THE *WILLIAMSON* REVOLUTION IN SOFTWARE'S STRUCTURE

Kevin Emerson Collins[†]

ABSTRACT

In *Williamson v. Citrix Online*, the Federal Circuit altered the threshold test for determining whether a functional claim limitation that does not use the term "means" is governed by the scope–narrowing rules of § 112(f). Before *Williamson*, there was a strong presumption that functional claim limitations without the word "means" were not governed by § 112(f). After *Williamson*, § 112(f) now governs all functional limitations without the word "means" that do not recite sufficient structure for performing the claimed function.

A common, incrementalist interpretation of *Williamson* is that the alteration of the § 112(f) threshold test will only have a small impact on the law of functional claiming. This interpretation frames *Williamson* as a case that will simply move the needle on the *quantitative question* about structure in the threshold test: How much structure need be recited in a limitation to avoid § 112(f)?

In contrast, this Essay argues that, at least for software limitations in particular, *Williamson* will have a revolutionary impact on the law of functional claiming. *Williamson* will force the Federal Circuit to formulate a new answer to the more fundamental *definitional question* about software's structure: What constitutes structure in a software invention in the first place? *Williamson* demands a revolution in the definition of software's structure because the Federal Circuit's pre–*Williamson* doctrine that software's structure is an algorithm cannot do the work that *Williamson* requires. Although the concept of an algorithm can define corresponding structure in a specification in the course of construing the scope of a limitation that is known to be governed by § 112(f), it cannot identify structure in a claim limitation in the course of the threshold determination of whether or not a limitation is governed by § 112(f). After demonstrating that *Williamson* demands a revolution in software's structure under § 112(f), this Essay briefly also considers several different paths that the *Williamson* revolution could take.

DOI: https://dx.doi.org/10.15779/Z38SJ19R03

† Professor of Law, Washington University.

TABLE OF CONTENTS

I. INTRODUCTION

In *Williamson v. Citrix Online*, the Federal Circuit altered the threshold test for determining whether a limitation that employs functional language is subject to § 112(f) of the Patent Act.[1] Section 112(f) expressly sanctions broad, functional claim language: claim limitations "may be expressed as a means . . . for performing a specified function without the recital of structure [or] material . . . in support thereof."[2] However, as a price for using such language, § 112(f) mandates a scope–narrowing rule of claim construction: functional limitations "shall be construed to cover [only] the corresponding structure [or] material . . . described in the specification and equivalents thereof."[3] In the decade preceding *Williamson*, the Federal Circuit developed formalistic, strong presumptions that usually allowed a patent

1. Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1347–49 (Fed. Cir. 2015) (en banc).
2. 35 U.S.C. § 112(f) (2012).
3. *Id.*

drafter to opt into or, more importantly, out of § 112(f). If a functional claim limitation contained the word "means," § 112(f) almost always governed. However, if the limitation employed a synonym for "means," such as "device" or "mechanism"—words that the Federal Circuit refers to as "nonce" words—§ 112(f) almost never applied.[4] *Williamson* expanded the reach of § 112(f) by reducing the strength of the latter presumption and allowing it to be overcome with a showing that a limitation recites "function without reciting sufficient structure for performing that function."[5] In short, *Williamson* promises to rein in the scope of some overbroad, functionally defined claims that do not employ the term "means."

A year after *Williamson*, there is little evidence of how the Federal Circuit will interpret its new § 112(f) threshold test.[6] One common prediction about *Williamson*'s future impact is that it may lead to only incremental change in the law of functional claiming. This view frames *Williamson* as a case that will do nothing but move the needle a bit on the *quantitative question* about structure in the threshold test: How much structure in a functional claim limitation is enough to avoid § 112(f)?[7] In contrast, this Essay argues that, with respect to software patents in particular, *Williamson* mandates a revolution in the law of functional claiming.[8] To bring *Williamson* to bear on software patents, the Federal

4. *Williamson*, 792 F.3d at 1348–49.

5. *Id.* at 1349 (citing Watts v. XL Sys., Inc., 232 F.3d 877, 880 (Fed. Cir. 2000)).

6. The Federal Circuit has decided only one case addressing the § 112(f) threshold test since *Williamson*. Media Rights Tech. v. Capital One Fin. Corp., 800 F.3d 1366, 1373 (Fed. Cir. 2015) (holding that a "compliance mechanism" limitation is subject to § 112(f)). For an overview of several district court cases in which the § 112(f) threshold test was performed both before *Williamson* and again after, see Paul R. Gugliuzza, *Early Filing and Functional Claiming*, 96 B.U. L. REV. 1223, 1232–43 (2016).

7. Gugliuzza, *supra* note 6; Luiz Felipe Oliveira, *Means-Plus-Function Claims: An Analysis of the Federal Circuit's Ruling in* Williamson v. Citrix Online, 11 J. INTELL. PROP. L. & PRACTICE 199 (2016); Eric P. Raciti, *Means Plus Function Claiming: What Does It Mean to Be a Means, When Are Means Means, and Other Meaningful Questions*, LANDSLIDE, March–April 2016, at 19. Incrementalists who assert that *Williamson*'s impact is limited to the quantitative question of the threshold test may, in theory, predict either minor or significant movement of the needle.

8. Software patents are widely recognized as troubling for a number of policy reasons, and the expansive scope that can be achieved through unbridled functional claiming has been identified as one of these reasons. Kevin Emerson Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 WASH. U. L. REV. 1399 (2013); Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905 (2013). Although there are good policy reasons to want to see *Williamson* lead to a revolution that cuts back on the permissible scope of functional software claims, this Essay does not enter the policy debate over

Circuit must ask and answer a fundamental question that, to date, it has not yet openly addressed. This fundamental question is the *definitional question* about software's structure. What properties of a software program should count as structure when they are recited as claim limitations? In other words, what is the structure of a functional software limitation in the first place?

To support the argument that *Williamson* mandates a revolution in the definition of software's structure, this Essay makes three cumulative points. First, it highlights the sui generis nature of the definitional question about § 112(f) structure in software. Identifying structure may be an intuitive exercise in most technologies, but it is not in software. Second, this Essay argues that the Federal Circuit ducked the definitional question about software's structure in its pre–*Williamson* jurisprudence. The Federal Circuit did define software's structure as an algorithm in the context of the search for corresponding structure in the specification during claim construction, but an algorithm provides only a relational definition that is meaningless in the search for structure in a claim's limitations during the § 112(f) threshold test. Third, this Essay maps out the difficult path forward that the Federal Circuit must follow in the post–*Williamson* era in order to provide a meaningful answer to the definitional question about software's structure in the § 112(f) threshold test.

Software is an unusual technology. For most technologies, including mechanical technologies, the answer to the definitional "What is structure?" question is intuitive to technological neophytes and experienced patent judges alike: structure includes the physical, spatial, and material properties of a technology. However, in software, the definitional question does not have a simple answer. The physical, structural qualities of a software invention are irrelevant to the definition of what a software inventor has actually invented. Software has been engineered with the express goal of allowing programmers to remain ignorant of the physical structure of their inventions. There is no relevant physical structure on which an economically rational patent regime can rely to curtail the scope of functional claims; instead, software inventions can only reasonably be defined by reciting what the software does or how it performs in functional terms. Thus, a sui generis definition of structure is needed to make § 112(f) a meaningful limit on software claims. This definition must invoke software's metaphorical or logical structure in the sense that a more granular, functional description of how software performs should count as

functional claiming. It more modestly reveals the conceptual poverty of the pre–*Williamson* definition of software's structure and the need that *Williamson* creates to rethink that definition.

a description of structure, despite the fact that the description is still functional as a purely linguistic matter.

To be clear, the idea that § 112(f) requires a sui generis definition of software's structure had been recognized well before *Williamson*. In the context of searching for the corresponding structure of a § 112(f) limitation in the specification during claim construction, the Federal Circuit has long held that software's structure is an "algorithm" or a step–by–step procedure specifying how to perform the function recited as a claim limitation.[9] Given that the Federal Circuit has already answered the definitional question in this context, the incrementalist prediction about the likely impact of *Williamson* might seem eminently reasonable, even with respect to software patents. Why not assume that the well–established answer to the definitional question in the context of identifying § 112(f) corresponding structure in the specification during claim construction can be carried over and used to identify § 112(f) structure in the claim limitations in the threshold test?[10] The flaw in this assumption lies in the overlooked, purely relational nature of the Federal Circuit's pre–*Williamson* definition of an algorithm. This definition of an algorithm can only identify software's structure in relation to the baseline provided by particular claim limitations: an algorithm is a step–by–step procedure specifying how to perform *a function recited as a claim limitation*. During the search for corresponding structure in the specification needed to construe a § 112(f) limitation, this relational definition of software's structure may be awkward at times as a policy matter, but it at least provides a conceptually coherent doctrinal rule.[11] However, it is nonsensical to use this relational definition to identify sufficient claim structure in the context of the threshold test to determine whether § 112(f) applies in the first place. Asking whether a claim limitation recites a step–by–step procedure for specifying how to perform a claim limitation makes no sense. Given that an algorithm is defined only in relation to the tasks specified as claim limitations, one cannot identify the structure in claim limitations that is needed to avoid § 112(f) under *Williamson* using the pre–*Williamson* definition of an algorithm.

Before *Williamson*, the strong presumptions of the § 112(f) threshold test and the relational definition of an algorithm allowed the Federal Circuit

9. WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1349 (Fed. Cir. 1999).

10. The Federal Circuit has noted that "[n]aturally, there is some analytical overlap between . . . [the] two steps" of the threshold test and the identification of corresponding structure. Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1294–1304 (Fed. Cir. 2014).

11. The awkwardness inheres in the formalistic, not substantive, limit on permissible claim breadth that using a relational definition of software's structure generates. *See infra* note 92.

to suppress the true extent of the sui generis law needed to bring § 112(f) to bear on software. They allowed the Federal Circuit to maintain the appearance that the application of § 112(f) to software claims was at least close to situation normal, when it was, in fact, anything but. *Williamson*, however, will pull back the curtain and force the Federal Circuit to openly grapple with the unusual nature of software's structure. This is the *Williamson* revolution.

While the call for revolution to implement *Williamson* is clear, what form that revolution will take is not. There are several ways in which the Federal Circuit can, in theory, make do with its relational definition of an algorithm in the threshold test in a post–*Williamson* world. However, the change required to harmonize the application of § 112(f) to software and the application of § 112(f) to other technologies is a revolution that defines software's structure in a stand–alone manner and that does not incorporate the functions recited in a particular claim limitation as a baseline. A stand–alone definition of software's structure would likely require a levels–of–generality analysis. Functional descriptions of a software invention exist at many different levels or rungs on a ladder of generality. At some point on the descent of this ladder, a highly general, structureless description of a software program transforms into a granular description that refers to metaphorical, but not literal, structure. Articulating such a stand–alone definition of software's structure would unquestionably be a difficult undertaking.[12] The echoes of Learned Hand's levels–of–generality test for drawing the idea/expression dichotomy in copyright are clear, and that dichotomy is notorious for its lack of ex ante clarity even without the technological complexity of software.[13] A stand–alone definition of structure could be based on what I have elsewhere described as a line between functional limitations reciting end–user preferences (that should be subject to § 112(f)) and functional limitations reciting how those end–user preferences are achieved (that should not be subject to § 112(f)).[14] Yet regardless of its final formulation, the initial step for developing the needed definition should ideally be an interdisciplinary conversation that leverages the expertise of lawyers, computer scientists, and economists to identify the levels of generality at which a functional description of software could be

12. It is precisely the difficulty of formulating such a stand–alone definition of software that has led the Federal Circuit to give only lip service to the statutory mandate in § 112(f) for step–plus–function method claims. *See infra* notes 130–132.

13. *See* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) ("Nobody has ever been able to fix that boundary, and nobody ever can.").

14. *See* Collins, *supra* note 8, at 1466–67.

deemed to be logical structure for the purpose of § 112(f) as a menu of options.[15]

The initial three parts of this Essay provide background. Part II introduces § 112(f). Part III identifies why the application of § 112(f) to software requires sui generis rules that focus on logical structure, and it presents the Federal Circuit's cases that require the disclosure of an algorithm in the specification as the corresponding structure for § 112(f) software limitations. Part IV discusses the § 112(f) threshold test in greater detail and explains how *Williamson* altered it. The final two parts address life after *Williamson*. Part V argues that the Federal Circuit's pre–*Williamson* software cases provide little guidance for courts seeking to answer the definitional "What is structure?" question as part of the threshold test and that *Williamson* therefore mandates revolutionary change in § 112(f) as it applies to software. Part VI briefly maps out several different paths that the revolution could take, focusing principally on the levels–of–generality analysis that is needed to develop a stand–alone, rather than relational, definition of software's structure. Part VII concludes.

II. THE POLICY AND DOCTRINE OF § 112(F)

During the first half of the twentieth century, the Supreme Court invalidated a number of claims that employed purely functional language because such claims granted patentees rights that were overbroad with respect to the patentees' actual contributions to progress.[16] Most famously, the Court's 1946 opinion in *Halliburton Oil Well Cementing Co. v. Walker* discussed the overbreadth that inheres in the "overhanging threat of the functional claim" at length:

> Just how many different devices there are of various kinds and characters which would serve to [fulfill the claimed function] we do not know. . . . In this age of technological development there may be many other devices beyond our present information or

15. *Id.*

16. *See, e.g.*, Gen. Elec. Co. v. Wabash Appliance Corp., 304 U.S. 364, 368–71 (1938) ("The claim uses indeterminate adjectives which describe the function of the grains to the exclusion of any structural definition, and thus falls within the condemnation of the doctrine that a patentee may not broaden his product claims by describing the product in terms of function."); Holland Furniture Co. v. Perkins Glue Co., 277 U.S. 245, 257–58 (1928) ("That the patentee may not by claiming a patent on the result or function of a machine extend his patent to devices or mechanisms not described in the patent is well understood."). The explanation of why functional claims are overbroad is more complicated than is commonly acknowledged. *See* Collins, *supra* note 8, at 1411–24.

> indeed our imagination which will perform that function and yet fit these claims.[17]

While these Supreme Court cases remain good law in the sense that a patent applicant who is the first to invent a technology that performs a function cannot claim all devices that can perform that function, Congress softened their impact by enacting what is now § 112(f) as part of the 1952 Patent Act:

> 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.[18]

Section 112(f) sanctions functional claim limitations, which can be helpful to time–constrained patent drafters because describing an invention without using functional limitations is often a difficult undertaking. However, § 112(f) exacts a price from patent drafters who use functional limitations. It codifies an exception to the default rule of claim construction specified in *Phillips v. AWH*[19]: the permitted functional claim language refers only to devices that have the "corresponding structure" or "material" that the patentee discloses in the specification, as well as its equivalents.[20] A limitation construed under § 112(f) is usually significantly narrower than the same limitation would be if it were to be construed under the default rules of claim construction specified in *Phillips* because, under § 112(f),

17. Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 12 (1946). *Halliburton* is sometimes mistakenly labeled as an indefiniteness holding rather than an overbreadth holding. *See* Collins, *supra* note 8, at 1429 n.122.

18. 35 U.S.C. § 112(f) (2012).

19. Phillips v. AWH Corp., 415 F.3d 1303, 1311–24 (Fed. Cir. 2005) (en banc) (laying out the default rules of claim construction).

20. 35 U.S.C. § 112(f). Some of the Supreme Court's reasoning in its functional–claiming cases focused narrowly on the vice of using functional claiming at the point of novelty. *See, e.g.*, *Gen. Elec.*, 304 U.S. at 371. However, Congress's response to these cases was not limited to functional limitations at the point of novelty. Section 112(f), on its face, applies to all functional limitations. There are good reasons to believe breadth at a claim's point of novelty is far more problematic than breadth at limitations describing aspects of prior art technologies. *See* 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 (2011); Lemley, *supra* note 8, at 958–59. This Essay, however, brackets the issue of whether § 112(f) should have more bite at a claim's point of novelty and leaves that inquiry for another day.

claim scope hews more closely to the particular embodiments disclosed in the specification.[21]

Today, the process of claim construction under § 112(f) proceeds in three phases. The first phase is a threshold test: Is a given claim limitation the type of limitation that is subject to § 112(f), or should *Phillips* determine the meaning of the claim language? The doctrinal formulation of the threshold test has shifted over time—and was most recently altered by *Williamson*—but, to one degree or another, it has always involved a search for sufficient structure in the claim limitation.[22] The more structure that a claim limitation recites, the less pressing the policy concerns about overbroad, functional claiming and the less likely the scope–limiting rule of § 112(f) is to govern claim construction. The second phase employs the default rules of claim construction to interpret the meaning of the functional language employed in the claim.[23] The third phase then identifies the "corresponding structure" disclosed in the specification that is capable of performing the disclosed function.[24] In order to fall within literal claim scope, a technology must literally perform the function specified in the claim limitation, and its structure must be the corresponding structure disclosed in the specification or its equivalents.

While the first and third phases both involve a search for structure, each one looks for structure in a different place in the patent document. The first phase (the threshold test) looks for structure in the claim limitations

21. Articulating an interesting historical argument, John Duffy argues that prior to the creation of the Federal Circuit, the default rule of claim construction for all limitations resembled the rule articulated in § 112(f) and that the stakes of the § 112(f) threshold test were therefore low. John F. Duffy, *Counterproductive Notice in Literalistic Versus Peripheral Claiming*, 96 B.U. L. REV. 1197, 1206–10 (2016). Professor Duffy also argues as a statutory matter that Congress intended § 112(f) to bring the default rule of claim construction to bear on functional claims. *Id.* According to Professor Duffy, it was the Federal Circuit's shift to a broader, "literalistic" rule of claim construction by the 1990s that "transformed" § 112(f) into a scope–narrowing rule of claim construction and that introduced economic importance into the § 112(f) threshold test. *Id.*

22. The threshold test and *Williamson*'s impact thereon are addressed at greater length below. *See infra* Part IV.

23. *See Generation II Orthotics, Inc. v. Med. Tech., Inc.,* 263 F.3d 1356, 1364–65 (Fed. Cir. 2001).

24. That is, the specification must contain descriptive text by which a person of skill in the field of the invention would "know and understand what structure corresponds to the means limitation." Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1340 (Fed. Cir. 2008). An enabling specification is not enough. Biomedino LLC v. Waters Techs. Corp., 490 F.3d 946, 953 (Fed. Cir. 2007). The disclosed structure must also be clearly linked to the function recited in the claim limitation. B. Braun Med., Inc. v. Abbott Labs., 124 F.3d 1419, 1424 (Fed. Cir. 1997).

themselves to determine whether § 112(f) governs. In contrast, the third phase looks for structure in the specification in order to determine the scope of a claim limitation that is governed by § 112(f).

If a limitation is governed by § 112(f) and the specification does not disclose any corresponding structure for that limitation, then the claim is invalid for indefiniteness under § 112(b) of the Patent Act.[25] Indefiniteness holds that claims that employ limitations whose meanings cannot be ascertained with reasonable certainty are invalid.[26] Indefiniteness is a common–sense rule: the scope of a claim to a "thingamajig" cannot be ascertained, there is poor public notice, and there are many instances in which neither the validity nor the infringement analyses can proceed. Section 112(f) states that a functional claim limitation refers to the corresponding structure and its equivalents, so a § 112(f) limitation has no discernable meaning if the specification does not disclose any corresponding structure.[27]

The apparatus limitations that are governed by § 112(f) are frequently called "means–plus–function" limitations. However, § 112(f) also governs action limitations in method claims. A claim limitation "may be expressed as a . . . step for performing a specified function without the recital of . . . acts in support thereof, and such claim shall be construed to cover the corresponding . . . acts described in the specification and equivalents thereof."[28] Following the statue, a "step" is a generic element of a process, and an "act" refers to one of the sub–steps that are needed to implement a "step."[29] While the Federal Circuit has occasionally discussed the notion of a step–plus–function method claim that is subject to § 112(f), it has never applied § 112(f) to an action in a method claim.[30]

25. *In re* Donaldson Co., 16 F.3d 1189, 1195 (Fed. Cir. 1994) (en banc).

26. Nautilus, Inc. v. Biosig Instruments, Inc., 134 S. Ct. 2120, 2124 (2014). Indefiniteness derives from the statutory requirement that a patent "particularly point[] out and distinctly claim[] the subject matter which the inventor regards as the invention." 35 U.S.C. § 112(b) (2012).

27. *Donaldson*, 16 F.3d at 1195.

28. 35 U.S.C. § 112(f) (2012).

29. O.I. Corp. v. Tekmar Co. Inc., 115 F.3d 1576, 1582–83 (Fed. Cir. 1997).

30. *See, e.g.*, Masco Corp. v. United States, 303 F.3d 1316, 1326–28 (Fed. Cir. 2002); Seal-Flex, Inc. v. Athletic Track & Court Const., 172 F.3d 836, 848–51 (Fed. Cir. 1999) (Rader, J., concurring); *O.I. Corp.*, 115 F.3d at 1582–84.

III. SECTION 112(F) AND SOFTWARE

The application of § 112(f) to software requires a sui generis modification of conventional § 112(f) doctrine. Section III.A posits that this modification is required because software is, for practical purposes at least, a purely functional technology without any physical structure that is relevant to what a software inventor invents. Section III.B explores how the Federal Circuit had already adapted § 112(f) to software before *Williamson* by concluding that an algorithm is the corresponding structure in the specification for software means–plus–function limitations.

A. THE SOFTWARE/§ 112(F) MISMATCH[31]

There is a fundamental mismatch between the rules of § 112(f) and software inventions: in most technologies, § 112(f) builds on an intuitive distinction between the physical structure and the function of a technology, but in software, § 112(f) cannot build on this distinction. An economically rational patent regime cannot require that the physical, structural properties of a software program be recited as claim limitations. Software inventions are, at least as a practical matter and for the purpose of patent law, a purely functional technology.

In the historical core of § 112(f) claiming in the mechanical arts, the principles distinguishing the structural properties of an invention from its functional properties are self–evident, intuitive concepts. In brief, a structural property is what an invention "is," whereas a functional property is what an invention "does."[32] For example, a "skid plate,"[33] "button and hole arrangement,"[34] and "a rotatable disc with an opening sits above the

31. The argument in this Section draws from Collins, *supra* note 8, at 1440–43.

32. *In re* Swinehart, 439 F.2d 210, 212 (C.C.P.A. 1971). The distinction between structural and functional properties is sufficiently basic and intuitive that it often underlies the philosophical analysis of the properties of things themselves. *See, e.g.*, Peter Kroes, *Technological Explanations: The Relation between Structure and Function of Technological Objects*, 3 SOC'Y FOR PHIL. & TECH. 18, 18 (1998) (discussing "two different modes of description, viz., a *structural* and a *functional* mode of description" for technological objects).

33. Chiuminatta Concrete Concepts, Inc. v. Cardinal Indus., Inc., 145 F.3d 1303, 1309 (Fed. Cir. 1998) (noting that the specification described a skid plate as "a generally rectangular strip of metal having rounded ends . . . between which is a flat piece" where "[t]he flat piece . . . is generally parallel to the base plate").

34. Al-Site Corp. v. VSI Int'l, Inc., 174 F.3d 1308, 1315–16 (Fed. Cir. 1999).

container cap"[35] are all structural descriptions of objects.[36] Few—if any—cases in the mechanical arts bother even to expressly pose the definitional "What is structure?" question because the answer is so self–evident. Of course, this clarity does not mean that the outcome of the § 112(f) threshold test is never contested in the mechanical arts. There are many cases in the mechanical arts in which it is unclear whether the quantum of structure recited as a claim limitation is sufficient to avoid § 112(f).[37] However, this uncertainty inheres in either the meaning of language or the answer to the quantitative question of the threshold test, not the answer to the definitional question of the threshold test. That is, the uncertainty follows from whether claim language refers to enough of the structural properties of a technology, not whether any particular property, once identified as a claim limitation, is a structural property.

In the software arts, however, § 112(f) cannot leverage the intuitive distinction between physical structure and function into a convenient proxy for limiting permissible claim scope. Of course, software programs do have physical, structural properties, just like other technologies do. Software may be commonly described as intangible, [38] but this description is technically inaccurate.[39] 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 or switches in the processor is open or

35. Sage Prods., Inc. v. Devon Indus., Inc., 126 F.3d 1420, 1428 (Fed. Cir. 1997). Here, "rotatable" adds a dollop of function to the description.

36. Structure is also an uncontroversial concept in the chemical and biochemical arts because structure is equated with molecular structure. *See* Ariad Pharm., Inc. v. Eli Lilly & Co., 598 F.3d 1336, 1350 (Fed. Cir. 2010) (en banc) (treating the recitation of molecular structure in a claim limitation as the type of structure that supports the claim's validity). However, the task of policing overbroad, functional claims in these arts has fallen to the written description doctrine rather than to means–plus–function claiming. Collins, *supra* note 8, at 1430–33.

37. *Cf. supra* note 7 and accompanying text (discussing the quantitative question of the § 112(f) threshold test).

38. *See, e.g.*, Bancorp Servs. v. Sun Life Assurance Co., 687 F.3d 1266, 1277–79 (Fed. Cir. 2012) (making an analogy between computer-executed and mental processes); *In re* Grams, 888 F.2d 835, 840 (Fed. Cir. 1989) (labeling software–executed processes as non–physical steps); 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.").

39. Microsoft Corp. v. AT&T Corp., 550 U.S. 437 (2007) (discussing the physicality of any copy of a software program that can generate functional effects in the course of assessing when software can be a "component" under § 271(f)).

closed.[40] These are software's physical, structural properties. The crux of the problem for § 112(f) is thus not that software lacks physical, structural properties but rather that those properties are usually irrelevant to the task of identifying, delineating, or defining a protectable software invention. This irrelevance can be seen on two distinct levels. First, minor changes in code can lead to significant changes in software's physical, structural properties. A software invention can be implemented in entirely different code in the same programming language or in an entirely different language, and the sets of code that all embody the invention have few, if any, structural properties in common.[41] Second, executing the same code on different hardware leads to radical changes in software's physical, structural properties. 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 each requiring the software to adopt entirely different physical, structural properties to yield the intended, functional result.[42] In sum, the many distinct embodiments of a software invention are unlikely to have any physical, structural properties in common that can be used to delineate what an inventor has invented. Any attempt to define software by its physical, structural properties will fail.[43]

The irrelevance of software's physical, structural properties to the definition of a software invention means that, as a practical matter, software is a purely functional technology. In the software arts, an invention is its function, not its structure.[44] For patent drafters writing software claims,

40. WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1348 & n.3 (Fed. Cir. 1999); Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform*, 7 UCLA J.L. & TECH. 1, 38–39 (2003) ("[T]he physical structure of the computer is critical if software is to execute and thereby perform its intended functions."); *cf. In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc) (holding that a computer programmed with a new software program is a new machine under the novelty doctrine that is structurally distinct from prior art machines).

41. Furthermore, hardware and software implementations of any given program are functionally interchangeable despite their radically different structural properties. *In re* Alappat, 33 F.3d 1526, 1583 (Fed. Cir. 1994) (Rader, J., concurring).

42. *See* W. DANIEL HILLIS, THE PATTERN ON THE STONE 56–58 (1998) (discussing interpreters and compilers); *cf. Microsoft Corp.*, 550 U.S. at 450 ("Software . . . is a stand-alone product developed and marketed for use on many different types of computer hardware." (internal quotations omitted)).

43. Plotkin, *supra* note 40, at 46 & n.126 ("For 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.").

44. Note, *Everlasting Software*, 125 HARV. L. REV. 1454, 1456 (2012).

software functionality is like a never–ending set of nested Russian dolls: you open up one more general functional description to look for structure, and all you find is another, more specific functional description.[45] The purely functional nature of a software program is not an accident. To the contrary, it is a key feature that has been engineered into the very nature of software. The value of modern software lies in the fact that the coding of a software program that possesses any given set of logical, functional properties need not involve any consideration of software's physical, structural properties.[46]

The purely functional nature of software raises a challenge for any patent doctrine that builds on the categorical distinction between the structural and functional properties of an invention and prohibits purely functional claim limitations. This mismatch between software and § 112(f) presents two doctrinally simple solutions, but each is extreme and unpalatable.[47] On the one hand, we could refuse to make any allowances for the purely functional nature of software and mandate that software patents recite physical, structural properties as limitations on claim scope. This option would render patent protection for software absurd and economically meaningless because avoiding infringement would be a trivial undertaking. On the other hand, we could exempt software from the strictures of § 112(f) and permit purely functional claiming at any level of generality. This option ignores the Supreme Court's cases from the early twentieth century that prohibit purely functional claiming and raises significant policy concerns about the social costs of software–claim overbreadth.[48] Given the problematic nature of both simple, but extreme, doctrinal solutions, an exploration of the sui generis ways in which § 112(f) could be modified for a purely functional art like software is a worthwhile undertaking. Exploring

45. Technically, the nested dolls do end at some point. *See infra* note 57 (discussing *In re* Katz).

46. Plotkin, *supra* note 40, 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."); *see also id.* at 26 ("The process of computer programming enables a programmer to create a machine that has a particular novel physical structure for performing a particular function without requiring the programmer to design the novel features of the machine in physical terms."); *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.").

47. If either were adopted, it might well be preferable to eliminate patent protection for software altogether.

48. *See supra* notes 16–17 and accompanying text.

when function should count as metaphorical structure offers the greatest promise for a way to use § 112(f) to navigate a course between the Scylla and Charybdis of trivially narrow and vastly overbroad software claims.

B. ALGORITHMS AS CORRESPONDING STRUCTURE

The need to create a sui generis doctrine for answering the definitional, "What is structure?" question of § 112(f) for software claims has not been lost on the Federal Circuit. The Federal Circuit has developed an extensive body of case law that identifies an algorithm as the corresponding structure in a patent disclosure for a § 112(f) software limitation.[49]

The Federal Circuit first introduced the notion of an algorithm as the corresponding structure in its 1999 opinion in *WMS Gaming, Inc. v. International Game Technologies*.[50] The district court construed a functional software limitation employing the term "means" broadly to encompass any table, formula, or algorithm for performing the claimed function, but the Federal Circuit insisted that the scope of the claim was limited to "the special purpose computer programmed to perform the disclosed algorithm" in the specification, as well as its equivalents.[51] In subsequent cases, the Federal Circuit has expressly defined an algorithm as "a series of instructions for the computer to follow,"[52] "a step-by-step procedure for accomplishing a given result,"[53] and a "sequence of computational steps to follow."[54] Simply put, "the essence of algorithms" is "what to do to perform a task" that is recited as a claim limitation.[55]

One line of § 112(f) cases in particular offers a good window through which to examine what constitutes an algorithm under the Federal Circuit's

49. *See infra* notes 50–59 and accompanying text. However, the Federal Circuit has not developed a definition of software's structure that can function in the § 112(f) threshold test. *See infra* Part V.

50. WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1347–50 (Fed. Cir. 1999).

51. *Id.* at 1349.

52. Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376, 1384 (Fed. Cir. 2011) (quoting *In re* Waldbaum, 457 F.2d 997, 998 (C.C.P.A. 1972)).

53. *Id.* at 1385 (quoting *In re* Freeman, 573 F.2d 1237, 1254 (C.C.P.A. 1978)).

54. Ibormeith IP, LLC v. Mercedes-Benz USA, LLC, 732 F.3d 1376, 1379 (Fed. Cir. 2013) (citations omitted). The Federal Circuit has also stated that there is no single format in which an algorithm must be communicated. Mathematical formulae, prose, and flow charts can all express algorithms. Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1340 (Fed. Cir. 2008).

55. Allen Newell, *The Models Are Broken, The Models Are Broken!*, 47 U. PITT. L. REV. 1023, 1026 (1986); *see also id.* at 1024 ("An algorithm is [a] . . . sequence of steps or operations for solving a class of problems.").

definition. Starting in the late 2000s, the Federal Circuit began to invalidate a large number of software claims with § 112(f) limitations for indefiniteness because the specification did not disclose an algorithm (and thus it did not disclose corresponding structure).[56] If the specification simply repeats the functions recited in the claims, the specification does not disclose an algorithm. For example, in *Finisar Corp. v. Direct TV Group, Inc.*, the Federal Circuit held a § 112(f) software limitation to be indefinite because the specification provided "nothing more than a restatement of the function, as recited in the claim."[57] However, if the specification provides a more granular functional description of the software than the claim does—that is, if it discloses a series of functional steps that, when strung together, achieve the claimed function—then the specification discloses an algorithm. For example, in *Typhoon Touch Techs. v. Dell*, a claim recited the § 112(f) limitation "means for cross-referencing said responses with one

56. *See* Eon Corp. IP Holdings LLC v. AT&T Mobility LLC, 785 F.3d 616, 621–24 (Fed. Cir. 2015); Triton Tech, LLC v. Nintendo, Inc., 753 F.3d 1375, 1378–80 (Fed. Cir. 2014); Robert Bosch, LLC v. Snap-On Inc., 769 F.3d 1094, 1011–12 (Fed. Cir. 2014); Function Media, L.L.C. v. Google Inc., 708 F.3d 1310, 1317–19 (Fed. Cir. 2013); Dealertrack, Inc. v. Huber, 674 F.3d 1315, 1330 (Fed. Cir. 2012); ePlus, Inc. v. Lawson Software, Inc., 700 F.3d 509, 518–20 (Fed. Cir. 2012); Noah Sys., Inc., v. Intuit Inc., 675 F.3d 1302, 1311–19 (Fed. Cir. 2012); *In re* Aoyama, 656 F.3d 1293, 1296–1300 (Fed. Cir. 2011); Rembrandt Data Techs. v. AOL, 641 F.3d 1331, 1338–43 (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, at *3–*4 (Fed. Cir. May 7, 2010); Net MoneyIN, Inc. v. VeriSign, Inc., 545 F.3d 1359, 1366–67 (Fed. Cir. 2009); Encyc. Britannica, Inc. v. Alpine Elecs., Inc., No. 2009-1087, 2009 WL 4458527, at *3–*6 (Fed. Cir. Dec. 4, 2009); Blackboard, Inc. v. Desire2Learn Inc., 574 F.3d 1371, 1382–85 (Fed. Cir. 2008); Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1339–41 (Fed. Cir. 2008); Aristocrat Techs. Austl. Pty. Ltd. v. Int'l Game Tech., 521 F.3d 1328, 1332–38 (Fed. Cir. 2008).

57. *Finisar Corp.*, 523 F.3d at 1340. The Federal Circuit has identified one category of functional limitations in software claims that do not require the disclosure of any corresponding structure. In *Katz Interactive Call Processing Patent Litig. v. Am. Airlines, Inc.*, the Federal Circuit held that the claimed functions of "processing," "receiving," and "storing" did not require the disclosure of an algorithm even though the limitations were subject to § 112(f) because such functions "can be achieved by any general purpose computer without special programming" and thus "it was not necessary to disclose more structure than the general purpose processor that performs those functions." 639 F.3d 1303, 1316 (Fed. Cir. 2011). Subsequent Federal Circuit cases that have considered the *Katz* exception have called it "narrow," found it not to apply, and required the disclosure of an algorithm in the specification to avoid indefiniteness. *See EON Corp.*, 785 F.3d at 621–23 (Fed. Cir. 2015); Ergo Licensing, LLC v. CareFusion 303, Inc., 673 F.3d 1361, 1364 (Fed. Cir. 2012).

of said libraries of said possible responses."[58] The Federal Circuit upheld the claim as definite because the specification "recite[d] a four-step algorithm for computer-implemented cross-referencing, starting with the entry of a response, then a search for the entered response in a library of responses, then determination whether a match exists in the library, and then execution of an action if a match exists."[59] The specification described a series of four functional tasks, each of which was more granular than the functional task recited as a claim limitation. When strung together, these disclosed tasks formed an algorithm for accomplishing the claimed task.

Together, the adoption of an algorithm as software's corresponding structure in *WMS Gaming* and the subsequent cases addressing the indefiniteness of software claims with § 112(f) limitations that fail to disclose an algorithm demonstrate that the Federal Circuit has adopted a sui generis definition of structure in the software arts. In most arts, the Federal Circuit defines structure for the purposes of § 112(f) in terms of a technology's physical, spatial, and material properties,[60] but, in software, it rejects equating software's structure with these physical properties on two different levels. First, the Federal Circuit rejects the notion that the disclosure of a generic computer or processor in the specification amounts to the disclosure of corresponding structure.[61] However, standing alone, this rejection could be taken to mean that a generic computer or processor is not *enough* physical structure.[62] It leaves open the possibility that the required structure is the physical structure that distinguishes one software program from another, and the disclosure of a generic computer cannot mark this distinction.[63] Second, by embracing an algorithm as corresponding structure, the Federal Circuit's § 112(f) software cases implicitly reject physical structure altogether. An algorithm is not the same type of structure

58. Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376, 1383 (Fed. Cir. 2011) (citation omitted).

59. *Id.* at 1385.

60. *See supra* notes 32–36.

61. *See, e.g.*, *EON Corp.*, 785 F.3d at 621; *Aristocrat Techs.*, 521 F.3d at 1333; WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1349 (Fed. Cir. 1999).

62. The analogy in the mechanical arts to the disclosure of a generic computer as corresponding structure might be the disclosure of a material like metal as the corresponding structure. Being made of metal is a structural property of a mechanical invention, but it does not provide enough information about structure to say that the specification has disclosed the corresponding structure.

63. *Cf. supra* note 40 and accompanying text (proposing that software's most relevant physical structure is a distribution of electronics on a storage medium or an arrangement of open and closed gates in a computer).

considered in § 112(f) for other technologies. Whereas the structure of other technologies for purposes of § 112(f) is physical structure, the steps that make up an algorithm are still functional in a literal sense. For example, each of the steps in the *Typhoon Touch Technologies* algorithm—such as the step of "search[ing] for the entered response in a library of responses"—is just as functional as the claimed task—namely "cross-referencing said responses with one of said libraries of said possible responses."[64] The difference between the claimed function and the functions disclosed in the patent specification is only the level of granularity of the functional description: the disclosed algorithm is still a functional description of what the software does, just one that is more specific than the functional description provided in the claims.[65] This definition of software's structure not as physical structure but rather as logical or metaphorical structure makes software a sui generis technology under § 112(f).[66] In no technology other than software does a sufficiently specific functional description of a technology ever get counted as corresponding structure.[67]

64. *See supra* notes 58–59.

65. The language that the Federal Circuit employs to describe what it is doing, however, does not always overtly reflect its practice. The Federal Circuit occasionally juxtaposes the language expressing an algorithm with functional language, implicitly supporting the (incorrect) inference that an algorithm is something other than a functional entity. For example, in order to explain its invalidation of a software claim with a § 112(f) limitation for indefiniteness, the Federal Circuit stated in *Ergo Licensing v. CareFusion* that "[t]he specification merely provides functional language and does not contain any step-by-step process for controlling the adjusting means." 673 F.3d 1361, 1365 (Fed. Cir. 2012). By implying that an algorithm—i.e., the step–by–step process—is not a form of functional language, the Federal Circuit ignores its own definition of an algorithm and sweeps the sui generis nature of software under the rug. To avoid an outright contradiction, the Federal Circuit's language in passages like this one must be interpreted to mean that the specification cannot merely provide functional language at the same level of generality as the functional language in the claim limitation.

66. The adoption of metaphorical structure in the software arts means that we are talking about different ideas altogether when we discuss structure for the purpose of § 112(f) and structure for the purpose of the § 101 prohibition on claims to propagating signals or software per se, articulated in *In re Nuijten*, 500 F.3d 1346, 1356 (Fed. Cir. 2007) ("While such a signal is physical and real, it does not possess concrete structure."); *id.* at n.6 (noting that the signal on a "storage medium" would be patent eligible). The structure required by *In re Nuijten* refers to physical structure, but the structure needed to satisfy § 112(f) does not.

67. Thus, "'[s]tructure' to a person of ordinary skill in the art of computer-implemented inventions may differ from more traditional, mechanical structure." Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1298 (Fed. Cir. 2014). In this same passage, the Federal Circuit also states that "looking for traditional 'physical structure' in a computer software claim is fruitless because software does not contain physical structures." *Id.* This second statement is technically inaccurate. Software does not violate materialism; it does

IV. *WILLIAMSON* AND THE § 112(F) THRESHOLD TEST

The § 112(f) threshold test determines whether a claim limitation is subject to the default rule of claim construction articulated in *Phillips* or the scope–narrowing rule of claim construction specified in § 112(f). Since the middle of the 1990s, two competing theories of the reach of § 112(f) have struggled for dominance in the Federal Circuit: *intent theory* and *scope theory*. Neither has ever completely dominated the other, but the weight given to each has varied over time. For at least the decade prior to *Williamson*, intent theory had the upper hand. With *Williamson*, the Federal Circuit gave the upper hand to scope theory.

Intent theory suggests that § 112(f) is an option that Congress placed at the disposal of patent drafters and that § 112(f) should apply when, and only when, patent drafters use the word "means" as a signal that they intend to invoke it. Drawing on the literal text of the statute, intent theory posits that § 112(f):

> provides that an element in a claim for a combination 'may be expressed' as a means for performing a function, which indicates that the patentee is afforded the option of using the means-plus-function format. The question then is whether, in the selection of claim language, the patentee must be taken to have exercised that option.[68]

Thus, intent theory gives rise to twin presumptions in the threshold test: "the term 'means' (particularly as used in the phrase 'means for') generally invokes § [112(f)] and . . . the use of a different formulation generally does not."[69] At least on paper, these presumptions have always been rebuttable: enough structure accompanying the word "means" could take the limitation out of the ambit of § 112(f), and insufficient structure accompanying a non–means term could lead to the application of § 112(f).[70]

In contrast, scope theory dismisses the formalism of semantic word choice and focuses directly on whether the claim limitation recites enough structure so that the Supreme Court's policy concerns about overbroad

have physical structure. *See supra* note 40 and accompanying text. However, the Federal Circuit's statement captures the spirit of the law insofar as it implies that physical structure is and should be irrelevant in the § 112(f) analysis of software claims.

68. Greenberg v. Ethicon Endo-Surgery, Inc., 91 F.3d 1580, 1584 (Fed. Cir. 1996); *see also* York Prods., Inc. v. Cent. Tractor Farm & Family Ctr., 99 F.3d 1568, 1574 (Fed. Cir. 1996).

69. *Greenberg*, 91 F.3d at 1584.

70. *York Prods.*, 99 F.3d at 1574.

claim scope raised in *Halliburton Oil Well Cementing* are mitigated or eliminated.[71] That is, under scope theory, a claim limitation should be subject to § 112(f) whenever it fails to "recite a definite structure which performs the described function," regardless of the precise words employed.[72] Here, § 112(f) is mandatory, not optional, whenever a claim is drafted with excessively functional language.

By the 2000s, intent theory gained the upper hand in the Federal Circuit. The presumption against the application of § 112(f) to a limitation that did not use the term "means" evolved into a "strong" presumption[73] that could only be overcome by "a showing that the limitation essentially is devoid of anything that can be construed as structure."[74] The § 112(f) threshold test thus became highly formalistic, and the scope–limiting effect of § 112(f) could easily be avoided by any knowledgeable, motivated patent drafter.

In 2014, the conflict between intent and scope theories resurfaced in the disagreement between the Federal Circuit's majority and dissenting opinions in *Apple v. Motorola* over whether a claim limitation not using the term "means" was subject to § 112(f).[75] The majority adopted a strong version of intent theory. "The strong presumption created by not including means in a claim limitation . . . signals to the court that the patentee has chosen to . . . avoid[] the benefits of Section 112, ¶ 6."[76] In contrast, the

---

71. *See supra* notes 16–17 and accompanying text (discussing *Halliburton*'s invalidation of functional claims as overbroad).

72. Cole v. Kimberly-Clark Corp., 102 F.3d 524, 531 (Fed. Cir. 1996). Scope theory may have been quite novel in the § 112(f) threshold test in the 1990s. *See, e.g.*, Mas-Hamilton Grp. v. LaGard, Inc., 156 F.3d 1206, 1213–14 (Fed. Cir. 1998) (holding for the first time in the Federal Circuit that a phrase not using "means" was governed by § 112(f)); *cf.* Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1359–60 (Fed. Cir. 2004) (labeling *Mas-Hamilton* as an "exceptional case").

73. *Lighting World,* 382 F.3d at 1358. In fact, intent theory had become so strong that a claim limitation without the word "means" was not a § 112(f) limitation even if it was purely functional in the sense that it encompassed all structures that were capable of performing the claimed function. *Id.* at 1361–62.

74. Flo Healthcare Sols., LLC v. Kappos, 697 F.3d 1367, 1374 (Fed. Cir. 2012). Even under the strong–presumption formulation of the threshold test, the Federal Circuit's opinions are laced with scope–theory sound bites. *Lighting World*, 382 F.3d at 1360 (noting that § 112(f) does not apply to a term "that is simply a nonce word or a verbal construct that is not recognized as the name of structure and is simply a substitute for the term 'means for'"). However, very few claim limitations that did not employ the word "means" were held to actually be governed by § 112(f). *But see* Welker Bearing Co. v. PHD, Inc., 550 F.3d 1090, 1096–97 (Fed. Cir. 2008); MIT v. Abacus Software, 462 F.3d 1344, 1355 (Fed. Cir. 2006); *Mas-Hamilton*, 156 F.3d at 1213–14.

75. Apple Inc. v. Motorola, Inc., 757 F.3d 1286, 1297 (Fed. Cir. 2014).

76. *Id.*

dissent argued that, under the majority opinion, "one minor drafting decision greatly expands the scope of the claim limitation" and "patent applicants are able to claim broad functionality without being subject to the restraints imposed by § 112 ¶ 6."[77]

Only a year after *Apple v. Motorola*, the Federal Circuit acted en banc in *Williamson v. Citrix Online* to undermine the primacy of intent theory and elevate scope theory in the threshold test.[78] Drawing from the policy concerns about the overbreadth of functional claims that led the Supreme Court to invalidate functional claims in the first place, the Federal Circuit noted that a strong presumption against using § 112(f) when a claim limitation did not use the word "means":

> has the inappropriate practical effect of placing a thumb on what should otherwise be a balanced analytical scale. It has shifted the balance struck by Congress in passing § 112, para. 6 and has resulted in a proliferation of functional claiming untethered to § 112, para. 6 and free of the strictures set forth in the statute.[79]

The Federal Circuit proceeded to flatly state that the "heightened burden" for overcoming the presumption against the application of § 112(f) in the absence of the term means "is unjustified."[80] To articulate the new threshold test, the Federal Circuit reverted to scope theory, holding that the presumption against the application of § 112(f) in the absence of the word "means" could be overcome whenever "the words of the claim are understood by persons of ordinary skill in the art to have a sufficiently definite meaning as the name for structure."[81]

77. *Id.* at 1337 (Prost, J., concurring in part and dissenting in part).

78. Williamson v. Citrix Online, LLC, 792 F.3d 1339 (Fed. Cir. 2015) (en banc). Only the section addressing the proper standard for the § 112(f) threshold was authored en banc.

79. *Id.* at 1349.

80. *Id. But see id.* at 1358 (Newman, J., dissenting) ("[I]t is the applicant's choice during prosecution whether or not to invoke paragraph 6, and the court's job is to hold the patentee to his or her choice.").

81. *Id.* at 1349. Presumably, the default claim construction methodology for non–112(f) limitations outlined in *Phillips v. AWH*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), should govern the search for sufficient structure in the § 112(f) threshold test. *See, e.g.*, Personalized Media Commc'ns, LLC v. U.S. Int'l Trade Comm'n, 161 F.3d 696, 704 (Fed. Cir. 1998) (stating that, in the § 112(f) threshold test, "the focus remains on whether the claim as properly construed recites sufficiently definite structure to avoid the ambit of § 112, ¶ 6"). The use of *Phillips* here makes sense as a policy matter. The Supreme Court's concerns about overbroad functional claims are only mitigated when the scope of the claim in infringement proceedings has sufficient structural limitations, and *Phillips* controls claim scope for infringement purposes. However, the Federal Circuit often speaks of the

However, *Williamson* does not represent a complete triumph of scope theory over intent theory. The Federal Circuit did not abandon the use of presumptions in the § 112(f) threshold test. The presumption that § 112(f) does not apply to limitations without the term "means" endures, and *Williamson* only makes it easier to rebut.[82] The converse, pre–*Williamson* strong presumption that the use of the word "means" triggers § 112(f) remains in place.[83]

V. THE NEED FOR A REVOLUTION

At first glance, *Williamson* might not appear to call for a revolution in the § 112(f) threshold test for software claims. While *Williamson*'s turn to scope theory will force the Federal Circuit to ask and answer the definitional "What is structure?" question as part of the threshold test on a more frequent and open basis, the Federal Circuit has already discussed software's structure in § 112(f) cases in two distinct doctrinal contexts in its pre–*Williamson* opinions. First, it has entertained arguments about the presence or absence of structure in claim limitations that were intended to rebut the

search for sufficient structure in the threshold test not in the exact terms outlined in *Phillips*, but rather in different terms that mirror *Phillips* only in a rough manner. For example, Federal Circuit cases do not usually ask whether a claim limitation has a "meaning" that is sufficiently structural, which is the usual way of talking about claim construction, but rather whether a limitation "connotes" sufficient structure. *See, e.g.*, Apex Inc. v. Raritan Comput., Inc., 325 F.3d 1364, 1373 (Fed. Cir. 2003) ("The threshold issue for all the limitations involving the term 'circuit' is whether the term itself connotes sufficient structure to one of ordinary skill in the art to perform the functions identified by each limitation."). A connotation standard may identify sufficient structure to avoid § 112(f) even when that structure does not amount to a claim limitation under *Phillips*. In some cases, the Federal Circuit seems far more willing to read structure from the specification into the claims as part of the search for sufficient structure in the threshold test than it would be to read limiting features of preferred embodiments into the claim limitations under *Phillips*. *See, e.g.*, *infra* note 113 and accompanying text (discussing *Apple*, 757 F.3d at 1300). This willingness to rely on the specification to identify sufficient structure in a claim limitation may have resulted, in part, from the now–defunct strong presumption against the invocation of § 112(f) when the limitation does not employ the term "means." It is unclear whether this willingness will persist after *Williamson*.

82. *Williamson*, 792 F.3d at 1349. A *Williamson* concurrence notes that the majority does not clarify what force the not–strong presumption against § 112(f) that follows from the absence of "means" has after *Williamson*. *Id.* at 1357 n.2 (Reyna, J., concurring in part and dissenting in part). Earlier Federal Circuit opinions suggest that the burden of persuasion is a preponderance of the evidence. *See, e.g.*, *Apex*, 325 F.3d at 1372. But, whether *Williamson* returns to this burden is unclear.

83. *Williamson,* 792 F.3d at 1349.

pre–*Williamson*, strong presumptions of the threshold test.[84] Second, after limitations have been labeled as § 112(f) limitations, it has looked for corresponding structure in the specification in the form of algorithms. However, upon closer examination, neither type of pre–*Williamson* case provides much guidance for courts trying to answer the definitional question about software's structure as part of the threshold test. Starting with the better–developed body of law, Section V.A demonstrates that the Federal Circuit's existing algorithm jurisprudence cannot identify structure as part of the threshold test because an algorithm is a relational entity that can only be discerned by comparing a functional description of how a software program works to a claim limitation. Section V.B then looks for crumbs of wisdom in the Federal Circuit's pre–*Williamson* cases that address the definitional question as part of the strong–presumption threshold test.

A. PRECEDENT ON ALGORITHMS AS CORRESPONDING STRUCTURE

There are two distinct problems with using the Federal Circuit's definition of an algorithm that was developed to identify corresponding structure in the specification during claim construction in order to identify sufficient structure in the claims as part of the § 112(f) threshold test. First, § 112(f) has traditionally required each limitation, considered independently, to recite sufficient structure, but no single, functionally defined claim limitation can ever be an algorithm. Second, even if the § 112(f) threshold test is altered so that it looks for an algorithm in a series of functional claim limitations, the inherently relative nature of the Federal Circuit's definition of an algorithm leads to an unworkable conceptual muddle.

No single, functional claim limitation can ever be an algorithm because identifying a single claim limitation as an algorithm involves a category error. Functional claim limitations are singular, and algorithms are plural. As the Federal Circuit has defined the concept, an algorithm is a sequence of steps for performing a task.[85] Only a series of steps, performed one after the other, can be an algorithm. A single–step procedure for performing a task is simply a restatement of the task, not an algorithm. Thus, a judge or

84. All pre–*Williamson* § 112(f) software cases have employed the strong presumptions in the threshold test. *WMS Gaming*—the case in which the Federal Circuit first grappled with the application of § 112(f) to software—was decided only five years before the Federal Circuit articulated the strong presumptions, and there were no cases in that five–year window that employed scope theory to animate the § 112(f) threshold test in software cases. *See* WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339 (Fed. Cir. 1999).

85. *See supra* notes 49–55 and accompanying text.

examiner cannot ask whether an individual claim limitation recites structure in the form of an algorithm.

Rather, if the plan is to carry the Federal Circuit's algorithm–as–structure analysis over from its precedent on identifying corresponding structure in a specification and use it to identify sufficient structure in a claim, judges and examiners must break with precedent in a different way and ask whether a claim stating a series of functional limitations constitutes an algorithm.[86] This approach to identifying structure in the threshold test would require a significant doctrinal shift, as the threshold test usually examines whether each functional limitation recites sufficient structure on its own, not whether a series of functional limitations collectively recites sufficient structure. In fact, even when a claim states that the same "means" performs more than one function, each function is supposed to be analyzed independently to determine whether it has sufficient structure to evade the strictures of § 112(f).[87] However, searching for structure in an aggregate of functional limitations is the only feasible approach if the Federal Circuit's well–established algorithm analysis is to be used to answer the definitional "What is structure?" question of the threshold test in software claims.

Yet, even if one is willing to alter the search for structure in this software–specific manner, the Federal Circuit's algorithm case law is decidedly unhelpful in the threshold test. The crux of the problem is that the Federal Circuit has only defined an algorithm in a relative manner. More specifically, it has only defined an algorithm in the specification in relation to a particular claim limitation. A specification's functional description of a software invention is an algorithm if the specification describes how the software operates in a manner that is *more granular* than the baseline functional description of the software in the claim limitation.[88] One cannot identify an algorithm under the Federal Circuit's definition without first

86. This question assumes that the claim recites more than one functional limitation. If a claim recites only a single limitation and that limitation is functional, then the claim is invalid. Section 112(f) only saves functional limitations in combination claims from invalidity under *Halliburton Oil Well Cementing Co. See In re* Hyatt, 708 F.2d 712, 712–15 (Fed. Cir. 1983). However, a claim may recite a single functional software limitation in combination with one or more non–functional limitations. It is not clear that the single functional limitation in such a claim could ever be held to recite sufficient structure to avoid § 112(f) when structure is equated with an algorithm.

87. *See, e.g.*, Media Rights Techs. v. Capital One Fin. Corp., 800 F.3d 1366, 1373 (Fed. Cir. 2015); Noah Sys., Inc. v. Intuit Inc., 675 F.3d 1302, 1318–19 (Fed. Cir. 2012).

88. *See supra* notes 57–59 and accompanying text.

having identified the baseline task in the claim that the algorithm is supposed to perform.

To appreciate the relative nature of the Federal Circuit's definition of an algorithm, consider the impossibility of examining a specification in isolation from the claims and meaningfully opining on whether it describes corresponding structure. The exact same specification can provide corresponding structure for one claim and yet fail to provide corresponding structure for another claim. For example, consider a hypothetical patent on a digital rights management technology.[89] Assume that the specification states that the invention performs three functions: it controls a client system's data output by diverting a commonly used data pathway of a media player application to a controlled data pathway, monitors the controlled data pathway, and disrupts media content playback at the controlled data pathway when playing the media file content is outside of the usage restriction applicable to the media file.[90] Is this functional description of how the invention works an algorithm? The answer depends on the claim at issue. Assume that claim 1 recites a single, umbrella limitation such as "a compliance mechanism for ensuring that only media content within the usage restriction applicable to the media is played." Here, the specification likely does describe structure in the form of an algorithm because it provides a series of more granular steps that explain how to perform the less granular task recited as the claim limitation. Now, assume that claim 2 recites three separate functional limitations that parallel the three disclosed functions. That is, assume that claim 2 recites separate "mechanism for controlling," "mechanism for monitoring," and "mechanism for disrupting" limitations. The unchanged specification no longer describes structure or an algorithm. Rather, it merely restates the functional tasks that are recited as claim limitations.[91] Under the Federal Circuit's relational definition of an algorithm, the existence of an algorithm, and thus corresponding structure, is contingent on the baseline of the functional description provided by the claims. A judge or examiner cannot look at a description of software functionality, standing alone, and reach the conclusion that the description describes structure.

89. In this hypothetical, the technology and language, but not the legal analysis, is based on the patent at issue in *Media Rights Technologies v. Capital One Financial Corp*, 800 F.3d 1366 (Fed. Cir. 2015).

90. *Cf. id.* at 1369–70.

91. *See supra* note 57 and accompanying text (noting that a restatement of the functional claim limitations is not an algorithm).

The Federal Circuit's relational definition of corresponding structure for software limitations does not map neatly onto the nature of corresponding structure in other technologies. In the mechanical arts, the answer to the definitional question is never relational or contingent. Any given property of a mechanical invention is either a structural or a functional property; a three–legged stool is not structure with respect to one claim but function with respect to another. There may be relativity when determining whether any given structure in the specification can perform the claimed function. It may well be that a three–legged stool is corresponding structure for a "means for supporting" limitation, but not for a "means for cutting" limitation. However, there is no relativity in the more fundamental definitional question of what constitutes structure in the first place.

The Federal Circuit's relational definition of structure in software is conceptually workable, if a bit awkward, when it is put to work in the context of identifying corresponding structure in the specification during claim construction. Because the search for structure in the specification is always conducted after a claim limitation has been held to be subject to § 112(f), the functional description recited in the claim is always available to serve as a baseline.[92] However, when searching for sufficient structure in

92. The Federal Circuit's relational definition of structure in software is awkward even when looking for corresponding structure because it places a formalistic, rather than substantive, limit on permissible patent scope. A relational definition of structure does not cap permissible claim scope at any particular level of generality, but instead mandates that every functional claim limitation be narrowed to a disclosed algorithm and its equivalents, regardless of the level of specificity at which the limitation is drafted. *See* Collins, *supra* note 8, at 1463–67. In an ideal patent regime that imposes substantive limits on what can be patented, two claims that are coextensive—that is, two claims that describe the same set of technologies—should be subjected to the same restrictions because they raise identical policy concerns. However, when operating on the basis of a relational definition of corresponding structure, § 112(f) does not achieve this goal. For example, if a first claim limitation recites function A and the specification recites algorithmic steps 1, 2, and 3 for function A, the claim containing the limitation is valid only if limitation A is restricted to the "structure" of steps 1, 2, and 3, as well as its equivalents. But, if a second claim were to directly recite functions 1, 2, and 3—which are identical to steps 1, 2, and 3—as separate limitations, § 112(f) would also require that each limitation in this claim be restricted in scope to the more granular, algorithmic steps (or sub–algorithmic steps, depending on your perspective) disclosed in the specification. Functions that are corresponding structure when recited as the steps of an algorithm in the specification cease to be structure when they are recited as a series of claim limitations. The limitation reciting function 1 would need to be restricted to something like the "structure" of sub–steps 1a, 1b, 1c, and its equivalents. To be clear, even the formalistic limitation on permissible patent scope that follows from the Federal Circuit's relational definition of an algorithm in § 112(f) does important work when software claims employ extremely general functional limitations. In some situations, however, it may not do enough work. It may narrow an extremely general claim to a very

a claim limitation as part of the § 112(f) threshold test, the relational definition of structure is conceptually bankrupt. There is no baseline to which to compare the claim limitation, and a relative definition is useless without a baseline. Just as it is impossible to know whether functional language in a specification constitutes an algorithm without consulting the baseline functional task provided by the claims, it is impossible to know whether functional language in a claim constitutes an algorithm without consulting a baseline task of some kind. The claim cannot provide this baseline. It is nonsensical to ask whether the claim limitations are steps in a step–by–step procedure for performing the functions specified in the claim limitations.[93]

When a trained patent lawyer looks at a software claim with a series of functional limitations, she may initially believe that she can use the concept of an algorithm to identify structure in that series of limitations during the § 112(f) threshold test. For example, presented with the hypothetical claim 1 (including the limitation "a compliance mechanism for ensuring that only media content within the usage restriction applicable to the media is played") and claim 2 (including limitations reciting "mechanism for controlling," "mechanism for monitoring," and "mechanism for disrupting"), she may see structure in claim 2 but not claim 1. The underlying intuition is that the limitations of claim 2 form an algorithm for performing the task listed in claim 1. But, this intuition adopts the function recited in claim 1 as the baseline task to be performed without any justification for doing so. Every series of functional limitations is a step–by–step process for doing something. Unless every series of functional limitations in a software claim recites an algorithm and thus sufficient structure to avoid the application of § 112(f), the pre–*Williamson* definition of an algorithm cannot be used to identify sufficient structure in the post–*Williamson* threshold test.

B. PRECEDENT ON THE PRE–*WILLIAMSON* THRESHOLD TEST

Before *Williamson*, the Federal Circuit occasionally entertained arguments addressing whether the amount of structure in a claim limitation was sufficient to rebut the strong presumptions of the § 112(f) threshold test. In principle, the reasoning in these arguments could provide insight

general, but still overbroad, claim. In other situations, it may counterproductively do too much work. It may further narrow an already sufficiently narrow claim.

93. *But cf. infra* note 121 and accompanying text (considering the possibility that a claim could recite both a task and a series of steps for accomplishing that task).

into a definition of software's structure for the post–*Williamson* threshold test. In actuality, however, they do not. Some pre–*Williamson* opinions use conclusory reasoning, baldly asserting a generic, programmed computer is sufficient structure to avoid § 112(f).[94] As explored below, other pre–*Williamson* opinions employ reasoning that is either misguided or protean. Subsection V.B.1 notes, and dismisses as unwise, the pre–*Williamson* cases in which software's structure is defined in terms of physical structure. Subsection V.B.2 turns to a sparse line of cases suggesting that functional claim limitations describing an operation's input, output, connections, or how its output may be achieved should count as logical structure. This definition of software's structure is radically underdeveloped, but it hints at what would be needed to formulate a stand–alone definition of software's logical structure for the post–*Williamson* § 112(f) threshold test.

1. *Circuit as Physical Structure*

A number of the Federal Circuit's pre–*Williamson* cases which found sufficient structure in a claim limitation lacking the term "means" to adhere to the presumption against the application of § 112(f) in the threshold test derive from a single precedent: *Apex Inc. v. Raritan Computer, Inc.*[95] The *Apex* claim contained a "programmed logic circuit" limitation for performing a variety of functions,[96] and the Federal Circuit concluded that the presumption against § 112(f) had not been overcome because a circuit requires a "conducting path" and thus entails physical structure: "The term 'circuit' is defined as 'the combination of a number of electrical devices and conductors that, when interconnected to form a conducting path, fulfill some desired function.'"[97] Citing *Apex* as precedent, the Federal Circuit has found enough structure in a number of other software limitations lacking the word "means" to conclude that the presumption against the application

94. *See, e.g.,* LG Elecs., Inc. v. Bizcom Elecs., Inc., 453 F.3d 1364, 1372 (Fed. Cir. 2006) ("The claim itself provides sufficient structure, namely 'a CPU and a partitioned memory system,' for performing the stated function, 'controlling the communication unit.'"), *rev'd on other grounds sub nom.* Quanta Comput., Inc. v. LG Elecs., Inc., 553 U.S. 617 (2008).

95. Apex Inc. v. Raritan Comput., Inc., 325 F.3d 1364, 1373–74 (Fed. Cir. 2003).

96. *Id.* at 1368.

97. *Id.* at 1373. For additional definitions of "circuit" that require an electrical pathway, see *Linear Technology Corp. v. Impala Linear Corp.*, 379 F.3d 1311, 1320 (Fed. Cir. 2014).

of the § 112(f) threshold test had not been overcome. Some of these limitations used the term "circuit,"[98] but others did not.[99]

Whatever merit this invocation of the physical structure of a circuit has when patent claims describe relatively simple hardware devices, it should be irrelevant to a definition of software's structure in a post–*Williamson* threshold test based on scope theory. The arbitrary nature of the Federal Circuit's "circuit" doctrine is on full display in *MIT v. Abacus Software*.[100] The patent at issue contained two functional limitations: an "aesthetic correction circuitry" for performing a function[101] and a "colorant selection mechanism" for performing a function.[102] Addressing the former limitation, the Federal Circuit identified enough structure for the presumption against § 112(f) not to be overcome. "[T]he term 'circuitry,' by itself, connotes sufficient [physical] structure," and, laundering function into additional structure, the "aesthetic correction" language "adds further structure by describing the operation of the circuit."[103] In contrast, the presumption against § 112(f) was overcome for the latter limitation. "The term 'mechanism' standing alone connotes no more structure than the term 'means,'" and, refusing to launder function into structure, the "colorant selection" modifier did not have a generally understood meaning in the art that connoted structure, either.[104] Here, the Federal Circuit's "circuit" doctrine is simply an example of "the doctrine of magic words" in action.[105] There is nothing that is meaningfully more structural about the "aesthetic

98. *See, e.g.*, *id.* at 1320–21; Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc., 711 F.3d 1348, 1365 (Fed. Cir. 2013); MIT v. Abacus Software, 462 F.3d 1344, 1353–55 (Fed. Cir. 2006).

99. Inventio AG v. ThyssenKrupp Elevator Am. Corp., 649 F.3d 1350, 1358 (Fed. Cir. 2011).

100. *MIT*, 462 F.3d at 1353–57.

101. *Id.* at 1348 (describing an "aesthetic correction circuitry for interactively introducing aesthetically desired alterations into [the] appearance signals to produce modified appearance signals").

102. *Id.* (describing a "colorant selection mechanism for receiving said modified appearance signals and for selecting corresponding reproduction signals . . . to produce . . . a colorimetrically-matched reproduction").

103. *Id.* at 1355–56. The dissent argued that the limitation was a § 112(f) limitation. *Id.* at 1360–65 (Michel, C.J., dissenting). The majority's rejoinder fell back on the strong presumptions. *Id.* at 1356 ("[T]he dissent appears to misapprehend the strength of the presumption that applies when the term 'means' does not appear in the claim.").

104. *Id.* at 1353–55.

105. Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CALIF. L. REV. 1, 9 (2001) (describing a rule under which software was patentable so long as the applicant recited "magic words and pretended that she was patenting something else entirely").

correction circuitry" limitation than the "colorant selection mechanism" limitation.

2. *Inputs and Outputs as a Clue to Logical Structure*

In his majority opinion in *Apple v. Motorola*, issued only a year before *Williamson*, Judge Reyna held that a functional software limitation without the word "means" recited enough structure so that the presumption against § 112(f) was not overcome because it described the "input" and "output" of the software, as well as "how its output may be achieved."[106] With this reasoning, Judge Reyna laid some preliminary groundwork for a stand–alone definition of logical structure in software—a definition that can be used to identify structure in a functional description of how software works without employing some other functional description as a baseline.

Among the many patents and issues addressed, *Apple v. Motorola* upheld several apparatus claims on the use of finger contacts to change the visual field or select files on a touchscreen computer.[107] Each of the claims described one or more "heuristics" for determining that certain finger contacts correspond with certain commands.[108] The district court construed the term "heuristics" to mean "one or more rules to be applied to data to assist in drawing inferences from that data."[109] It then held the heuristics limitations to be subject to § 112(f) under the threshold test, despite the then–present strong presumption to the contrary, because the claims described functions "without describing the structure necessary to perform the functions."[110]

On appeal, Judge Reyna's majority opinion reversed and held that the heuristics limitation was not subject to § 112(f) because the limitation recited "the heuristics' operation within the context of the invention, including the inputs, outputs, and how certain outputs are achieved."[111] In

106. Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1300 (Fed. Cir. 2014).

107. *Id.* at 1294–1304. The *Apple* opinion is much better known for its implications for injunctions for standard–essential patents. *See id.* at 1331–32.

108. *Id.* at 1295 (noting, for example, that one limitation recited "a vertical screen scrolling heuristic for determining that one or more finger contacts correspond to a one-dimensional vertical screen scrolling command rather than a two-dimensional screen translation command *based on an angle of initial movement of a finger contact with respect to the touch screen display*").

109. *Id.* Judge Posner sat by designation as the district court judge. *Id.* at 1294.

110. *Id.* at 1295.

111. *Id.* at 1301; *see also id.* at 1299 ("Structure may . . . be provided by describing the claim limitation's operation, such as its input, output, or connections."); *id.* at 1300 (stating that the heuristics limitation "describes the limitation's operation, including its

some ways, this inputs–and–outputs line of reasoning is problematic. Its support in Federal Circuit precedent is weak.[112] More importantly, the claim at issue arguably did not actually include any reference to how outputs are achieved as a limitation on claim scope. Judge Reyna relied on features of the preferred embodiments disclosed in the specification that do not actually limit claim scope to find that the claim language connoted sufficient structure.[113] However, abstracting from the actual facts of the case, Judge Reyna's opinion is interesting because it suggests one factor that might be relevant in developing a stand–alone definition of software's structure. The gist of Judge Reyna's reasoning is that functional claim language should count as a description of the logical structure of software if it is sufficiently specific that it describes how the invention works—that is, if it describes "inputs, outputs, and how certain outputs are achieved"—rather than what the invention does for its user.[114] This approach to answering the threshold

---

input, output, and how its input may be achieved"). Judge Reyna also argued "that 'heuristic' has a known meaning" that connotes structure, but he eventually concluded that "[w]e need not decide here whether the term 'heuristic,' by itself, connotes sufficient structure to maintain the presumption against means-plus-function claiming" because the inputs–and–outputs argument was sufficient to demonstrate that the strong presumption against § 112(f) had not been rebutted. *Id.* at 1300–01.

112. *Apple* cites two cases to support looking to the "input, output, and connections" of a program as a clue for identifying structure in a claim limitation. *Id.* at 1299. The first case, *Linear Technology Corp. v. Impala Linear Technology Corp.*, held that the presumption against § 112(f) was not overcome principally because the term "circuit" connoted tangible structure. 379 F.3d 1311, 1320–21 (Fed. Cir. 2014); *see also supra* Subsection V.B.1. However, it did also note in passing that the term "circuit" became more structural when coupled in the claim with a description of the circuit's "operation" and its "desired output." *Linear*, 379 F.3d at 1320. The second case, *Lighting World, Inc. v. Birchwood Lighting, Inc.*, held that the presumption against § 112(f) was not overcome principally because the term "connector" had a generally understood meaning in the art as a name for physical structure. 382 F.3d 1354, 1361–62 (Fed. Cir. 2004).

113. The claim language clearly recited the heuristic's "objectives." *Apple*, 757 F.3d at 1302. Beyond that, however, all of the opinion's support for the inputs, outputs, and connections as claim limitations drew from the written description. *Id.* at 1302–03. One of the dissent's critiques in *Apple* was that the majority relied too heavily on the preferred embodiments in the specification to identify structure in the claims. *Id.* at 1335 (Prost, J., concurring in part and dissenting in part). This type of disagreement should not be surprising given that the Federal Circuit has not even been clear about whether *Phillips* governs the search for sufficient structure in § 112(f) threshold test. *See supra* note 81.

114. *Apple*, 757 F.3d at 1301. The language used to convey this reasoning, however, is at times misleading. As the opinion puts it, "[t]he limitation's operation is more than just its function; it is how the function is achieved in the context of the invention." *Id.* at 1299. This passage overlooks the fact that a description of "how the function is achieved in the context of the invention" is itself inevitably a description of function. *See supra* note 65 and accompanying text (noting that the Federal Circuit sometimes misleadingly juxtaposes

test's definitional question is underdeveloped, but it at least hints at a way of defining software's structure that could work in the post–*Williamson* § 112(f) threshold test.[115]

VI. WHICH REVOLUTION?

Williamson's shift to a threshold test based on scope theory mandates a revolution in the law of functional claiming as applied to software because pre–*Williamson* opinions are of little use when identifying software's structure as part of the § 112(f) threshold test. However, it is not yet clear which of a number of possible revolutions *Williamson* will yield. The remainder of this Essay considers four possibilities. The first three stick with the Federal Circuit's relational definition of an algorithm and attempt to resolve the problem that this definition creates in different, unsatisfying ways.[116] The fourth possible revolution charts a new course: following the lead of *Apple v. Motorola*, the Federal Circuit could articulate a new, stand–alone definition of software's structure for the purposes of § 112(f).

A. THREE UNDERWHELMING REVOLUTIONS

A first possible revolution is that all software claims with multiple functional limitations will be held to recite algorithms and avoid the strictures of § 112(f). All claims with multiple functional limitations recite an algorithm insofar as they present a series of steps for accomplishing the task of providing the patented technology's end–user with some utility.[117] However, this outcome is unlikely in light of the expectation that a tilt

algorithms with functional descriptions). Judicial opinions would be more conceptually accurate if they were to say that a description of inputs, outputs, and how certain outputs are achieved is a type of functional description that counts as metaphorical structure in software because it serves the needed policy of limiting claim scope.

115. At least one district court has used *Apple*'s inputs–and–outputs reasoning to find sufficient structure in a claim limitation lacking the word "means" to avoid § 112(f) status after *Williamson*. *See* Finjan, Inc. v. Proofpoint, Inc., No. 13-cv-05808-HSG, 2015 WL 7770208, at *11 (N.D. Cal. Dec. 3, 2015) ("The term 'content processor' has a sufficiently specific structure. Independent claim 1 describes how the 'content processor' interacts with the invention's other components (the transmitter and receiver), which informs the term's structural character . . . . [T]he intrinsic evidence establishes the structural character of 'content processor' through its interaction with the system's other components.").

116. The first and third of these possible revolutions must accept that the search for software's structure does not require a separate search for sufficient structure within each functional limitation but rather a search for structure in a series of limitations. *See supra* notes 85–87 and accompanying text.

117. Claims with only a single functional limitation would not recite structure under this approach. *Cf. supra* note 86 (noting the invalidity of single–means claims).

toward scope theory in the threshold test should increase, not decrease, the number of software limitations subject to § 112(f).

A second possible revolution inverts the outcome of the first: perhaps all functional software limitations will be governed by § 112(f). A claim cannot be an algorithm for itself[118] So, perhaps all functional software limitations will be labeled as structureless tasks and no functional software limitations will connote sufficient structure. This route forward satisfies the expectation that the turn to scope theory in *Williamson* should increase the number of software limitations subject to § 112(f). In fact, it would lead to a tidal shift in the law of software patents. Before *Williamson*, most functional software limitations using "nonce" words were not subject to § 112(f), but after *Williamson*, all such limitations would be.[119]

A third possible revolution amounts to a mandate for a new style for drafting software claims. Perhaps a software claim recites an algorithm, and thus sufficient structure, if it initially recites both a general task as a limitation and then a more specific, step–by–step process for achieving that task. This approach bootstraps the baseline task that is needed to use the concept of an algorithm to identify structure into an earlier part of the claim. To continue the hypothetical addressed above, consider a claim to a digital rights management technology reciting "mechanism for controlling," "mechanism for monitoring," and "mechanism for disrupting" as separate limitations.[120] Imagine that the claim also includes a statement of the overall task to be performed, perhaps as part of the preamble, such as "a compliance mechanism for ensuring that only media content within the usage restriction applicable to the media is played." Now, a relative definition of an algorithm could get some traction in the threshold test by comparing two different parts of the same claim. Perhaps claims that explicitly state a baseline task contain structure in other limitations if those other limitations constitute a step–by–step procedure for performing that task.[121]

118. *See supra* text accompanying note 93.

119. This revolution would extend the formalism of the pre–*Williamson* law: all claims would be pushed down one step on the ladder of generality, regardless of whether they are initially drafted in a general or specific manner. *See supra* note 92.

120. *See supra* notes 90–91 and accompanying text.

121. In substance, this third revolution in the required form of a functional software claim is not much different from the second revolution under which every functional software limitation is subject to § 112(f). Rather than narrowing the scope of the broad, functional limitation to the algorithm disclosed in the specification (and its equivalents), it narrows the scope of the claim to the algorithmic steps recited as claim limitations.

B. A FOURTH REVOLUTION: DEVELOP A STAND–ALONE DEFINITION OF SOFTWARE'S STRUCTURE

A fourth possible revolution takes a decidedly different tack on the problem. Following the lead of *Apple v. Motorola*, the Federal Circuit could develop the stand–alone definition of software's structure that is needed for the threshold test to draw a principled line that subjects some, but not all, functional software limitations to § 112(f) based on the limitations' absolute breadth. In *Williamson*, the Federal Circuit recognized that the policy concerns about overbreadth expressed in the Supreme Court's functional–claiming cases from the first half of the twentieth century require a threshold test motivated by scope theory.[122] When there is enough structure recited in a claim's limitations, the number of "other devices beyond our present information or indeed our imagination which will perform [the claimed] function and yet fit [the] claims" is sufficiently reduced to ward off the most serious social costs of functional patents.[123] The same policy concerns should frame a stand–alone answer to the sui generis definitional "What is structure?" question of the threshold test in software. At what level of claim specificity are the restraints on competition engendered by functional software claims acceptable because the claim is restricted in scope to particular means for achieving a useful end?

The basic premise of a stand–alone definition of structure in software must be a line on a ladder of generality or abstraction. Functional descriptions of a software invention exist at many different levels of generality, ranging from extremely specific (e.g., an in–depth description of modules and their interaction) to extremely general (e.g., a description of the utility enjoyed by the end–user of the software). A stand–alone definition of structure must identify the point on the ascent of this ladder at which a granular description of a software program that constitutes logical structure transitions into a highly general, structureless description of a software program.[124] The descriptions both above and below the line are technically functional descriptions in a linguistic sense, but descriptions at

122. *See supra* notes 16–17 and accompanying text.

123. Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 12 (1946).

124. In a sense, the term "stand–alone structure" is misleading. The identification of structure still requires a comparison to a baseline: Is the functional description above or below the point at which the transition occurs? However, this baseline is fixed and always available. Unlike a comparison to the baseline provided by a claim, this baseline does not shift dramatically as the claim language varies or disappears altogether when the search focuses on finding structure in that very claim.

or below the line count as metaphorical or logical structure for the purposes of § 112(f).

In his work on functional claiming in software, Mark Lemley alludes to a level–of–generality analysis when he argues that § 112(f) must make a distinction between a "goal" (not structure) and a "way of implementing a goal" (structure),[125] or a "problem" the patentee solved (not structure) and "what the patentee . . . actually did" to solve the problem (structure).[126] The difficulty with these distinctions is that goals and problems can be expressed in functional language at many different rungs on the ladder of abstraction. Your "goal" is likely to be part of my way of implementing a different "goal." A software inventor may be motivated by the "problem" that two modules interact in an inefficient manner or by the "problem" that the parties to a transaction face settlement risk.[127] What is needed in the post–*Williamson* world is a stand–alone definition of what constitutes a general, structureless problem or goal and, inversely, what constitutes a structural way of implementing a goal or a patentee's actual solution to a problem. Simply defining an implementation in relation to a goal is not enough if what constitutes a goal remains a moving target.

Ideally, a stand–alone definition of structure should govern all inquiries into the structure of a software invention under § 112(f). *Williamson* specifically addresses only the § 112(f) threshold test, so the most immediate repercussion of *Williamson* is understandably a need for a new definition of software's structure that works in this context. However, once a stand–alone definition of structure has been developed, there is no reason not to use it to identify the corresponding structure of § 112(f) limitations in the specification, as well.[128]

125. Lemley, *supra* note 8, at 947.

126. *Id.* at 963.

127. *Cf.* Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014) (holding that a claim to a computer–executed method of reducing settlement risk is patent–ineligible). The interplay between the Federal Circuit's *Williamson* opinion and the Supreme Court's *Alice* opinion raises interesting questions that are beyond the scope of this Essay. Nonetheless, it is worth noting that a stand–alone definition of software's structure developed to implement *Williamson* might shed some light on when a claim "purport[s] to improve the functioning of the computer itself," and thus when a claim is patent–eligible even if it is directed to an abstract idea. *Id.* at 2359. Perhaps improvements in logical structure are examples of improvements in the functioning of the computer itself.

128. Technically, the Federal Circuit could adopt a stand–alone definition of structure for the threshold test and continue to use its relational definition of structure to identify corresponding structure in the specification. However, once the stand–alone definition has been formulated, it makes little sense not to use it in both contexts. A stand–alone definition

Drawing the line that is needed to identify logical structure in the software arts will not be an easy task. The echoes of Learned Hand's levels–of–generality test for drawing the idea/expression dichotomy in copyright are clear, and that test is notorious for its lack of ex ante clarity.[129] In a yet closer parallel, the Federal Circuit shelved the project of bringing § 112(f) to bear on step–plus–function limitations in method claims, despite the statutory requirement to do so, because of the difficulty of developing a stand–alone definition of an "act."[130] In method claims, there is no clear, intuitive difference between those actions that constitute steps and those that constitute acts.[131] The only difference between them is the position of a functional description of an action on a ladder of generality. An act describes a more specific functional task than a step does. When is a functional operation recited as a claim limitation sufficiently specific to qualify as an act rather than a step? The Federal Circuit punted on this question because of its difficulty.[132] Yet, this question closely parallels the question that the Federal Circuit must answer in order to create a stand–alone definition of software's structure.

The line marking a binary legal distinction between general and specific software–implemented functions on the ladder of abstractions has no "natural" position determined by extra–legal concerns. Nor is it feasible to ask the ultimate economic question—at what level of abstraction does a

of structure in the context of the search for corresponding structure in the specification would eliminate the formalism of lowering the permissible level of generality of a functional claim by one rung regardless of the claim's initial breadth. *See supra* note 92.

129. Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930). The actual copyright doctrine that has developed to administer the idea/expression dichotomy in software cases is not helpful in patent law. In copyright, the idea/expression dichotomy seeks, among other goals, to prevent copyright from granting an author control over software functionality at any level of abstraction. Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992). In patent law, § 112(f) must draw a line between claims to functionality that are sufficiently specific and those that are excessively general.

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

131. In an extended concurrence addressing step–plus–function claims, Judge Rader recognized that the distinction between a step and an act is "inherently more problematic" than the distinction between a function and a structure, at least in non–software claims. Seal-Flex, Inc. v. Athletic Track & Court Const., 172 F.3d 836, 848–49 (Fed. Cir. 1999) (Rader, J., concurring).

132. Judge Rader justified this punt by leaning heavily on intent theory to create an even stronger–than–usual strong presumption in the § 112(f) threshold test that method claim limitations without the (rarely used) term "step" are not step–plus–function limitations. *Id.* at 849. Yet, if *Williamson* applies to both apparatus and method claims, the Federal Circuit can no longer rely on a strong presumption against the application of § 112(f) to avoid drawing a distinction between steps and acts in method claims.

particular patent claim provide sufficient, but not excessive, reward to a particular software innovator?—as part of each and every patent examination at the Patent and Trademark Office and invalidity defense in the courts. What is needed is an administrable proxy for the ultimate economic question to replace the proxy of physical structure that is used in mechanical technologies but that is unavailable in software. I have suggested elsewhere that one possible proxy uses consumer preferences as an anchor.[133] Perhaps descriptions of functionality that map onto the reasonable consumer's desires should be labeled as goals because these preferences contribute to the definition of product markets, and perhaps descriptions of functionality that map onto technological ways of satisfying those desires should be seen as ways of achieving those goals and thus as logical structure. This proxy is far from straightforward, but it at least provides a stable point of reference to which the courts can peg a stand–alone definition of software's structure.

The next step to develop a stand–alone definition of structure for software under § 112(f) should involve an interdisciplinary conversation among patent lawyers, computer scientists, and economists.[134] The desired output of such a conversation would be a menu of stable levels of generality, grounded in computer science, at which all software can be described, and an analysis of the implications of selecting any one of these levels as a stand–alone definition of logical structure in software for the purpose of § 112(f). Having identified a menu of options and clarified the consequences of each option, the Federal Circuit would be well positioned to lead the revolution in software patent law called for by *Williamson*. It is possible that the conversation will not ultimately produce the distinctions necessary to provide an acceptable answer to the definitional "What is structure?" question. At the end of the day, the lesson learned might be that there are, in fact, no viable distinctions upon which to build a reformed law of software patents with the desired clarity. The purely functional nature of software may mean that § 112(f) just cannot regulate software patents in the

133. Collins, *supra* note 8, at 1421–23, 1466.

134. *See id.* at 1466–67 (arguing that such a conversation is needed). Professor Pamela Samuelson's efforts at revamping intellectual property protection for computer software provide an interesting model of the needed conversation, but not of the desired outcome, because the scope of the task at hand after *Williamson* is quite different from the scope of the task that Professor Samuelson undertook. *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308 (1994) (proposing a sui generis regime for intellectual property protection for software).

way that it regulates mechanical patents.[135] Nonetheless, the conversation would still be valuable. By taking a stand–alone definition of software structure off the table, it would bolster the case for a different revolution.

VII. CONCLUSION

Williamson v. Citrix Online altered the threshold test for determining whether a functional claim limitation that does not use the term "means" should be construed using the scope–narrowing rules of § 112(f). On its face, *Williamson* may only appear to implicate the quantitative "How much structure is enough?" question of the threshold test. That is, *Williamson* may only appear to require a bit more structure in the claim limitations than was previously required in order to evade § 112(f). However, with respect to software patents in particular, this Essay demonstrates that *Williamson* calls for a revolution in the law of functional claiming. *Williamson* will finally force the Federal Circuit to address the definitional "What is structure?" question as part of the § 112(f) threshold test in a more open and thorough manner.

The idea that *Williamson* will trigger a revolution in the legal definition of software's structure may be counterintuitive because, before *Williamson*, there was already a well–established answer to the definitional question in the context of identifying software's corresponding structure in the specification. Software's corresponding structure was an algorithm, or step–by–step procedure, for performing the function recited as a claim limitation. A revolution is needed, however, because this definition of an algorithm is relative and thus not well suited for identifying software's structure in the context of the threshold test that *Williamson* addresses. A search for an algorithm can identify structure in a functional description of a software program only in relation to the functional language employed in a claim limitation. The concept of an algorithm is not helpful when assessing whether the functional claim language itself recites structure: examining the functional language in a claim in relation to itself would be like staging a legal theater of the absurd. For this reason, *Williamson* requires a complete reconceptualization of what constitutes software's structure under § 112(f).

135. *Cf.* Kevin Emerson Collins, *Patent-Ineligibility as Counteraction*, 94 WASH U. L. REV. (forthcoming 2017) (arguing that purely functional technologies cause regulatory inefficacy in doctrines that, like § 112(f) and written description, rely on the recitation of structure to limit the overbreadth of functional claims).