentation & Diagnostics Corp. v. Elekta AB, 344 F.3d 1205, 1211 (Fed. Cir. 2003)).

Aristocrat Technologies has led to a wave of cases in which the Federal Circuit has invalidated software claims drafted with means-plus-function limitations because the specification fails to disclose an algorithm.[252] This is the invention-algorithm equation at work in the software arts, with algorithms framed as the metaphorical structure of software inventions. The inventor's protectable invention is a set of algorithms for achieving a specified function. Claims that reach beyond those algorithms violate the commensurability principle and are thus invalid for overbreadth.

2. Algorithms as Possessed Inventions

To date, the Federal Circuit has not considered many invalidity arguments based on the written description doctrine in the software arts.[253] However, the Federal Circuit's one high-profile software case that uses the written description requirement to curtail an overbroad claim—*LizardTech Inc. v. Earth Resource Mapping, Inc.*[254]—clearly employs the invention-algorithm equation. It frames algorithms as the metaphorical structures of software inventions and tethers permissible claim scope to the algorithms disclosed in the specification.

LizardTech involved a patent on technology for compressing large digital images.[255] The prior art broke the large image into discrete tiles and compressed the tiles individually, but this procedure resulted in seams or boundary effects in the image when the picture was displayed.[256] The patented technology overcame this shortcoming and compressed the tiles of a large image in a manner that yielded a "seamless" image.[257] The Federal Circuit addressed two nested claims. The broader claim encompassed any method of compressing the tiles of a large image so as to create a seamless compression file, and the narrower claim described a method of achieving this end that involved "maintaining updated sums."[258]

---

252. *See supra* note 235. Determining whether the specification discloses a step-by-step algorithm for performing the claimed function, or whether the specification repeats the functions already listed as claim limitations, has proven difficult. *Compare In re* Aoyama, 656 F.3d 1293 (Fed. Cir. 2011) (no algorithm disclosed), *with* Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376 (Fed. Cir. 2011) (algorithm disclosed).

253. Most written description arguments have involved the biochemical arts. *See supra* note 128.

254. 424 F.3d 1336 (Fed. Cir. 2005); *see also* Int'l Automated Sys., Inc. v. Digital Persona, Inc., 565 F. Supp. 2d 1276 (D. Utah 2008) (applying *LizardTech* to a software invention).

255. For a detailed discussion of the technology and claims at issue in *LizardTech*, see Merges, *supra* note 144, at 1657–72.

256. *LizardTech*, 424 F.3d at 1339.

257. *Id.*

258. *Id.* at 1340.

Wash U Law Repository

The allegedly infringing software technology did not maintain updated sums, so it infringed the broader claim, but not the narrower claim.[259]

The plaintiff's victory on infringement of the broader claim, however, was Pyrrhic as the Federal Circuit invalidated that claim under the written description doctrine. The Federal Circuit held that the specification disclosed a particular algorithm for performing the compression process that included the step of maintaining updated sums, and it invalidated the broad claim because it was not limited to methods employing that algorithm.[260] In other words, "a specific algorithm was recited in the patent specification [but] the asserted claim had been broadened (by dropping a limiting feature present in the algorithm described in the specification)" and was therefore invalid for overbreadth.[261] Just as the functional claims in *Eli Lilly* were too broad because they reached too far beyond the structure of the DNA molecules disclosed in the specification,[262] the functional claims in *LizardTech* were too broad because they reached too far beyond the algorithm for achieving the claimed function disclosed in the specification. This, again, is the invention-algorithm equation at work in the software arts, with an algorithm identified as the metaphorical structure of a software invention.

3. Algorithms as Indicators of Particular Machines

In its 2010 opinion in *Bilski v. Kappos*,[263] the Supreme Court reaffirmed that claims to abstract ideas do not describe patentable subject matter under section 101. In *Bilski*'s aftermath, the Federal Circuit has been grappling with the difficult task of distinguishing unpatentable software claims that describe abstract ideas from patentable software claims that do not.[264] One theme in these post-*Bilski* software cases is that

259. *Id.* at 1340–41.

260. *Id.* at 1342, 1343. The Federal Circuit also held that the claim was invalid for lack of enablement, but enablement had not been addressed by the district court. *Cf. supra* note 179.

261. Merges, *supra* note 144, at 1665; *see also* PTO 112 Guidelines, *supra* note 122, at 7170–71 (citing *LizardTech* for the proposition that the written description requirement looks to the algorithms disclosed in the specification to restrict the permissible scope of software claims).

262. *See supra* notes 131–35 and accompanying text.

263. 130 S. Ct. 3218 (2010).

264. The Federal Circuit initially issued a series of panel opinions with seemingly contradictory holdings. *Compare* Ultramercial, LLC v. Hulu, LLC, 657 F.3d 1323 (Fed. Cir. 2011), *vacated by* Wild Tangent, Inc. v. Ultramercial, LLC, 132 S. Ct. 2431 (2012) (holding an internet-based business method to be patentable subject matter), *with* Dealertrack, Inc. v. Huber, 674 F.3d 1315 (Fed. Cir. 2012) (holding an internet-based business method to be unpatentable subject matter). It then took the issue en banc and authored a highly fractured opinion. CLS Bank Int'l v. Alice Corp. Pty., 717 F.3d 1269 (Fed. Cir. 2013) (en banc).

an algorithm limitation is a sign of a software claim to a particular machine and, in turn, a particular machine is one antipode of an abstract idea.[265] If the Federal Circuit pursues this theme, it will be adopting the invention-algorithm equation. It will be framing algorithms as the metaphorical structures of software inventions in its section 101 analysis and using the invention-algorithm equation to invalidate functional claims that are not limited to particular structures for achieving claimed functions. In other words, the Federal Circuit will be developing its contemporary prohibition on claims to abstract ideas in the software arts on the model of the Supreme Court's nineteenth century prohibition on claims to principles in the mechanical arts in cases like *Wyeth* and *O'Reilly*: in both situations, the courts uphold functional claims only if they are tied to the structural properties of a particular machine, whether literal or metaphorical.[266]

For at least a decade before *Bilski*, the Federal Circuit had been treating the patentable subject matter requirement of section 101 as a mere formality in the software arts and using an extremely expansive test for patentable subject matter.[267] The Federal Circuit's 2008 opinion in *In re Bilski* represented a significant shift insofar as it adopted a test for patentable subject matter with more bite: the machine-or-transformation test.[268] Under the machine-or-transformation test, a method claim describes a statutory "process" under section 101 only if the method is either tied to a particular machine or responsible for transforming an article into a different state or thing.[269] The Federal Circuit used its newly minted test to invalidate claims that could be infringed by human execution of a set of legal contracts entirely unaided by machines of any

265. There are several distinct definitions of an abstract idea, so there are several distinct antipodes. *See* Kevin Emerson Collins, Bilski *and the Ambiguity of "an Unpatentable Abstract Idea,"* 15 LEWIS & CLARK L. REV. 37 (2011).

266. *See supra* Part II.C.1. Importantly, the requirement that a machine of any kind be used to infringe the claims is not a sufficient condition for patentable subject matter. The *Wyeth* claim was clearly tied to machinery, as it claimed the cutting of ice by power other than human power. Similarly, the *Morse* claim presumptively could not be infringed without using a machine of some kind. However, neither claim was tied to any *particular* machine with a specified set of structural properties. Similarly, under the invention-algorithm equation, a patentable claim to a programmed computer must be tied to a computer that executes a particular algorithm, not simply a computer that accomplishes a functionally specified result.

267. State St. Bank & Trust Co. v. Signature Fin. Grp., Inc., 149 F.3d 1368 (Fed. Cir. 1998) (restating the useful, concrete, and tangible results test).

268. *In re* Bilski, 545 F.3d 943 (Fed. Cir. 2008) (en banc), *aff'd on other grounds sub nom.* Bilski v. Kappos, 130 S. Ct. 3218 (2010).

269. *Id.* at 954.

Wash U Law Repository

kind (except pencils and sheets of paper, perhaps) and that were therefore clearly not tied to a particular machine.[270]

The Supreme Court affirmed the Federal Circuit's holding in *Bilski*, but it used different reasoning to justify its conclusion.[271] The Court agreed with the Federal Circuit in general terms that the section 101 restriction on patentable subject matter was more than a formality, but it held that its earlier (and difficult to parse) opinions in *Gottschalk v. Benson*,[272] *Parker v. Flook*,[273] and *Diamond v. Diehr*[274] provided the ultimate test for patentable subject matter. Drawing its rhetoric from these cases, but not providing much in the way of a reasoned analysis to explain its conclusion, the Court labeled the *Bilski* claims as impermissible attempts to patent an abstract idea.[275] Importantly, however, the Court did not dismiss the machine-or-transformation test as irrelevant to the identification of patentable subject matter. Rather, the Court stated that the machine-or-transformation test may often serve as a "useful and important clue" and "an investigative tool" for determining whether a claim impermissibly describes an abstract idea.[276]

The *Bilski* claims were not technically software claims. They could be infringed either with or without the use of a programmed computer. The implications of *Bilski* for true software claims are therefore only now being addressed as a matter of first impression in the Federal Circuit. Whether analyzed under the machine-or-transformation test or the prohibition on patenting abstract ideas, the Federal Circuit has repeatedly rejected the argument that software claims describe particular machines merely because they are tied to a general purpose computer that exhibits a specified behavior.[277] Precisely what additional limitations are required to transform a computer-executed abstract idea into a patentable computer-technology invention remains hazy, but algorithm limitations are emerging

270. *Id. at* 963–64. Nor were the claims responsible for transforming an article into a different state or thing. *Id.*

271. 130 S. Ct. 3218 (2010).

272. 409 U.S. 63 (1972).

273. 437 U.S. 584 (1978).

274. 450 U.S. 175 (1981).

275. *Bilski*, 130 S. Ct. at 3227, 3230.

276. *Id.* at 3225–27.

277. *See, e.g.*, CLS Bank Int'l v. Alice Corp. Pty., 717 F.3d 1269, 1286 (Fed. Cir. 2013) (en banc); Bancorp Servs. v. Sun Life Assurance Co., 687 F.3d 1266, 1278 (Fed. Cir. 2012); Dealertrack, Inc. v. Huber, 674 F.3d 1315, 1333 (Fed. Cir. 2012); Fort Props., Inc. v. Am. Master Lease LLC, 671 F.3d 1317, 1323 (Fed. Cir. 2012); Fuzzysharp Techs. Inc. v. 3DLabs Inc., 447 F. App'x 182, 185–86 (Fed. Cir. 2011). *But see infra* note 284.

as an important indicator of software claims that describe a "particular machine" and thus that recite patentable subject matter.[278]

For example, in *Dealertrack, Inc. v. Huber*, the Federal Circuit held that a claim to a computer-aided method of selling bonds described an abstract idea and thus was not patentable subject matter.[279] To support its conclusion, the court noted that, although the specification disclosed several algorithms for achieving the claimed function, the scope of the claims themselves was "construed not to be limited to any particular algorithm" for selling bonds.[280] The negative implication is that the Federal Circuit would have looked more favorably on the claim if it had recited an algorithm limitation. Similarly, in *CyberSource Corp. v. Retail Decisions, Inc.*,[281] the Federal Circuit addressed whether a claim to a method of fraud detection was patentable subject matter. It invoked the fact that the claim did "not limit its scope to any particular . . . algorithm" for detecting fraud to bolster its conclusion that the claim described an unpatentable abstract idea.[282] In sum, both *Dealertrack* and *CyberSource* explicitly consider the invention-algorithm equation when determining if a software claim is too broad to be patentable subject matter.

In other post-*Bilski* software cases addressing the reach of patentable subject matter, the Federal Circuit discusses the substance of the invention-algorithm equation without employing the term "algorithm." For example, in *Ultramercial, Inc. v. Hulu, LLC*, the majority opinion reasoned that patentable subject matter:

> focuses on whether the claims tie the otherwise abstract idea to a *specific way* of doing something with a computer, or a *specific computer* for doing something; if so, they likely will be patent eligible. On the other hand, claims directed to *nothing more than the*

278. Algorithm limitations can weigh in favor of patentability regardless of the rhetorical framework in which the doctrine of patentable subject matter is couched. Talk of particular machines derives from the machine-or-transformation test, but algorithms may also demonstrate the absence of preemption, Gottschalk v. Benson, 409 U.S. 63, 71–72 (1972), the existence of an application of an abstract idea, *Diehr*, 450 U.S. at 187, the inclusion of a meaningful limitation on claim scope, *CLS Bank*, 717 F.3d at 1281, 1286, or the presence of enough in addition to an abstract idea to make a claim patentable subject matter, Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1297 (2012).

279. *Dealertrack*, 647 F.3d 1315.

280. *Id.* at 1334.

281. 654 F.3d 1366 (Fed. Cir. 2011).

282. *Id.* at 1372.

Wash U Law Repository

> *idea* of doing that thing on a computer are likely to face larger problems.[283]

Substituting "an algorithm for performing a function" for "a *specific way* of doing something" and "the function itself" for "*the idea* of doing that thing," *Ultramercial* provides a succinct articulation of the invention-algorithm equation.[284]

C. Are Algorithms a Patent Medicine?

Patent law's functionality malfunction has generated the problem of software-patent overbreadth,[285] and a continuation of the Federal Circuit's uncoordinated, tentative steps toward the adoption of an invention-algorithm equation offers one potential solution to the problem.[286] In fact, short of a more radical move such as abandoning the peripheral claiming regime in the software arts[287] or categorically excluding software from patentable subject matter,[288] the invention-algorithm equation may represent the most promising route forward for incremental, judicial reform of software-patent scope.[289]

However, it is far from clear that algorithms represent a real cure for the ills of software patents rather than a patent medicine. The remainder of this section addresses three reasons for doubting that the invention-algorithm equation will put software patents on par with patents in other technologies. Part IV.C.1 addresses a threshold concern that could, with concerted effort, be overcome: the Federal Circuit must use the invention-structure equation in a uniform manner, regardless of the style in which

---

283. No. 2010-1544, 2013 WL 3111404, at *13 (Fed. Cir. June 21, 2013); *see also CLS Bank*, 717 F.3d at 1302 (Rader, J., concurring in part and dissenting in part) (employing similar language).

284. However, there are also themes within these same post-*Bilski* software cases that undermine the invention-algorithm equation. For example, the Federal Circuit repeatedly cites its twenty-year-old opinion in *In re Alappat*, 33 F.3d 1526 (Fed.Cir.1994) (en banc), to suggest that any computer programmed to implement a new process is a special-purpose computer that is patentable subject matter. *Ultramercial*, 2013 WL 3111404, at *13, *16; *CLS Bank*, 717 F.3d at 1302 (Rader, J., concurring in part and dissenting in part). Additionally, the disclosure of algorithms for performing the claimed functions in the specification sometimes appears to weigh in favor of patentable subject matter even when it is unclear that the algorithms are limitations on claim scope. *Ultramercial*, 2013 WL 3111404, at *15.

285. *See supra* Part III.B.

286. *See supra* Parts IV.A & B.

287. A central claiming regime would presumptively allow courts to tailor claim scope to an inventor's contribution on the fly during infringement proceedings. BURK & LEMLEY, *supra* note 1, at 158, 160; *cf.* Burk & Lemley, *supra* note 25 (discussing the merits of a shift to central claiming in all arts).

288. *See supra* note 21 and accompanying text.

289. Mark Lemley has recently made precisely this argument. Lemley, *supra* note 1.

software claims are drafted or the doctrine through which it is applied. Part IV.C.2 examines the deeper and more intractable problem that follows from the infinitely recursive definition of an algorithm. Part IV.C.3 notes a concern that derives from the Supreme Court's identification of algorithms in the abstract as unpatentable subject matter in *Gottschalk v. Benson*.[290]

1. Uniformity

If algorithms are to rectify the functionality malfunction in the software arts, the invention-algorithm equation must be applied to software claims in a consistent, uniform manner. It must limit the scope of functional limitations in software claims regardless of the style in which the claims are drafted. Today, the Federal Circuit is not close to achieving this goal. The lack of consistency exists both within the Federal Circuit's rules governing means-plus-function limitations and between the three doctrines that employ the invention-structure equation.

In theory, the rules of means-plus-function claiming codified in section 112(f) should apply whenever a claim recites functional limitations that fail to recite a sufficient quantum of the structural properties of the technology that an inventor actually produces and discloses. This should be a substantive threshold that is directly tied to the Supreme Court's holding in *Halliburton* and the prevention of claim overbreadth.[291] However, the Federal Circuit has interpreted section 112(f) in a formalistic manner, allowing patent applicants to evade its scope-restricting rule of claim construction simply by altering the style in which a claim is drafted.

Initially, it is unclear how—or even if—the scope-restricting rules of section 112(f) will govern software claims drafted as method rather than apparatus claims. The text of section 112(f) clearly applies to method claims. It requires the scope of limitations in method claims reciting a "step" for accomplishing "a specified function" to be limited to the "acts described in the specification and equivalents thereof."[292] Yet, by the late 1990s, the courts had barely even discussed the application of step-plus-function claiming to method claims.[293] In a 1999 concurring opinion in

290. 409 U.S. 63 (1972).
291. *See supra* Part II.C.2.
292. 35 U.S.C. § 112(f) (2012).
293. Brad A. Schepers, Note, *Interpretation of Patent Process Claims in Light of the Narrowing Effect of 35 U.S.C. § 112(6)*, 31 IND. L. REV. 1133, 1162 (1998) ("It is nothing less than remarkable that four and one-half decades have passed without the emergence of judicial guidance concerning how (or if) paragraph six applies to process or method claims.").

Wash U Law Repository

Seal-Flex, Inc. v. Athletic Track & Court Construction,[294] the Federal Circuit finally broached the topic, noting generically that a limitation in a method claim that recites a "function" without sufficient supporting "acts" for performing the function should be limited in scope to the acts disclosed in the specification and their equivalents.[295] To date, the Federal Circuit has still never identified a step in a method claim as a step-plus-function limitation.[296] For example, the Federal Circuit would be unlikely to subject claim 22 of the '670 patent to the scope-narrowing rules of section 112(f), despite the fact that it is a purely functional claim.[297] If the rules of means-plus-function claiming are to enforce the invention-algorithm equation and provide a meaningful limit on the scope of software claims, the Federal Circuit must extend them to encompass method claims. Apparatus and method claims can always be drawn to the same software invention, and there is no persuasive reason to draw a categorical distinction between them in terms of permissible claim scope.

Furthermore, even within the realm of apparatus claims, the Federal Circuit's formalistic threshold rules for determining whether a functional claim limitation is a means-plus-function limitation lead to disparate treatment of similar claims. If a claim uses the term "means for," the Federal Circuit employs a presumption that section 112(f) applies that can be rebutted if sufficient structural limitations on the means are also present.[298] Inversely, if a claim does not use that term, there is a presumption that section 112(f) does not apply that can be rebutted if the claim fails to recite sufficient structural limitations.[299] If these twin presumptions could be readily rebutted, the Federal Circuit's threshold rules for identifying means-plus-function claims would govern most functional apparatus claims. However, the Federal Circuit interprets each

294. 172 F.3d 836 (Fed. Cir. 1999) (Rader, J., concurring).

295. *Id.* at 847–51.

296. That is, to the best of the author's knowledge. *Cf.* Zimmeck, *supra* note 15, 180 n.81 (collecting cases). In part, the problem is simply the Federal Circuit's presumptions. The Federal Circuit has held that the absence "step for" language creates a presumption against the applicability of section 112(f). *Seal-Flex*, 172 F.3d at 850 (Rader, J., concurring). There is no reason why a patent drafter would opt into the scope-restricting rules of 112(f) if he can obtain broader, valid claims by not opting in. In part, however, the Federal Circuit's reluctance to use section 112(f) to construe method claims likely stems from a deeper problem. Method claims are often functional all the way down, and employing 112(f) to limit the scope of method claims would raise the intractable problem of infinite recursion, just as it does in software apparatus claims. *See infra* Part IV.C.2.

297. *See supra* notes 207–09 and accompanying text. However, the same claim redrafted as an apparatus claim that used the language "means for" would be subject to 112(f).

298. TriMed, Inc. v. Stryker Corp., 514 F.3d 1256, 1259 (Fed. Cir. 2008).

299. Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1358 (Fed. Cir. 2004).

of its presumptions as "a strong one that is not readily overcome."[300] When apparatus claims to software inventions do not use the term "means for," the court often finds sufficient structure in claim limitations to preclude the application of section 112(f) even when the claim is in effect a purely functional claim.[301] In fact, the Federal Circuit has recently stated that it is up to the patent drafter to signal whether he has elected to invoke the rules of means-plus-function claiming, suggesting that patent drafters can opt out of the scope-restricting provisions of section 112(f) whenever they choose to do so.[302] As a result, many purely functional software apparatus claims are never narrowed under section 112(f).

The formalistic rules governing means-plus-function claims would not matter at the end of the day if the other doctrines that employ the invention-algorithm equation were to impose the same restrictions on overbroad functional claims that section 112(f) does. That is, if the invention-structure equation were to be recognized as a unitary design principle of patent law, then the particular doctrine through which the scope-restricting mechanism was administered would be irrelevant. Courts could use the written description doctrine or the doctrine of patentable subject matter to invalidate the overbroad, functional claims that evade the scope-constraining rules of means-plus-function claiming. However, the Federal Circuit is far from achieving this goal, and, because of the inherent difficulties with identifying an algorithm discussed in the following section, it is unlikely to achieve this goal anytime in the near future.

2. Recursion

Algorithms are recursive entities: algorithms have sub-algorithms, which have sub-sub-algorithms, etc. Algorithms have two qualities that make them recursive. First, they specify a way of achieving a task that consists of listing yet more tasks that need to be achieved. An algorithm

---

300. *Id.*

301. For example, the term "programmed logic circuit" was held to have sufficient structure to prevent the application of 112(f). Apex Inc. v. Raritan Computer, Inc., 325 F.3d 1364 (Fed. Cir. 2003) (holding that "circuit" denotes sufficient structure); *see also* LG Electronics, Inc. v. Bizcom Electronics, Inc., 453 F.3d 1364, 1372 (Fed. Cir. 2006), *rev'd on other grounds sub nom.* Quanta Computer, Inc. v. LG Electronics, 553 U.S. 617 (2008) (holding that a CPU and a partitioned memory system were sufficient structure to place a claim limitation reciting "a control unit" beyond the purview of 112(f)).

302. Flo Healthcare Solutions, LLC v. Kappos, 697 F.3d 1367, 1372 (Fed. Cir. 2012). Across all technologies, there is empirical evidence suggesting that patent drafters are trying to opt out of 112(f). Dennis Crouch, *Means Plus Function Claiming*, PATENTLYO (Jan. 14, 2013), http://www.patentlyo.com/patent/2013/01/means-plus-function-claiming.html (noting the decline in patent applications using "means for" language).

Wash U Law Repository

specifies a step-by-step procedure for solving a problem or accomplishing a task,[303] but each of the steps of the procedure is in and of itself simply the statement of a problem that needs solving or a task that needs achieving. Again, software is functional all the way down.[304] Second, algorithms can be formulated at many different levels of abstraction.[305] Together, these two qualities make algorithms infinitely recursive. An algorithm is a step-by-step procedure for solving a problem, but each step of the procedure is a more specific problem in need of solving for which an algorithm can be specified. Sub-algorithms with more precise step-by-step procedures exist for each of the steps of an algorithm.[306]

The recursive nature of algorithms creates a difficulty for courts seeking to frame algorithms as the metaphorical structures of software inventions and to curtail permissible claim scope with the invention-algorithm equation. Regardless of the specificity of the functional limitation, it is always possible to demand greater specificity in the form of an algorithm for performing that function. Assume a software claim with a limitation that recites Function A. Under the invention-algorithm equation, the claim must be either invalidated as overbroad or limited to one or more algorithms for performing the claimed function. Assume that the specification discloses an embodiment of a software program for performing Function A that serially performs steps A_1, A_2, and A_3 and that those steps are an algorithm for Function A. The invention-algorithm equation requires that the claim encompass only software programs that use this algorithm (and perhaps is equivalents) to perform Function A. So far, the result makes perfect sense. But, what if the claim had originally been drafted in a narrower manner to expressly recite steps A_1, A_2, and A_3 as limitations? When recited as claim limitations, these three steps are indistinguishable from three functional limitations that specify problems to be solved or tasks to be achieved. Steps A_1, A_2, and A_3 thus must be conceptualized as Functions A_1, A_2, and A_3. Under a rote application of the invention-structure equation, each of these functional limitations must

303. *See supra* notes 212–15 and accompanying text.

304. *See supra* Part III.A.

305. John Swinson, *Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection*, 5 HARV. J.L. & TECH. 145, 146–50 (1991). Copyright law assumes that algorithms describe software programs at one particular level of abstraction. Gates Rubber Co. v. Bando Chem. Indus., 9 F.3d 823, 834–35 (10th Cir. 1993). Whether or not this is a reasonable assumption in copyright law, it is clearly an oversimplification in patent law.

306. Swinson, *supra* note 305, at 148 ("There is a continuum between the high-level description of the solution to the problem and the low-level machine code. The only change is the detail of expression.").

be limited in scope to a particular algorithm—perhaps the algorithms consisting of the steps A_{1a}, A_{1b}, and A_{1c}; A_{2a}, A_{2b}, and A_{2c}; and A_{3a}, A_{3b}, and A_{3c} (assuming each step of the original algorithm is performed through a three-step sub-algorithm). This result is odd, as it makes the permissible scope of a patent contingent on the form in which the claim is drafted. An overbroad initial claim gets transformed into a permissible, narrower claim, but an initial claim that is identical in scope to the permissible, narrower claim is overbroad.

And, of course, the problem does not end here. The recursion is infinite. If an inventor were to recite any of the steps of a sub-algorithm as steps in a claim, they, too, would be purely functional claim limitations, and they, too, would have to be limited to particular algorithms. Embracing algorithms as metaphorical structure in a rote fashion leads down the rabbit hole. It creates a patent regime that makes the permissible breadth of a claim depend entirely upon the level of specificity at which claim limitations are initially drawn and that bizarrely penalizes patent applicants who initially file claims that are more modest in scope.

In most arts, the invention-structure equation does not lead to infinite regress.[307] A technology's structural properties are not its functional properties at a lower level of generality. Rather, structural properties are categorically distinct from functional properties.[308] Furthermore, the distinction is, in most cases at least, an intuitive one that judges and examiners can readily grasp: what things are as a matter of structure is distinct from what things do as a matter of their function. However, the translation of the invention-structure equation into the invention-algorithm equation erases the intuitive distinction. There is no inherent difference between a "goal" and a step in an algorithm for achieving that goal or a "function" and a step in the process for implementing that function.[309] It is all a matter of context. A goal becomes a step in an implementation process simply by juxtaposing it with a more general goal, and a step in an implementation process becomes a functional goal by bracketing the more general function and removing it from consideration.[310]

---

307. *But see supra* note 296 (noting the recursion problem in step-plus-function method claims).

308. *See supra* notes 28–30 and accompanying text.

309. *But see* Lemley, *supra* note 1, at 3, 39 (articulating these distinctions and relying on them when formulating a proposal to use algorithms to curtail the scope of functional claims).

310. In a submission to the PTO on how to apply section 112(f) to functional software claims, Google argues that the PHOSITA can distinguish a functional description of software that describes an algorithm from one that does not. Google thereby implies that there is no problem of infinite recursion. Comments of Google Inc., *supra* note 244, at *14–*21. However, the submission does not attempt to

Wash U Law Repository

To avoid the software-specific problem of infinite regress, the Federal Circuit cannot employ the invention-algorithm equation in a rote manner as it does today. It cannot limit all functional limitations in software claims to algorithms for performing those functions. It needs to identify a bottom as a matter of policy—a level of generality below which a functional property of a software program counts as metaphorical structure, regardless of whether the property is recited as a limitation in a claim or as a step of an algorithm disclosed in a patent specification that is read into a claim through 112(f). This undertaking would initially require consultation with computer scientists to create a taxonomy of a variety of levels of abstraction at which the functional properties of a software program can be formulated.[311] Then, as a determination that is endogenous to patent policy and the economics of sufficient incentives, the courts would have to identify the level of abstraction at which algorithmic descriptions of software become sufficiently specific to count as the descriptions of the metaphorical structure of software inventions. That is, courts would have to determine the level of specificity at which the functions performed by a software program constitute an answer to the "how" question that provides sufficient rewards to inventors without over-rewarding them.[312] In so doing, courts would need to draw the very line on the spectrum between the general functions that are likely to map onto markets (e.g., end-user preferences) and the specific functions that are not (e.g., how-functions) that, to date, they have avoided drawing in the non-software arts.[313]

Whether either of these undertakings, and the second one in particular, lies within the institutional competence of an Article III court such as the Federal Circuit is an open question, at best. To obtain the technical and economic data that is needed to fix the definition of an algorithm at the optimal level of specificity, courts must rely on the self-interested disclosures of patentees, alleged infringers, and amici, all made in the course of arguing whether an individual patent is broad or narrow, valid or invalid.[314] It may be that algorithms offer the best hope for fixing patent

---

provide criteria for distinguishing a functional goal (overbroad) from a functional step of an algorithm (not overbroad).

311. *See, e.g.*, David C. Bohrer & Michael I. Frankel, *The Question Left Unanswered in* WMS Gaming*: What Is the Algorithm?*, INTELL. PROP. & TECH. L.J., Apr. 2004, at 8 (arguing that Unified Modeling Language could provide a standard method for describing algorithms).

312. *See supra* notes 32–33 and accompanying text.

313. *See supra* notes 80–90 and accompanying text.

314. Dan Burk and Mark Lemley argue that the courts are the best institution through which to achieve industry-specific patent reform precisely because "[t]he [adversarial] litigation process will provide judges with the information they need to decide cases" in an industry-specific manner. BURK & LEMLEY, *supra* note 1, at 104.

law's functionality malfunction in the software arts, but it may also be that the courts lack the institutional competence to do the needed work. If the invention-algorithm equation is to be a software-specific patch for the functionality malfunction, other institutions that can more readily collect and weigh the needed data may need to take the lead.

3. Gottschalk v. Benson

In *Gottschalk v. Benson*,[315] the Supreme Court invalidated a patent on a software-executed method of converting numbers from one form of notation to another, holding that the patent did not recite patentable subject matter under section 101 of the Patent Act. The Court couched its reasoning in terms of algorithms. It identified mathematical algorithms as some of "the basic tools of scientific and technological work," and it held that mathematical algorithms *per se* are unpatentable abstract ideas.[316]

Broadly speaking, *Benson* established a script for assessing whether or not a software claim describes patentable subject matter. It has ensured that courts view algorithm limitations in software claims as red flags that warn of potential invalidity issues. Because any software patent that implicates something that looks like an algorithm must be subject to additional scrutiny to determine if it is a claim to an algorithm *per se*, software patents implicating algorithms are viewed as relatively more problematic than other software claims from a social-welfare perspective and relatively less likely to be upheld as valid in the courts.[317] The Supreme Court's final two cases in its software trilogy of the 1970s and early 1980s came to different bottom-line conclusions concerning the validity of algorithm patents, with *Parker v. Flook* invalidating a patent[318] and *Diamond v. Diehr* upholding one.[319] But, importantly, they both adhered to *Benson*'s script. A long line of cases in the lower courts both before and after *Diehr* was also centered on the hunt for impermissible

315. 409 U.S. 63 (1972).

316. *Id.* at 67. The terminology used to describe algorithms in opinions addressing patentable subject matter is inconsistent. *Benson* refers to algorithms as "ideas." *Id.* at 71. Later Supreme Court opinions modify the terminology to "abstract ideas," *see, e.g.*, Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980), and contemporary Federal Circuit opinions describe *Benson*'s holding as an example of the abstract-ideas exclusion. CLS Bank Int'l v. Alice Corp. Pty., 717 F.3d 1269, 1277–78 (Fed. Cir. 2013) (en banc). Yet, some post-*Benson* Supreme Court cases refer to mathematical algorithms as "laws of nature." *See, e.g.*, Diamond v. Diehr, 450 U.S. 175, 185–86 (1981); Parker v. Flook, 437 U.S. 584, 590 (1978).

317. Until recently, the Patent and Trademark Office still followed this script. *See* Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478, 7489 n.50 (Feb. 28, 1996).

318. *Flook*, 437 U.S. at 588–96.

319. *Diehr*, 450 U.S. at 191–93.

Wash U Law Repository

claims to algorithms *per se*.[320] The screening that *Benson* mandates to identify the subset of algorithm patents that does not describe patentable subject matter means that the validity of all patents reciting algorithms has been viewed with a greater amount of skepticism. Patent drafters have thus avoided reciting algorithm limitations whenever possible.[321]

For the last forty years—an era that encompasses the birth and maturation of the modern software industry—*Benson* has anchored the conventional wisdom on algorithms in the law of software patents.[322] The use of algorithms as the metaphorical structure of software inventions, however, will require turning *Benson* on its head while, at the same time, upholding it. Where *Benson* suggests that software claims reciting algorithms are *more* likely to be problematic, the invention-algorithm equation suggests that software claims reciting algorithms are *less* likely to be problematic. Where *Benson* holds that claims to algorithms are unpatentable claims to abstract ideas, the invention-algorithm equation holds that functional software claims that do not recite algorithm limitations are unpatentable claims to abstract ideas.

The combination of *Benson* and the invention-structure equation puts the drafters of software patents in a difficult bind. In most arts, claims that simply recite the structural properties of invention are routinely upheld. They represent a clear way of working around the prohibition on functional claims. A claim that describes a chemical compound by its full molecular structure, or a claim to a mousetrap that recites only its arrangement of parts, is unlikely either to be an unpatentable basic tool

---

320. For an extensive examination of *Benson*'s legacy in software cases through 1990, see Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1032–1113 (1990).

321. Robin Feldman draws a direct connection between *Benson* and the broad, functional claims of contemporary software parts. Software inventors opted not to include any algorithmic specificity in their claims, and thus sought sweepingly broad functional claims, precisely to avoid having their claims red-flagged under *Benson*. ROBIN FELDMAN, RETHINKING PATENT LAW 109–12 (2012).

322. For a small sampling of commentary that examines algorithm patents through the lens established in *Benson*, see Donald S. Chisum, *The Patentability of Algorithms*, 47 U. PITT. L. REV. 959 (1986); Irah H. Donner & J. Randall Beckers, *Throwing Out Baby* Benson *with the Bath Water: Proposing a New Test for Determining Statutory Subject Matter*, 33 JURIMETRICS J. 247 (1993); Jonathan N. Geld, *General Does Not Mean Generic—Shedding Light on* In re Alappat, 4 TEX. INTELL. PROP. L.J. 71 (1995); Lee A. Hollaar, *Justice Douglas Was Right: The Need For Congressional Action On Software Patents*, 24 AIPLA Q.J. 283 (1996); Newell, *supra* note 212; Samuelson, *supra* note 320; Richard H. Stern, *Tales from the Algorithm War:* Benson *to* Iwahashi, *It's Deja Vu All Over Again*, 18 AIPLA Q.J. 371 (1991); Jur Strobos, *Stalking the Elusive Patentable Software: Are There Still* Diehr *or Was It Just a* Flook*?*, 6 HARV. J.L. & TECH. 363 (1993); Swinson, *supra* note 305; Maximilian R. Peterson, Note, *Now You See It, Now You Don't: Was It a Patentable Machine or an Unpatentable "Algorithm"? On Principle and Expediency in Current Patent Law Doctrines Relating to Computer-Implemented Inventions*, 64 GEO. WASH. L. REV. 90 (1995).

claim or to be invalidated for overbreadth. In the software arts, however, *Benson* means that the analogs of raw structural claims—namely raw algorithm claims—are basic tool claims that are not patentable subject matter. Yet, a patent attorney who recites the metaphorical structure of a software invention in the abstract and claims an algorithm *per se* will have his claim invalidated under *Benson*. In software—and only software—claim drafters must walk a fine line between insufficient and excessive "structural" limitations.

Benson and the invention-algorithm equation are clearly in tension, yet there are perhaps ways to manage this tension and create a workable set of validity rules for software patents. For example, perhaps there is a large Goldilocks zone. If *Benson* and the invention-algorithm equation were only to eliminate a small number of claims at each extreme, then there could be ample space in the middle for patent drafters to include algorithm limitations without claiming algorithms in the abstract.[323] Alternatively, perhaps *Benson* only pertains to a particular type of algorithm. For example, there is clearly textual support for limiting *Benson* to the realm of the mathematical algorithm. *Benson* describes its own holding as a bar on claims to a mathematical formula,[324] and the Court later reaffirmed in *Flook* that *Benson* was aimed only at mathematical algorithms.[325] Yet, the Court has more recently extended the logic of *Benson* and *Flook* to the abstract idea of "the basic concept of hedging," complicating a narrow, mathematical interpretation of *Benson*.[326] In any case, managing the tension between *Benson* and the invention-algorithm equation will require a reconsideration and clarification of *Benson* that, to date, the courts have been reluctant to undertake.

In conclusion, consider one recent example of the conflict between *Benson* and the invention-algorithm equation. *Bilski v. Kappos* is best known for the Supreme Court's decision to hold a broad claim to a method

323. It is not self-evident, however, that a Goldilocks zone exists. Under the computer-science definition of an algorithm, *see supra* notes 212–15 and accompanying text, most all method claims describe algorithms in the abstract. Most all method claims describe a step-by-step procedure for achieving a result of some kind. A more precise definition of the type of algorithm that cannot be claimed in the abstract under *Benson* may therefore be needed to create a Goldilocks zone.

324. Gottschalk v. Benson, 409 U.S. 63, 71–72 (1972).

325. Parker v. Flook, 437 U.S. 584, 585–86 (1978). The basic-tools rationale of *Benson* also has an intuitive fit with mathematical principles in particular.

326. Bilski v. Kappos, 130 S. Ct. 3218, 3231 (2010). Additionally, the Federal Circuit has been less than clear in its position on what constitutes a mathematical algorithm under *Benson*. *Compare In re* Iwahashi, 888 F.2d 1370, 1374 (Fed. Cir. 1989), *with In re* Trovato, 42 F.3d 1376, 1378–80 (Fed. Cir. 1994). Furthermore, the distinction between mathematical and non-mathematical algorithms is conceptually problematic, so the distinction may not prove to be an effective limiting principle for *Benson*. Newell, *supra* note 212, at 1204; Samuelson, *supra* note 320, at 1123–24.

Wash U Law Repository

of hedging a certain type of risk in commodities markets to be an unpatentable claim to an abstract idea.[327] However, *Bilski* also addressed the patentability of a narrower, dependent claim in which a mathematical formula specified how the fixed price for the hedged transaction was to be determined.[328] Under the invention-algorithm equation, the case for the patentability of the narrow, dependent claim under section 101 should have been stronger than the case for the patentability of the broad, independent claim. The addition of the specific way of performing the hedging as a limitation on claim scope should have weighed in favor of patentability. Yet, the Supreme Court saw no daylight at all between the two claims. The Court stated that the dependent claim simply "reduced" the concept of hedging "to a mathematical formula" and was therefore unpatentable subject matter under *Benson* and *Flook*.[329] The mathematical nature of the algorithm led to the Court to pay no attention whatsoever to a scope-restricting, algorithmic limitation on a broad, functional claim. *Benson* trumped the invention-algorithm equation.

V. CONCLUSION

The root cause of the problem of software-patent overbreadth is patent law's functionality malfunction. The invention-structure equation is patent law's traditional doctrinal mechanism for curtailing the scope of functional claims, and it breaks down when it is brought to bear on technologies that, like software, are functional on all relevant levels of definition. Patent law's conventional scope-curtailing doctrines therefore have not been able to get the traction in the software arts that they get in most other arts, and purely functional, overbroad software claims have become the norm.

Courts can attempt to fix the functionality malfunction in the software arts by translating the invention-structure equation into an invention-algorithm equation. They can identify algorithms as the metaphorical structure of software inventions and limit the scope of software claims to the particular algorithms produced and disclosed by an inventor. In fact, the Federal Circuit has already begun to use the invention-algorithm equation to curtail the scope of software claims, albeit in a halting, uncoordinated, and inconsistent way. However, the recursive nature of an algorithm and *Gottschalk v. Benson*, among other problems, mean that

327. *Bilski*, 130 S. Ct. at 3231.
328. *Id.* at 3223–24.
329. *Id.* at 3231.

even a more sustained focus on algorithms is unlikely to put the scope of software patents on par with the scope of patents in other arts.

This Article cannot speak directly to the best route forward for dealing with software patents. Perhaps the costs of the overbreadth of contemporary software patents are the minimum price that needs to be paid for sufficient incentives. Alternatively, perhaps these costs are so significant and difficult to eliminate that the best route forward is to exclude software from patentable subject matter. Or, perhaps they can be reduced dramatically through the adoption of a technology-specific rule such as the invention-algorithm equation. This Article argues only that the minimization of the costs of software-patent overbreadth is going to be more difficult than has previously been recognized. The functionality malfunction is a structural problem (in more than one sense). The open question is not whether software as a technology is intrinsically different in a way that increases the costs of software patents relative to patents in other technologies. The open question is rather how large those software-specific costs must be.

Wash U Law Repository