The *Williamson* Revolution in Software's Structure

Kevin Emerson Collins·

Williamson v. Citrix Online altered the threshold test for determining whether a functional claim limitation that not use the term "means" should be construed using the scope-narrowing rules of section 112(f). More specifically, *Williamson* expands the reach of section 112(f) by replacing a formalistic, strong-presumption test with a substantive, sufficient-structure test. In most arts, *Williamson* will likely only have a small impact at the margin. *Williamson* will simply move the needle a bit on the *quantitative question* about structure in the threshold test: How much structure in a claim limitation is enough to avoid section 112(f)? A bit more structure will be needed after *Williamson* than was needed before. In contrast, in the software arts, *Williamson* will have a revolutionary impact. *Williamson* will finally force the Federal Circuit to address the *definitional question* about software's structure in the threshold test: What constitutes structure in a software limitation in the first place?

This essay makes two basic points. First, it demonstrates that *Williamson* will trigger a significant change in how section 112(f) applies to software patents. Before *Williamson*, the Federal Circuit was able to duck the difficult, definitional "What is structure?" question in almost all of its software cases, thanks to a combination of the formalistic, strong-presumption threshold test and a purely relative definition of structure in the rule for identifying corresponding structure in the specification. Now, after *Williamson*, the Federal Circuit must either acknowledge that the functional limitations of a software claim have no structure and that all functional software limitations are governed by section 112(f), or it must develop a new, stand-alone definition of software's structure. Second, this essay maps the difficult road that lies ahead if the Federal Circuit chooses to develop a stand-alone definition of software's structure.

· Professor of Law, Washington University.

The *Williamson* Revolution in Software's Structure

Kevin Emerson Collins·

In the first half of the twentieth century, the Supreme Court declared a number of patent claims using purely functional limitations—that is, limitations not reciting any of the structural properties of the technology that an inventor actually invented—to be facially invalid for overbreadth.[1] In response, Congress adopted what is now section 112(f) as part of the 1952

· Professor of Law, Washington University.

[1] *See infra* __.

Patent Act.[2] Section 112(f) expressly authorizes patent drafters to draft claims with purely functional limitations: a claim limitation "may be expressed as a means … for performing a specified function without the recital of structure [or] material … in support thereof."[3] However, as a price for using purely functional language, it mandates that the authorized limitations be construed using a statutorily specified, scope-narrowing rule of claim construction. Section 112(f) limitations "shall be construed to cover [only] the corresponding structure [or] material … described in the specification and equivalents thereof."[4]

In *Williamson v. Citrix Online*, the Federal Circuit altered the threshold test for determining whether a claim limitation is subject to section 112(f).[5] In the decade preceding *Williamson*, the Federal Circuit developed formalistic, strong presumptions that usually allowed a patent drafter to opt in or out of section 112(f). If a claim limitation contained the words "means for" performing a function, section 112(f) almost always governed, but, if it employed a synonym for "means," such as a "device for" or "mechanism for," performing the function, section 112(f) almost always never governed.[6] *Williamson* expanded the reach of section 112(f) by reducing the strength of the latter presumption that limitations without the term "means for" are not section 112(f) limitations. That presumption can now be overcome with the

[2] 35 U.S.C. § 112(f) (2012). This Essay refers to the statute as section 112(f), the title given to the statute by the AIA. Prior to the AIA, the statute was known for many years as section 112, paragraph 6.

[3] 35 U.S.C. § 112(f) (2012).

[4] *Id*.

[5] Williamson v. Citrix Online, LLC, 792 F.3d 1339 (Fed. Cir. 2015) (*en banc* in part).

[6] *Id.* at 1348–49.

substantive showing that a limitation recites "function without reciting sufficient structure for performing that function."[7]

In the mechanical arts, *Williamson* will likely only lead to incremental change. *Williamson* will move the needle on the *quantitative question* about structure in the section 112(f) threshold test. How much structure is enough to avoid the scope-narrowing rules of section 112(f)? After *Williamson*, a bit more structure is needed than was needed before. However, with respect to software patents in particular, *Williamson* mandates revolutionary change. To bring *Williamson* to bear on software patents, the Federal Circuit will be forced to openly ask and answer a more fundamental question that, to date, is has ducked. This more fundamental question is the *definitional question* about software's structure in the threshold test: What properties of a software program should count as structure to begin with when they are recited as claim limitations? The goals of this essay are to highlight the *sui generis* nature of the definitional question in the software arts; prove that, when posed as part of the threshold test, the definitional question raises a novel doctrinal issue that the Federal Circuit did not squarely confront in its pre-*Williamson* software cases;[8] and map out the difficult path forward that the Federal Circuit must follow in the post-*Williamson* era to provide a meaningful answer to the definitional question of the threshold test in software.

The definitional question is clearly important as a conceptual, logical matter. Knowing what constitutes structure in a claim limitation is a precondition for tallying the amount of structure recited and assessing whether that amount is sufficient to avoid section 112(f). What

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

[8] There are exceptions. *See infra* Section IV.B.

might be less clear, however, is why the definitional question doesn't have a simple answer in software. In mechanical technologies, it does have a simple answer. What constitutes structure is usually intuitive to a technological neophyte, let alone an experienced patent judge: structure includes the physical, spatial, and material properties of a technology. Here, the answer to the quantitative "How much structure is enough?" question may involve intricate legal reasoning, but the answer to the definitional "What is structure?" question does not. Software, however, is an intrinsically odd type of technology. The physical, structural qualities of a software invention are either the open or closed gates of a programmed computer or the arrangement of electrons on a storage medium. Yet, these structural qualities 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, meaning that software inventions can only reasonably be defined by reciting what the software does or how it performs in functional terms. There is no relevant, physical structure on which an economically rational patent regime can rely to curtail the scope of functional claims. To bring section 112(f) to bear on software, the Federal Circuit must develop a *sui generis* definition of logical structure in software—a definition which, importantly, invokes metaphorical structure in the sense that it is still a functional description of how software performs, not literal structure in the sense of the physical properties of a programmed computer or software recorded on a storage medium.

To be clear, the ideas that software lacks relevant physical structure and that section 112(f) requires a *sui generis* definition of software's structure had been recognized well before *Williamson*. Ever since its 1999 opinion in *WMS Gaming, Inc. v. International Game*

Technologies, the Federal Circuit has held that, in the context of searching for the corresponding structure in the specification, software's structure is something akin to logical, rather than physical, structure: it is an "algorithm" or a step-by-step procedure specifying how to perform the function recited as a claim limitation.[9] However, what has not been recognized is that the Federal Circuit's algorithm jurisprudence is of little to no use for answering the definitional question in the context the section 112(f) threshold test. Although the Federal Circuit's case law on algorithms might lead even an informed reader to presume that the definitional question about software's structure has already been fully asked and answered prior to *Williamson*, it has in fact been answered in a manner that is specific to identifying corresponding structure in the specification and that does not give the Federal Circuit any traction when identifying structure in claim limitations.

The underlying reason for this lack of traction is that the Federal Circuit has to date only offered a *relative* definition of an algorithm, and thus of software's structure, not a stand-alone definition. Again, an algorithm is step-by-step procedure specifying how to perform *a function recited as a claim limitation*. Whether a specification recites an algorithm can only be determined by comparing the series of functions disclosed in the specification to the baseline functionality recited as a claim limitation. A reader cannot examine a series of functions in isolation and say that it is or is not an algorithm. Corresponding structure is defined only in relation to a particular claim; the exact same disclosure may be structure in relation to a more general claim but not in relation to a more specific claim. During the search for corresponding structure, this relative definition of software's structure may be a bit awkward at times, but it at

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

least provides workable rule. However, during the search for structure in the claims, it is conceptually bankrupt. There is no self-evident baseline to which to compare the functional claim language. If an algorithm is what counts as structure in a software claim, for what task, precisely, must functional claim limitations be a step-by-step procedure?

Before *Williamson*, the strong presumptions allowed the Federal Circuit to sweep the difficult task of answering the definitional "What is structure?" question in the section 112(f) threshold test under the rug. They allowed the Federal Circuit to suppress the true extent of the *sui generis* nature of the law needed to bring section 112(f) to bear on software. They created the appearance that the application of section 112(f) to software claims was at least close to situation normal, when it is in fact anything but. After *Williamson*, however, the curtain will be pulled back, and the Federal Circuit will be forced to confront the definitional question about software's structure in the threshold test head-on. This confrontation will lead to a revolution in the doctrine of section 112(f) as applied to software, although the nature of that revolution remains to be seen.

A first, and doctrinally simple, revolution would be to fully embrace a relative definition of structure that results from defining software's structure as an algorithm. Perhaps patent drafters can disclose structure in the specification of a software patent in the form of an algorithm, but perhaps they cannot recite structure in the limitations of a software claim. Following this route, *Williamson* would subject every functional software limitation to the scope-narrowing rules of section 112(f), regardless of the level of specificity at which the limitation is drafted, and vastly expand the reach of section 112(f).

A second, and doctrinally more difficult, revolution would be to formulate the stand-alone definition of software's structure that is needed if there is ever to be any structure in functional software claims. Perhaps the Federal Circuit will finally identify a level of specificity or granularity that is required for a functional description of a software invention to count as logical structure, regardless of whether the functional description occurs in the specification or the claims, and, if it occurs in the specification, regardless of the generality of the claims in need of corresponding structure. The formulation of a stand-alone definition of structure in software for the purpose of the section 112(f) threshold test would be a difficult undertaking.[10] 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.[11] A stand-alone definition of structure is what Mark Lemley has alluded to with the distinctions in software patents between a "goal" and "steps for achieving a goal,"[12] a "goal" and a "way of implementing a goal,"[13] and the "problem" the patentee solved and "what the patentee … actually did."[14] It could be based on what I have elsewhere described as a line between functional limitations reciting end-user preferences that should be subject to section 112(f) and functional limitations reciting how those end-user preferences are achieved that should not.[15] Regardless of its verbal formulation, the initial step for developing the needed doctrine should

[10] 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 section 112(f) for step-plus-function method claims. *See infra* __.

[11] Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930); *see infra* __.

[12] Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 Wis. L. Rev. 905, 907.

[13] *Id.* at 947

[14] *Id.* at 963.

[15] Kevin Emerson Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 Wash U. L. Rev. 1399, 1421–23, 1466 (2013).

be an interdisciplinary conversation that brings the expertise of lawyers, computer scientists, and economists to bear on the problem in order to sketch the possible levels of abstraction at which a functional description of software could be deemed to be logical structure for the purpose of section 112(f).[16]

This essay proceeds as follows. The first three parts provide background. Part I briefly summarizes the history, policy, and doctrine section 112(f). Part II discusses the section 112(f) threshold test in greater detail and explains how *Williamson* altered that test. Turning to software patents before *Williamson*, Part III identifies why the application of section 112(f) to software requires *sui generis* rules that focus on logical structure, and it reviews the Federal Circuit's cases that require the disclosure of an algorithm in the specification as the corresponding structure for section 112(f) software limitations. The final two parts address life after *Williamson*. Part IV 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 section 112(f) as it applies to software. Either practically all software limitations become section 112(f) limitations, or the Federal Circuit must develop a stand-alone definition of structure. Part V briefly discusses the difficulty of the latter task of defining the structure of a software invention in a free-standing manner.

[16] *Id.* at 1466–67.

I. History, Policy, and Doctrine

During the first half of the twentieth century, the Supreme Court invalidated a number of claims that employed purely functional claim language to identify the outer bounds of an inventor's rights. In these cases, the Court repeatedly stated that functional claims were problematic as a policy matter because they granted patentees rights that were overbroad with respect to the patentees' actual contributions to progress.[17]

In 1928, *Holland Furniture Co. v. Perkins Glue Co.* involved a patent on a new starch-based glue with a low water content that had previously been attainable only in glue made from animal fat.[18] The inventor obtained a broad claim to any starch-based glue "having substantially the properties of animal glue."[19] The Court noted that the claim described the scope of the patentee's rights "not in terms of [the product's] own physical characteristics or chemical properties . . . but wholly in terms of the manner of use of the product,"[20] and it invalidated the claim because the functional language lead to overbreadth.[21] "As a description of the invention, [the functional language] is insufficient, and, if allowed, would extend the monopoly beyond the invention."[22]

---

[17] Due to the rhetoric in which the opinions are cast, these cases are sometimes mistakenly labeled as indefiniteness, rather than overbreadth, cases. Collins, *supra* note __, at 1429 n.122.

[18] 277 U.S. 245, 247 (1928).

[19] *Id.* at 250.

[20] *Id.* at 256

[21] *Id.* at 257 ("A claim so broad, if allowed, would operate to enable the inventor, who has discovered that a defined type of starch answers the required purpose, to exclude others from all other types of starch, and so foreclose efforts to discover other and better types. The patent monopoly would thus be extended beyond the discovery, and would discourage rather than promote invention.").

[22] *Id.* at 258.

A decade later, *General Electric Co. v. Wabash Appliance Corp.* again invalidated a functional claim for overbreadth.[23] A General Electric employee produced a new tungsten filament for a light bulb that had two commercially valuable properties—it suffered from neither sagging nor offsetting[24]—and General Electric obtained a functionally defined patent claim that encompassed tungsten filaments "made up mainly of a number of comparatively large grains of such size and contour as to prevent substantial sagging and offsetting during a normal or commercially useful life for such a lamp or device."[25] The Court held the claim "invalid on its face" because of its functional language granted the inventor overbroad rights.[26] "The claim . . . falls within the condemnation of the doctrine that a patentee may not broaden his product claims by describing the product in terms of function" and "vividly illustrates the vice of a description in terms of function."[27]

Finally, in 1946, the Supreme Court issued the most famous of its functional-language cases, *Halliburton Oil Well Cementing Co. v. Walker*.[28] Mr. Walker made an improved apparatus for measuring the depth of oil wells. Prior art devices measured the time at which sound waves sent into the well returned to the top of the well as echoes,[29] and Mr. Walker added a "mechanical acoustical resonator" to the echo-measurement device that made it easier to

[23] 304 U.S. 364 (1938).

[24] *Id.* at 366–367.

[25] *Id.* at 368.

[26] *Id.*

[27] *Id.* at 371. More specifically, the Court prohibited the use of "conveniently functional language at the exact point of novelty." *Id.*; *see also* Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 8 (1946).

[28] 329 U.S. 1 (1946).

[29] *Id.* at 4–5.

record the faint echoes from the small protrusions into the well created by the tubing collars or couplings at pipe junctions.[30] The functional claims before the Supreme Court recited "means associated with [the prior-art] device for tuning said receiving means to the frequency of echoes from the tubing collars of said tubing section to clearly distinguish the echoes of said couplings from each other."[31] Focusing on the "overhanging threat of the functional claim," the Court invalidated it:[32]

> [W]hat [Walker] claims here is that his patent bars anyone from using in an oil well any device heretofore or hereafter invented which combined with the [prior art] performs the function of clearly and distinctly catching and recording echoes from tubing joints with regularity. Just how many different devices there are of various kinds and characters which would serve to emphasize these echoes, we do not know... In this age of technological development there may be many other devices beyond our present information or indeed our imagination which will perform that function and yet fit these claims.[33]

These three Supreme Court cases barring purely functional claims because of overbreadth remain good law today 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. However, Congress softened the impact of the Court's functional-claiming cases by enacting what is now section 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,

[30] *Id.* at 6–7.

[31] *Id.* at 8–9.

[32] *Id.* at 12.

[33] *Id.* at 12.

material, or acts described in the specification and equivalents thereof.[34]

Section 112(f) allows patent drafters to use functional claim limitations, which can be helpful to ensure that patents play their expected role of providing an incentive to innovate because describing an invention without using functional limitations is sometimes difficult and invalidating any claim with purely functional limitation might leave inventors without effective patent protection. However, section 112(f) exacts a price from patent drafters who use functional limitations. It codifies an exception to the default rule of claim construction: 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.[35] A limitation under section 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 v. AWH* because, under section 112(f), claim scope is hewn much more closely to the particular embodiments disclosed in the specification.[36]

Today, the process of claim construction under section 112(f) proceeds in three steps. The first step is a threshold test: Is any given limitation the type of limitation that is subject to

[34] 35 U.S.C. § 112(f) (2012).

[35] *Id.* The Supreme Court's cases focused on the vice of functional claiming at the point of novelty. *See supra* note __. 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 aspects of prior art technologies. 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 __, at 958–59. This essay, however, brackets the issue of whether section 112(f) should have more bite at a claim's point of novelty and leaves it for another day.

[36] Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005) (en banc). Patent owners also often prefer having *Phillips* rather than section 112(f) govern the construction of a claim limitation because section 112(f) limitations often lead to claims being held invalid for indefiniteness. *See infra* __.

section 112(f), or should *Phillips* determine the meaning of the claim language?[37] The second step employs the default rules of claim construction to interpret the meaning of the functional language employed in the claim.[38] The third step then identifies the "corresponding structure" disclosed in the specification that is capable of performing the disclosed function.[39] In order to infringe, a technology must literally perform the function specified in the claim limitation, and its structure must be that of the corresponding structure disclosed in the specification or its equivalents.

If the specification does not disclose any corresponding structure, then the claim is invalid for indefiniteness under section 112(b) of the Patent Act.[40] The indefiniteness doctrine holds that claims that employ limitations whose meanings cannot be ascertained with reasonable certainty are invalid.[41] Indefiniteness is a common-sense rule: the scope of a claim to a "thingamajig" cannot be ascertained, so there is poor public notice, and there are many instances in which neither the validity nor infringement analyses can proceed. Section 112(f) states that the meaning of a functional claim limitation is the corresponding structure in the

[37] The threshold test is addressed at greater length below. *See infra* Section II.

[38] Generation II Orthotics, Inc. v. Med. Tech., Inc., 263 F.3d 1356, 1364–65 (Fed. Cir. 2001).

[39] 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, 952 (Fed.Cir.2007) ("The inquiry is whether one of skill in the art would understand the specification itself to disclose a structure, not simply whether that person would be capable of implementing a structure. Accordingly, a bare statement that known techniques or methods can be used does not disclose structure.") (citations omitted). The disclosed structure must also be clearly linked to the function recited in the claim limitation. B. Braun Medical, Inc. v. Abbott Laboratories, 124 F.3d 1419, 1424 (Fed. Cir. 1997).

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

[41] Nautilus, Inc. v. Biosig Instruments, Inc., 134 S. Ct. 2120 (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).

disclosure and its equivalents, so the limitation has no discernable meaning if there is no corresponding structure.[42]

By its literal terms, section 112(f) also governs method claims. That is, a claim limitation "may be expressed as a … step for performing a specified function without the recital of … acts in support thereof, and such [method] claim shall be construed to cover the corresponding … acts described in the specification and equivalents thereof."[43] While the Federal Circuit has occasionally discussed the notion of a step-plus-function method claim that is subject to section 112(f), it has not yet met a method claim that it believed should be subject to section 112(f).[44] For this reason, section 112(f) is often equated exclusively with "means-plus-function" limitations in product or system claims.

II. The Section 112(f) Threshold Test

The section 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 section 112(f). Section II.A discusses the intent and scope theories of the threshold test that have jockeyed for importance over the last two decades, and it reports how *Williamson v. Citrix Online* elevates scope theory over intent theory. Looking at the mechanical arts, Section II.B discusses several difficulties that *Williamson* will create for the

[42] *Donaldson*, 16 F.3d at 1195.

[43] 35 U.S.C. § 112(f) (2012).

[44] Seal-Flex, Inc. v. Athletic Track and Court Const., 172 F.3d 836, 848–51 (Fed. Cir. 1999) (Rader, J, concurring); O.I. Corp. v. Tekmar Co. Inc., 115 F.3d 1576, 1582–84 (Fed. Cir. 1997).

Federal Circuit when it seeks to answer the quantitative "How much structure is enough?" question wound up with the threshold test.

A. Intent Theory Versus Scope Theory

Two competing theories of the reach of section 112(f) have struggled for dominance over the last several decades. *Intent theory* suggests that section 112(f) is an option that Congress placed at the disposal of patent drafters and that section 112(f) should apply when, and only when, patent drafters use the word "means" as a signal that they intend to invoke it. In contrast, *scope theory* suggests that the section 112(f) threshold test should look beyond the patent drafters' semantic word choice and focus on whether the claim limitation recites enough structure so that the Supreme Court's policy concerns about overbroad claim scope raised in *Holland Furniture*, *General Electric*, and *Halliburton Oil Well Cementing* are mitigated or eliminated. Neither intent theory nor scope theory has ever achieved complete dominance in the Federal Circuit, but the weight given to each has varied over time. In the decade or so prior to *Williamson*, intent theory had the upper hand. With *Williamson*, however, the Federal Circuit established the dominance of scope theory.

The choice between the intent and scope theories reflects many of the policy tradeoffs wound up with the classic choice between rules and standards.[45] Like other rules, a threshold test based on intent theory produces greater certainty. It "provides clarity and predictability for the public and the patentee alike. It helps the public to determine when claim elements are expressly limited to structures disclosed in the specification (or their equivalents) and provides

[45] Frederick Schauer, Playing by the Rules: A Philosophical Examination of Rule-Based Decision-Making in Law and in Life (1991).

the patentee with the tools for reliability invoking or avoiding means-plus-function claiming."[46] However, also like other rules, it is under-inclusive with respect to the justifiable outcome because it frequently sanctions the very functional claims that the Supreme Court viewed as economically overbroad.[47] Inversely, scope theory yields a more difficult-to-administer standard that imposes a restriction on patentability that is more closely tailored to eliminating the overbreadth of functional claims.

In 1996, both intent and scope theory were given first billing in different opinions by different panels. In *Greenberg v. Ethicon Endo-Surgery*, the Federal Circuit articulated a strong version of intent theory in the course of deciding that the term "detent mechanism" in a medical-instrument claim was not subject to section 112(f).[48] Drawing on the literal text of section 112(f), the court argued that section 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.[49]

Thus, intent theory gives rise to twin presumptions of threshold test: "the term 'means' (particularly as used in the phrase 'means for') generally invokes section [112(f)] and … the use

[46] Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1297–98 (Fed. Cir. 2014).

[47] *See supra* __. The explanation of why functional claims are overbroad is more complicated that is commonly acknowledged. Collins, *supra* __, at 1411–24.

[48] Greenberg v. Ethicon Endo-Surgery, Inc., 91 F.3d 1580 (Fed. Cir. 1996).

[49] *Id.* at 1584; *see also* York Prods., Inc. v. Central Tractor Farm & Family Ctr., 99 F.3d 1568, 1574 (Fed. Cir. 1996) ("In determining whether to apply the statutory procedures of section 112, ¶ 6, the use of the word 'means' triggers a presumption that the inventor used this term advisedly to invoke the statutory mandates for means-plus-function clauses."); *cf.* Donald Chisum, Chisum on Patents § 18.03[5][e][ii][E] ("To apply Section 112/6's claim construction to a claim phrase that does not use the key word means is, potentially, a significant step beyond the strict, literal language of the statute.").

of a different formulation generally does not."[50] In contrast, in *Cole v. Kimberly-Clark Corp.*, the majority opinion articulated a version of scope theory to explain the policy behind the section 112(f) threshold test. The court downplayed intent theory. "Merely because a named element of a patent claim is followed by the word 'means,' … does not automatically make that element a 'means-plus-function' element," and "merely because an element does not include the word 'means' does not automatically prevent that element from being construed as a means-plus-function element."[51] Instead, the court stated that claim limitations should be subjected to the scope-narrowing rules of section 112(f) threshold test if the term at issue is excessively functional and thus overbroad. That is, a section 112(f) claim limitation "must not recite a definite structure which performs the described function."[52]

By the 2000s, intent theory had clearly gained the upper hand. In *Lighting World, Inc. v. Birchwood Lighting, Inc.*, "the presumption flowing from the absence of the term 'means'" had become "a strong one that is not readily overcome."[53] In fact, the articulation of intent theory in *Lighting World* suggested that scope theory was entirely defunct because a claim limitation not using the word "means" might not be a section 112(f) limitation even if it encompassed all

[50] *Greenberg*, 91 F.3d at 1584; *see also* 1583 (noting that, if claims use the term "means for," the Federal Circuit noted that the drafters "intend to invoke" section 112(f) "and that there is therefore seldom any confusion about whether [112(f)] applies to a particular element.").

[51] Cole v. Kimberly-Clark Corp., 102 F.3d 524, 531 (Fed. Cir. 1996).

[52] *Id.* Judge Rader, who had previously authored an opinion emphasizing intent theory in *York Prods. v. Central Tractor Farm & Family Ctr.*, *see supra* note __, dissented in *Cole*, arguing that intent theory should animate the section 112(f) threshold test. [] Scope theory was arguably quite novel at this time. In its 1998 opinion in *Mas-Hamilton Group v. LaGard, Inc.*, 156 F.3d 1206, 1213–14 (Fed. Cir. 1998), the Federal Circuit held for the first time that a phrase not using "means" was governed by section 112(f). [Chisum]; *cf.* Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1359–60 (Fed. Cir. 2004) (labeled *Mas-Hamilton* as an "exceptional case").

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

structures that are capable of performing the claimed function.[54] The strength of intent theory was subsequently reinforced with the assertion that "[w]hen the claim drafter has not signaled his intent to invoke § 112 ¶ 6 by using the term 'means,' we are unwilling to apply that provision without a showing that the limitation essentially is devoid of anything that can be construed as structure."[55] Even under the strong-presumption formulation of the threshold test, the Federal Circuit's opinions are laced with scope-theory sound bites.[56] However, very few claim limitations that did not employ the word "means" were held to actually be governed by section 112(f).[57]

In 2014, the conflict between intent and scope theories again resurfaced in the disagreement between the Federal Circuit's majority and dissenting opinions in *Apple v. Motorola*.[58] The majority adopted a strong version of intent theory to reverse a district court holding that a claim limitation without the term "means" was not a means-plus-function claim. "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."[59] In contrast,

[54] *Id.* at 1361–62 (noting that the term "connector" is "certainly broad" and that it "may in the end include any structure that performs the role of connecting").

[55] Flo Healthcare Solutions, LLC v. Kappos, 697 F.3d 1367, 1374 (Fed. Cir. 2012)

[56] *Lighting World*, 382 F.3d at 1360 (noting that section 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'").

[57] *But see* Welker Bearing Co. v. PHD, Inc., 550 F.3d 1090, 1096–97 (Fed. Cir. 2008) ("mechanism"); MIT v. Abacus Software, 462 F.3d 1344, 1355 (Fed. Cir. 2006) ("mechanism"); Mas-Hamilton Group v. LaGard, Inc., 156 F.3d 1206, 1213–14 (Fed. Cir. 1998) ("element" and "member").

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

[59] *Id.* at 1297. The majority, however, seems to have forgotten about the Supreme Court's functional-language cases from the first half of the twentieth century in its explanation of the consequences that a patentee may suffer by using functional language in claim limitations that are not subject to section 112(f):

the 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."[60]

Only a year after *Apple*, 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.[61] 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 section 112(f) when a claim limitation did not use the word "means":

> has the inappropriate practical effect of placing 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.[62]

The Federal Circuit proceeded to flatly state that the "heightened burden" announced in *Lighting World* "is unjustified" and that it "should abandon characterizing as 'strong' the

---

> We recognize that the choice to draft a claim in 'broad [functional] terms' rather than in means-plus-function format may render the claim more vulnerable to an invalidity attack. Whether to draft a claim in broad [functional] terms is the claim drafter's choice, and any resulting risk that emanates from that choice is not a basis for the court to rewrite a claim in means-plus-function format.

Id. at 1298–99; *see also* Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1361–62 (Fed. Cir. 2004). This passage seems to recognize that claims drafted with "broad functional terms" may be vulnerable to attack under contemporary validity doctrines like enablement and written description, but it seems to ignore the Supreme Court's cases indicating that claims drafted with "broad functional terms" are facially invalid. *See supra* __.

[60] *Id.* at 1337 (Prost, J., concurring-in-part and dissenting-in-part).

[61] Williamson v. Citrix Online, LLC, 792 F.3d 1339 (Fed. Cir. 2015) (*en banc* in part). Only the section addressing the proper standard for the 112(f) threshold was authored *en banc*.

[62] *Id.* at 1349.

presumption that a limitation lacking the word 'means' is not subject to § 112 ¶ 6."[63] Instead, "[t]he standard is whether the words of the claim are understood by persons of ordinary skill in the art to have sufficiently definite meaning as the name for structure."[64]

However, even *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 section 112(f) threshold test. The presumption that section 112(f) does not apply to limitations without the term "means" persists, and *Williamson* only made it easier to rebut,[65] and the converse, pre-*Williams* strong presumption that the use of the word "means" does trigger section 112(f) remains in place.[66]

B. Uncertainties in the Quantitative Question after *Williamson*

One important question after the Federal Circuit's tilt toward scope theory in *Williamson* will be the quantitative question of the threshold test: How much structure is enough to avoid section 112(f)?[67] Four unresolved issues that all contribute to uncertainty in the answer to this question after *Williamson* are noted below.

First, as a *Williamson* concurrence notes, the Federal Circuit does not clarify what force the not-strong presumption against section 112(f) that follows from the absence of "means"

[63] *Id.* Judge Newman dissented on this point. Channeling intent theory, she argued that "it 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." *Id.* at 1358 (Newman, J., dissenting).

[64] *Id.* at 1349.

[65] *Id.* In the concurring section of his concurring-in-part and dissenting-in-part opinion, Judge Reyna argues that even this weaker presumption should be abandoned in favor of a more flexible framework. *Id.* at 1356–57.

[66] *Id.* at 1349.

[67] Michael Risch, *The Past and Future of Functional Claiming …*, Written Description (June 16, 2015), *available at* http://writtendescription.blogspot.com/2015/06/the-past-and-future-of-functional.html.

has after *Williamson*.[68] Earlier Federal Circuit opinions suggest that the burden of persuasion is a preponderance of the evidence,[69] but whether *Williamson* returns to this burden is unclear.

Second, future cases will have to flush out the substantive standard. Under what conditions does the claim limitation have "sufficiently definite meaning as the name for structure"? Any old bit of structure will likely not be enough. In fact, *Williamson* picks up on earlier case law suggesting that this sufficient-structure standard may not be unitary but may instead have two different prongs: "When a claim term lacks the word 'means,' the presumption can be overcome and § 112, para. 6 will apply if the challenger demonstrates that the claim term fails to 'recite sufficiently definite structure' *or else* recites 'function without reciting sufficient structure for performing that function.'"[70] Are these two requirements merely distinct verbal formulations of the same standard, or are they substantively different threshold that must both be satisfied? One post-*Williamson* district court opinion interpreted them to establish distinct thresholds for sufficient structure and held that the claims at issue satisfied the first but not the second.[71]

Third, and potentially most problematically, the Federal Circuit has never clarified the methodology that should be used to determine whether a claim recites sufficient structure. Some cases seem to state that the default claim-construction methodology specified in *Phillips*

[68] *Williamson*, 792 F.3d at 1357 n.2 (Reyna, J., concurring-in-part and dissenting-in-part).

[69] Apex Inc. v. Raritan Computer, Inc. 325 F.3d 1364, 1372 (Fed. Cir. 2003).

[70] *Williamson*, 792 F.3d at 1349 (quoting Watts v. XL Sys., Inc., 232 F.3d 877, 880 (Fed.Cir.2000)) (emphasis added); *see also* CCS Fitness, 288 F.3d 1359, 1369 (Fed. Cir. 2002).

[71] The district court concluded that the claim limitation "processor for associating the content data with dispatch record data" recited both "sufficiently definite structure" and "function without sufficient structure for performing that function."). GoDaddy.com, LLC v. RPost Comm'ns Ltd., 2016 WL 212676, at *53–*57 (D. Ariz. Jan. 19, 2016).

v. AWH[72] should control the search for sufficient structure.[73] The use of *Phillips* would make 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 search for sufficient structure in the threshold test not in the exact terms outlined in *Phillips* but rather in different terms that roughly mirror *Phillips*. For example, Federal Circuit cases don't 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.[74] Is connotation the same as meaning, or can a limitation connote structure (in the sense of vaguely invoking it) without that structure actually being a limit on literal claim scope under *Phillips*? Similarly, Federal Circuit cases are not precise about the hierarchy in authority of the possible interpretive sources that can be used to determine whether a claim limitation connotes sufficient structure. It is well established that the Federal Circuit performs the section 112(f) threshold test "on an element-by-element basis, based upon the patent and its prosecution history."[75] In some cases, as under *Phillips*, the structure comes from extrinsic evidence such as trade usage by the person having

[72] Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

[73] Personalized Media Communications, LLC v. U.S. Int'l Trade Comm'n, 161 F.3d 696, 704 (Fed. Cir. 1998) (stating that, in the section 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").

[74] *See, e.g.*, Apex v. Raritan Computer, 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.")

[75] Cole v. Kimberly-Clark Corp., 102 F.3d 524, 531 (Fed. Cir. 1996).

ordinary skill.[76] In other cases, again as under *Phillips*, the structure comes from intrinsic sources. Language accompanying the functional term in the limitation itself may provide the structure,[77] or the claim language may have structure when it is read in the context of the specification.[78] However, the key to *Phillips* and the Federal Circuit's claim construction doctrine as a whole is the guidance that it gives about which of these interpretive sources should be given greater weight.[79] To date, this discussion about the balance of the different interpretive sources has been absent in the search for sufficient structure during the threshold test. 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*.[80]

Fourth, it is unclear what constitutes a question of law and a question of fact in the threshold test. In *Teva Pharmaceuticals v. Sandoz Inc.*, the Supreme Court held that appellate

[76] Greenberg v. Ethicon Endo-Surgery, Inc., 91 F.3d 1580, 1583 (Fed. Cir. 1996) (holding that although its plain dictionary meaning was expressed in functional language, "as the name for structure, has a reasonably well understood meaning in the art")

[77] Watts v. XL Sys., Inc., 232 F.3d 877, 880–81 (Fed. Cir. 2000) (holding that a "sealingly connected" limitation was not governed by section 112(f) because it also recited "a second end formed with tapered external threads"); Al-Site Corp. v. VSI Int'l, Inc. 174 F.3d 1308, 1318 (Fed. Cir. 1999) (holding that a claim reciting a "member" for "mounting a pair of eyeglasses" was not a section 112(f) claim because, "the claims themselves contain sufficient structural limitations for performing" that function, such as the requirement that the eyeglass hanger member be "made from flat sheet material" with an "opening means formed ... below [its] upper edge").

[78] Inventio AG v. Thyssen Krupp Elevator Americas Corp., 649 F.3d 1350, 1357–59 (Fed. Cir. 2011) (giving significant weight to the written description in determining that a "modernizing device" limitation recited sufficient structure so that it was not governed by section 112(f)).

[79] There have been complaints about the clarity of the guidance given in *Phillips*. *See, e.g.*, Retractable Techs., Inc. v. Becton, Dickinson & Co., 659 F.3d 1369 (Fed. Cir. 2011) (Moore, J., dissenting from the denial of the petition for rehearing en banc) ("*Retractable* simply cannot be reconciled with our en banc decision in *Phillips*.").

[80] *See, e.g., infra* __ (discussing *Apple v. Motorola*). This willingness to rely on the specification to identify sufficient structure may have resulted in part from the strong presumption against the invocation of section 112(f) when the claim limitation does not employ the term "means," so it is unclear whether it will continue after *Williamson*.

courts cannot review district courts' claim construction rulings on an entirely *de novo* basis but must instead review any underlying issues of fact decided by the district court more deferentially under a clear error standard of review.[81] The section 112(f) threshold test is an issue of claim construction and thus ultimately a question of law,[82] but there are likely to be factual findings, such as the trade usage of a term to a person of ordinary skill, that are wound up in the decision. In *Lighting Ballast Control v. Phillips Electronics*, the Federal Circuit deferred to a district court's determination that the claim term "voltage source means" was not a section 112(f) limitation based on trade usage: "it is … understood by persons of skill in the art to connote a class of structures."[83] While trade usage is a question of fact, the quantitative question—that is, whether there is sufficient structure in that usage to overcome the presumption in favor of section 112(f) when a claim employs the term "means"—would seem to be a question of law on which no deference is due.

III. Section 112(f) and Software

The application of section 112(f) to software requires a *sui generis* modification of conventional section 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 section 112(f) to software before *Williamson* by

[81] Teva Pharms. v. Sandoz Inc., 135 S.Ct. 831 (2015).

[82] Markman v. Westview Instruments, Inc., 517 U.S. 370 (1996).

[83] Lighting Ballast Control LLC v. Philips Electronics North America Corp., 790 F.3d 1329, 1338–39 (Fed. Cir. 2015).

concluding that an "algorithm" is the corresponding structure in the specification for functional software limitations.

A. The Software/Section 112(f) Mismatch[84]

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, there is no such distinction. Software inventions are, at least as a practical matter and for the purpose of patent law, a purely functional technology. To bring section 112(f) to bear on software claims, courts must identify software's logical structure, i.e., the type of functional description of a software invention that should count as metaphorical structure.

In the historical core of section 112(f) claiming in the mechanical arts, what constitutes a structural property of an invention and how an invention's structural properties differ 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."[85] For example, a "skid plate,"[86] "button and hole arrangement,"[87] and "a rotatable disc with an opening sits above the container cap"[88] are all structural descriptions of objects.[89] Few—if any—cases in the

[84] Much of the argument in this section draws from Collins, *supra* note __, at 1440–43.

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

[86] Chiuminatta Concrete Concepts, Inc. v. Cardinal Industries, 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" and further specified that "[t]he flat piece … is generally parallel to the base plate").

[87] Al-Site Corp. v. VSI International, Inc., 174 F.3d 1308, 1315–16 (Fed. Cir. 1999).

[88] Sage Products, Inc. v. Devon Industries, Inc., 126 F.3d 1420, 1428 (Fed. Cir. 1997). Here, "rotatable" adds a dollop of function to the description.

mechanical arts bother even to expressly pose the definitional "What is structure?" question because the answer is so self-evident. To be clear, there are plenty of cases in the mechanical arts in which it is unclear whether a term recites any structure as a limitation on claim scope or whether the quantum of structure recited is sufficient.[90] However, most of this uncertainty inheres in either the meaning of language or the legal doctrine that controls the answer to quantitative question, not the nature of a structural property of a thing. 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.[91]

In the software arts, however, what should count as structure for the purposes of section 112(f) is not intuitive. Three links in a chain of arguments are important to emphasize here. First, software programs do have physical, structural properties, just like other technologies do. However, second, those physical, structural properties are of no use in section 112(f). They are irrelevant to the definition of a software invention because software inventions are functional entities on all relevant levels of definition. Thus, third, the definitional question—"What is structure in the software arts?"—must focus on logical structure. It must identify a functional definition of a software invention that is sufficiently granular to serve as

[89] Structure is also an uncontroversial concept in the chemical and biochemical arts because structure is equated with molecular structure. Ariad Pharm., Inc. v. Eli Lilly & Co., 598 F.3d 1336 (Fed. Cir. 2010) (en banc). 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 __, at 1430–33.

[90] *See supra* __ (discussing the quantitative question of the section 112(f) threshold test).

[91] The distinction between structural and functional properties is a 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 Phil. & Tech. 18, 18 (1998) (discussing "two different modes of description, viz., a *structural* and a *functional* mode of description" for technological objects).

metaphorical structure, at least if the scope-narrowing strictures of section 112(f) are to be brought to bear on software at all.

First, despite the commonplace description of software as "intangible,"[92] every real-world embodiment of a software program is a material, worldly entity that has structural properties, just like every embodiment of a mechanical technology has structural properties.[93] 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 closed.[94] "The instructions of the software program … electrically change the general purpose computer by creating electrical paths within the device."[95]

Second, the physical, structural, and material properties of a software embodiment are usually irrelevant to the task of identifying, delineating, or defining a protectable software invention, whether under section 112(f) or any other patent doctrine that invokes structure. In

[92] *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 and mental processes); *In re* Grams, 888 F.2d 835 (Fed. Cir. 1989) (same); *In re* Bernhart, 417 F.2d 1395 (C.C.P.A. 1969) (same); 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.").

[93] 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)); Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform*, 7 UCLA J.L. & TECH. 1, 8–12 (2003).

[94] WMS gaming, at 1348 & n.3; Plotkin, *supra* note __, at 38–39. In fact, patent law has unflinchingly recognized the tangibility of software embodiments in some of its doctrines even as it has denied the tangibility of software in others. For example, a computer programmed with a new software program has long been treated as a new machine under the novelty doctrine that is structurally distinct from prior art machines. *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc); *In re* Freeman, 573 F.2d 1237, 1247 (C.C.P.A. 1978); *In re* Bernhart, 417 Fe.2d 1395, 1399–1400 (C.C.P.A. 1969); *In re* Prater, 415 F.2d 1393, 1403 n. 29 (C.C.P.A. 1969).

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

most arts, it is natural to say that an invention is its structure, not its function.[96] In the software arts, however, the opposite characterization seems more natural: an invention is its function, not its structure.[97] A usable embodiment of a software invention may not be literally immaterial, but the description of a software invention must be nothing but function as a practical matter. The irrelevance of the physical, structural properties of a software embodiment to the definition of a software program has been engineered into the very nature of software itself. The core value of software lies in the fact that the design of software that possesses any given set of logical, functional properties need not involve any consideration of either the physical properties of the hardware or the distribution of electrical charges on a disk. In fact, "[F]or all practical purposes the programmer and others who think about and describe the program have no practical choice but to conceive of and describe it in terms of its logical structure [or function]. . . . It is far from clear that it would even be possible for the human mind to appreciate the physical structure of all but the simplest programs or to explain them in terms of their physical structures."[98] A software program can be implemented in entirely different code in the same programming language or in entirely different languages, and the sets of code

[96] *See supra* __.

[97] Note, *Everlasting Software*, 125 Harv. L. Rev. 1454, 1456 (2012).

[98] Plotkin, *supra* note __, at 46 & n.126; *see also id.* at 26 ("The process of computer programming enables a programmer to create a machine that has a particular novel physical structure for performing a particular function without requiring the programmer to design the novel features of the machine in physical terms."); *id.* at 36 ("[O]ne of computer science's express goals is to ensure that programmers can do their work in complete ignorance of the physical structure of . . . hardware . . . ."); *id.* at 44–45 ("[A] programmer who modifies the physical structure of a computer by providing source code to the computer need not even know that the computer's memory is being physically modified at all, much less understand or appreciate the nature of those physical modifications.") (citations omitted).

that all embody a software invention have few, if any, structural properties in common.[99] Thanks to interpreters and compilers, any given program can be implemented on a wide array of different computers, each possessing a different internal architecture and requiring the software to take on different physical, structural properties.[100] 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. For patent drafters writing software claims, 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.[101]

Third, 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 section 112(f) presents two doctrinally simple solutions, but both are so extreme that neither is palatable.[102] 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

[99] 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).

[100] *See* W. DANIEL HILLIS, THE PATTERN ON THE STONE, at 56–58 (1998) (discussing interpreters and compilers); *cf.* Microsoft Corp. v. AT&T Corp., 550 U.S. 437, 450 (2007) ("Software . . . is a stand-alone product developed and marketed 'for use on many different types of computer hardware . . . .'").

[101] Technically, the nested dolls do end at some point. *See infra* __ (discussing *In re Katz*).

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

trivial undertaking. On the other hand, we could exempt software from the strictures of section 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.[103] Given the problematic nature of both simple, but extreme, doctrinal solutions, an exploration of the *sui generis* ways in which section 112(f) could be modified in the purely functional arts like software is a worthwhile undertaking to explore.[104] Such an exploration of when function should count as metaphorical structure offers the greatest promise for way to use section 112(f) to navigate a course between the Scylla and Charybdis of trivially narrow and vastly overbroad claims.

B. Algorithms as Corresponding Structure

The need to create a *sui generis* doctrine for section 112(f) to deal with software claims has not been lost on the Federal Circuit. The Federal Circuit has developed an extensive body of cases addressing what constitutes the corresponding structure of a section 112(f) software limitation in a patent disclosure.[105]

Initially, it is important to emphasize that the Federal Circuit has rejected the idea that corresponding structure for the purposes of section 112(f) is physical structure. The Federal Circuit has repeatedly held that the disclosure of a general purpose computer or generic

[103] *See supra* __.

[104] The jury is still out, however, on whether it will succeed. *See infra* __.

[105] Some pre-*Williamosn* Federal Circuit cases also address what constitutes structure in a claim limitation as part of the section 112(f) threshold test, but the reasoning in these cases is either misguided (because it relies on physical structure) or radically underdeveloped. *See infra* Section IV.B.

microprocessor in the specification that can execute software is not the corresponding structure for software limitations.[106] Thus, as the Federal Circuit recently stated, "'[s]tructure' to a person of ordinary skill in the art of computer-implemented inventions may differ from more traditional, mechanical structure …. Indeed, the typical physical structure that implements software, a computer, cannot be relied upon to provide sufficiently definite structure for a software claim …."[107] Nor has the Federal Circuit even entertained the notion that the disclosure of a distribution of electronics on a storage medium or an arrangement of open and closed gates in a computer, which are the structural properties of a software invention that differentiate the invention from all other software programs, should be the corresponding structure for a software limitation.

Rather, the Federal Circuit has held that the corresponding structure in the specification for the purposes of section 112(f) is metaphorical structure—a technically functional description of software that can play the scope-limiting role that physical structure plays in the mechanical arts. This 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 section 112(f) and "structure" for the purpose of the section 101 prohibition of *In re Nuijten* on

[106] *See, e.g.*, EON Corp. v. AT&T Mobility LLC, 785 F.3d 616, 621 (Fed. Cir. 2015); Aristocrat Techs. Austl. Pty Ltd. V. Int'l Game Tech., 521 F.3d 1328, 1333 (Fed. Cir. 2008); WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1349 (Fed. Cir. 1999). *But see infra* __ (discussing the *Katz* exception). Technically, a microprocessor is a structural feature of the claimed technology. However, it should not be enough structure to serve as corresponding structure. The analogy in the mechanical arts to the disclosure of a microprocessor as corresponding structure might be the disclosure of a material like metal as the corresponding structure. Being made of metal is a structural property, but it is not enough information about structure to say that the specification has disclosed corresponding structure.

[107] Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1298 (2014). In this same passage, the opinion also states that "looking for traditional 'physical structure' in a computer software claim is fruitless because software does not contain physical structures." *Id.* This statement is technically inaccurate. *See supra* __.

claims to propagating signals or software *per se*.[108] Thus, there is ambiguity in the term "structure" in the law of software patents. The latter definition that is relevant to section 101 requires physical structure, and recitation of a microprocessor or a fixed storage medium is enough to avoid a patent-ineligible claim to software *per se*. However, the former definition that is relevant to section 112(f) does not refer to physical structure.

The Federal Circuit's answer to the definitional "What is structure?" question in the context of searching for corresponding structure in the specification is that structure is an "algorithm." What the Federal Circuit means by an algorithm is "in essence a series of instructions for the computer to follow,"[109] "a step-by-step procedure for accomplishing a given result,"[110] or a "sequence of computational steps to follow."[111] Simply put, "the essence of algorithms" is "what to do to perform a task" that is recited as a claim limitation.[112]

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*.[113] The claim limitations at issue used the word "means," and the parties did not dispute that the claim terms at issue were subject to section 112(f) under the threshold test. The district court construed the limitation broadly to encompass any table, formula, or algorithm for performing

[108] *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.* n.6 (noting that the signal on a "storage medium" would be patent eligible).

[109] 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)).

[110] *Id.* The PTO defines an algorithm as a "finite sequence of steps for solving a logical or mathematical problem or performing a task." [PTO Guidelines, quoting Microsoft Computer Dictionary, Microsoft Press, 5th edition 2002.]

[111] Ibormeith IP, LLC 14 v. Mercedes-Benz USA, LLC, 732 F.3d 1376, 1379 (Fed. Cir. 2013) (citations omitted).

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

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

the claimed function, but the Federal Circuit insisted that the scope of the claim was narrower because it was limited to "the special purpose computer programmed to perform the disclosed algorithm" in the specification, as well as equivalents.[114] In subsequent cases, the Federal Circuit has clarified that there is no single format in which an algorithm must be communicated. Mathematical formulae, prose, and flow charts can all express algorithms.[115]

Starting in the late 2000s, the Federal Circuit began to invalidate a large number of software claims with section 112(f) limitations for indefiniteness because the specification did not disclose an algorithm as corresponding structure.[116] These cases offer an interesting window into what constitutes an algorithm. 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.*, a claim recited the section 112(f) limitation "database editing means … for generating a hierarchically arranged set of indices for referencing data in said information database."[117] The Federal Circuit affirmed the district court's holding of indefiniteness because the specification provided "nothing more than a restatement of the

[114] *Id.* at 1349.

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

[116] Eon Corp. IP Holdings LLC v. AT&T Mobility LLC, 785 F.3d 616 (Fed. Cir. 2015); Triton Tech, LLC v. Nintendo, Inc., 753 F.3d 1375 (Fed. Cir. 2014); Robert Bosch, LLC v. Snap-On Inc., 769 F.3d 1094 (Fed. Cir. 2014); Function Media, L.L.C. v. Google Inc., No. 2012-1020, 2013 WL 516366 (Fed. Cir. Feb. 13, 2013); ePlus, Inc. v. Lawson Software, Inc., 700 F.3d 509 (Fed. Cir. 2012); Noah Sys., Inc., v. Intuit Inc., 675 F.3d 1302 (Fed. Cir. 2012); HTC Corp. v. IPCom GmbH & Co., 667 F.3d 1270 (Fed. Cir. 2012); Dealertrack, Inc. v. Huber, 674 F.3d 1315 (Fed. Cir. 2012); *In re* Aoyama, 656 F.3d 1293 (Fed. Cir. 2011); Rembrandt Data Techs. v. AOL, 641 F.3d 1331 (Fed. Cir. 2011); *In re* Katz Interactive Call Processing Patent Litig., 639 F.3d 1303 (Fed. Cir. 2011); Brown v. Baylor Healthcare Sys., No. 2009-1530, 2010 WL 1838921 (Fed. Cir. May 7, 2010); Net MoneyIN, Inc. v. VeriSign, Inc., 545 F.3d 1359, 1366–67 (Fed. Cir. 2009); Encyclopaedia Britannica, Inc. v. Alpine Elecs., Inc., No. 2009-1087, 2009 WL 4458527 (Fed. Cir. Dec. 4, 2009); Blackboard, Inc. v. Desire2Learn Inc., 574 F.3d 1371, 1383–85 (Fed. Cir. 2008); Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323 (Fed. Cir. 2008); Aristocrat Techs. Austl. Pty. Ltd. v. Int'l Game Tech., 521 F.3d 1328 (Fed. Cir. 2008).

[117] Finisar Corp. v. Direct TV Group, Inc., 523 F.3d 1323, 1340 (Fed. Cir. 2008).

function, as recited in the claim."[118] However, if the specification provides a more granular functional description—that is, if it discloses a series of functional steps that, when strung together, achieve the claimed function—then the specification does disclose an algorithm. For example, in *Typhoon Touch Techs. v. Dell*, a claim recited the section 112(f) limitation "means for cross-referencing said responses with one of said libraries of said possible responses."[119] The district court held that there was no corresponding algorithm disclosed in the specification and that the claims containing the limitation were invalid for indefiniteness. The Federal Circuit reversed, holding that 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."[120] 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 more granular tasks formed a process for accomplishing the claimed task. They were thus an algorithm, and they qualified as corresponding structure.

The Federal Circuit has identified one category of functional limitations in software claims that do not require the disclosure of any corresponding structure. In *In re Katz Interactive Call Processing Patent Litigation*, the Federal Circuit held that the claimed functions of "processing," "receiving," and "storing" did not require the disclosure of an algorithm even

[118] *Id.*

[119] Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376, 1383 (Fed. Cir. 2011).

[120] *Id.* at 1385.

though the limitations were subject to section 112(f).[121] It reasoned that the disclosure of an algorithm was needed to have corresponding structure when the claims described "specific functions that would need to be implemented by programming a general purpose computer to convert it into a special purpose computer capable of performing those specified functions."[122] However, it concluded that the claimed functions at issue in *Katz* could rely on the disclosure of a general purpose computer as the corresponding structure because they "can be achieved by any general purpose computer without special programming. As such it was not necessary to disclose more structure than the general purpose processor that performs those functions."[123]

To reiterate, apart from the "narrow" *Katz* exception,[124] section 112(f) software limitations require the disclosure of corresponding structure in the form of an algorithm or a step-by-step procedure for achieving the claimed functionality. However, an algorithm does not disclose physical structure. The steps that make up an algorithm are still functional descriptions of tasks to be performed. For example, each of the steps in the *Typhoon Touch Techs.* 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."[125] The difference is only the level of granularity of the functional description. The Federal Circuit does occasionally imply that an

[121] *In re* Katz Interactive Call Processing Patent Litigation, 639 F.3d 1303, 1316 (Fed. Cir. 2011).

[122] *Id.*

[123] *Id.*

[124] Subsequent Federal Circuit cases that have considered the *Katz* exception have called it "narrow," have not found it to apply, and have required the disclosure of an algorithm in the specification to avoid indefiniteness. EON Corp. v. AT&T Mobility LLC, 785 F.3d 616, 621–23 (Fed. Cir. 2015); Ergo Licensing, LLC v. CareFusion 303, Inc., 673 F.3d 1361, 1364 (Fed. Cir. 2012).

[125] *See supra* __.

algorithm is something different from a functional description of software by juxtaposing an algorithm to functional language. For example, it explained its invalidation of a software claim with a section 112(f) limitation for indefiniteness with the following reasoning: "The specification merely provides functional language and does not contain any step-by-step process for controlling the adjusting means."[126] This statement is misleading because it implies that an algorithm is not expressed with functional language. What the Federal Circuit means by this statement and other similar ones is, more precisely, that the purported algorithm cannot merely provide functional language at the same level of generality as the functional language that forms the claim limitation. Occasional loose judicial descriptions notwithstanding, an algorithm is a functional entity that relates metaphorical, not physical, structure.

IV. The Definitional Question after *Williamson*: Pre-*Williamson* Doctrine Is Unhelpful

At first glance, *Williamson* might not appear to call for a revolution in the section 112(f) threshold test in the software arts. 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 was, in theory at least, already identifying software's structure in section 112(f) cases in two distinct doctrinal contexts even before *Williamson*. First, it was entertaining arguments about the presence or absence of structure in claim limitations that were intended to rebut the pre-*Williamson*, strong presumptions of the threshold test. Second, after limitations had been labeled as section 112(f)

[126] Ergo Licensing, LLC v. CareFusion 303, Inc., 673 F.3d 1361, 1365 (Fed. Cir. 2012).

limitations, it was looking for corresponding structure in the specification in the form of algorithms.

However, *Williamson* will in fact require a radical shift in the Federal Circuit's section 112(f) software doctrine. Upon closer examination, pre-*Williamson* cases turn out to provide surprisingly little guidance for answering the definitional question as part of the threshold test. Starting with the better-developed body of law, Section IV.A demonstrates that the Federal Circuit's existing doctrine identifying algorithms as software's structure cannot simply be portaged from the search for corresponding structure in a specification to the search for structure in a claim limitation. The principal problem is that the Federal Circuit's definition of an algorithm only defines structure in relation to the baseline of a particular claim, and there is no baseline in relation to which one can assess whether a claim itself recites an algorithm. If the Federal Circuit were to rely on this relative definition of structure to identify structure in the threshold test, all software limitations would become section 112(f) limitations.[127] Section IV.B then looks for crumbs of wisdom in the Federal Circuit's pre-*Williamson* cases that address the definitional question as part of the threshold test. Because the strong presumptions did most of the work, opinions addressing this issue are generally of little help in the post-*Williamson* world, but one line of reasoning does hint at a protean test for identifying a stand-alone definition of structure—that is, a definition that does not rely on a comparison to the claims as a baseline.

[127] Alternatively, the Federal Circuit could conclude that no software limitations are subject to section 112(f), but this outcome is unlikely. *See infra* __.

A. Algorithms as Corresponding Structure as a Model?

The Federal Circuit's identification of algorithms as corresponding structure in the specification for section 112(f) software limitations is by now well established.[128] One understandable, initial reaction to *Williamson*'s impact on software in the patent community might therefore be a collective shrug of the shoulders.[129] If there is a well-established answer to the definitional question in the context of identifying corresponding structure in the specification, then it might seem reasonable to assume that this answer could be simply carried over and used to identify structure in the claim limitations.[130] Upon closer inspection, however, this assumption proves to be unfounded. The Federal Circuit's cases that position algorithms as corresponding structure for limitations that are already governed by section 112(f) are unhelpful when the task at hand is determining whether there is sufficient structure in the limitations to avoid the application of section 112(f) in the first place.

The initial problem is that there is no single, functionally defined claim limitation that is potentially subject to section 112(f) that can ever be an algorithm, and thus structure, in and of itself. Any attempt to identify a single claim limitation as an algorithm is premised on 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."[131] Only

[128] *See supra* Section III.B.

[129] [KEC: any commentary that minimizes impact of Williamson?]

[130] *Cf.* Apple Inc. v. Motorola Inc., 757 F.3d 1286, 1294–1304 (2014) ("The overall means-plus-function analysis is a two-step process. Naturally, there is some analytical overlap between these two steps.").

[131] *See supra* __.

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.

To carry the Federal Circuit's algorithm-as-structure analysis over from the identification of corresponding structure in a specification to the identification of sufficient structure in a claim limitation requires a reconfiguration of the usual threshold test. The Federal Circuit cannot ask whether an individual claim limitation recites structure in the form of an algorithm, but rather whether a series of strung-together functional limitations in a claim constitutes an algorithm. 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. Each function recited as a claim limitation is usually analyzed independently to determine whether it has sufficient structure to evade the strictures of section 112(f), even when the claim states that the same "means" describes more than one function.[132] However, the search 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 the software arts.

Yet, even if one is willing to alter the search for structure in this software-specific manner, the Federal Circuit's case law on algorithms is decidedly unhelpful in the threshold test. Section IV.A.1 notes that the Federal Circuit has only defined an algorithm, and thus

[132] Media Rights Technologies v. Capital One Financial Corp., 800 F.3d 1366, 1373 (Fed. Cir. 2015); Noah Sys., Inc. v. Intuit Inc., 675 F.3d 1302, 1318–19 (Fed. Cir. 2012).

corresponding structure for software limitations, in a *relative* manner. Section IV.B.2 then demonstrates why a relative definition of structure is not up to the task of identifying sufficient structure as part of the post-*Williamson* threshold test.

1. The Federal Circuit's Relative Definition of an Algorithm

The Federal Circuit has only defined an algorithm in a *relative* manner. A functional description of a software invention in the specification 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.[133] One cannot identify an algorithm under the Federal Circuit's definition without first having identified the baseline task in the claim that the algorithm is supposed to perform.

The relative nature of the Federal Circuit's definition of an algorithm is on full display in the fact that it is impossible to examine a specification in isolation from the claims and meaningfully opine on whether or not it describes corresponding structure. The exact same specification could 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.[134] Assume that the specification states that the invention performs three functions: it controls a data output of a client system by diverting a commonly used data pathway of a media player application to a controlled data pathway, monitors the controlled data pathway, and disrupts the playing of the media content at the

[133] *See supra* notes __ and accompanying text.

[134] This technology in this hypothetical, 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).

controlled data pathway when the playing of the media file content is outside of the usage restriction applicable to the media file.[135] Is this a functional description of how the invention works that should count as an algorithm and thus corresponding structure? 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." The specification likely does describe an algorithm for this claim because it provides a series of more-granular steps that explains 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. Now, the unchanged specification does not describe an algorithm. Rather, it merely restates the functional tasks that are recited as claim limitations.[136] A relative definition of corresponding structure means that courts must recognize that the mystery of disappearing structure is a real phenomenon. In sum, the existence of an algorithm, and thus corresponding structure, is relative. A judge cannot look at a description of software functionality, standing alone, and reach a conclusion that the description does or does not describe structure. A judge can only identify structure in relation to the baseline functional description provided by the claims.

The Federal Circuit's use of a relative definition of corresponding structure for software limitations does not map neatly onto the nature of corresponding structure in other

[135] *Cf. id.* at 1369-70.

[136] *See supra* __ (noting that a restatement of the functional claim limitations is not an algorithm).

technologies. In the mechanical arts, the answer to the definitional question is never relative 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.[137] 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 what constitutes a structural description of a technology.

To be clear, despite its *sui generis* nature, the Federal Circuit's relative definition of structure in software is conceptually workable, if a bit awkward, when it is put to work in the specific context of identifying corresponding structure in the specification.[138] This search for structure is always conducted after a claim limitation has been held to be subject to section 112(f), so the functional description recited in the claim is always available to serve as a baseline. When searching for corresponding structure, the courts can always ask the definitional question in a relational manner: Given this particular claim limitation, does the specification disclose structure? Furthermore, a relative definition of corresponding structure enables section 112(f) to do important normative work in some software cases. By mandating that functional claim limitations be curtailed in scope to the disclosed algorithm and its

[137] *Cf. supra* __.

[138] However, the relative definition of structure is conceptually bankrupt when it is employed in legal doctrine for identifying whether there is sufficient structure in a claim limitation to evade the strictures of section 112(f). *See infra* Section IV.A.2.

equivalents, section 112(f) narrows the permissible scope of software limitations drafted in sweepingly general, functional terms.[139]

2. A Relative Definition of Structure in the Threshold Test

When searching for corresponding structure in a specification for a section 112(f) software limitation, a relative definition of structure is adequate because the claim identifies the relevant task and provides the needed baseline.[140] However, when searching for sufficient structure in a claim limitation as part of the section 112(f) threshold test, a relative definition of structure is not adequate. There is no baseline task to which to compare the series of functions recited as claim limitations to determine if they form an algorithm. Is any given set of functional claim limitations a string of structureless tasks, or should the set count as logical structure because it details a step-by-step, algorithmic procedure for achieving the over-arching task defined by the invention's utility?[141] The answer, of course, is that a string of functional limitations can be either one of these things, depending upon how one chooses to frame the task at hand. As explored below, the Federal Circuit can use its relative definition of structure, developed in the context of the search for corresponding structure, to identify sufficient structure in the claims in only two ways. It can achieve either one of two extreme outcomes: it can subject no software limitations to section 112(f), or it can subject all software limitations to

[139] Of course, it also narrows the permissible scope of software limitations that are drafted in quite specific, functional terms. *See infra* __.

[140] *See supra* Section IV.A.1.

[141] This question assumes that the claim recites more than one functional limitation. If a claim recites only one functional limitation, that limitation would automatically be a structureless task if the Federal Circuit's definition of an algorithm were to govern the search for structure in the section 112(f) threshold test. *Cf. supra* notes __ and accompanying text.

section 112(f). What the relative definition of structure does not permit, however, is for the Federal Circuit to toggle between these two outcomes in a principled fashion.

On the one hand, perhaps all software limitations are steps in an algorithm for accomplishing the task of providing the end user of the patented technology with some utility.[142] Following this approach, all functional software limitations would recite sufficient structure after *Williamson*, and no software limitations, excepting perhaps those that employ the term "means" and that are still subject to the strong presumption,[143] would be governed by section 112(f). This outcome would be problematic in light of the Federal Circuits expectation stated in *Williamson* that *Williamson*'s tilt toward scope theory will reduce the formalism of the threshold test and increase the number of software limitations subject to section 112(f).[144]

On the other hand, perhaps all functional software limitations refer to structureless tasks. Following this approach, no functional software limitations connote sufficient structure, so all such limitations would be subject to the scope-narrowing rules of section 112(f), regardless of whether they use the term "means." This route forward would satisfy the expectation that *Williamson* will increase the number of software limitations subject to section 112(f), and it would do so in spades. In fact, this route forward means that *Williamson* would have triggered a revolution in the law of software patents. *Williamson* would have caused a tidal shift in the law of software patents from most functional software limitations not being subject to section 112(f) to all such limitations being subject to section 112(f).

[142] Again, claims with only a single functional limitation would never recite structure under this approach. *See supra* note __.

[143] *See supra* __.

[144] *See supra* __.

Before moving on, it is worth pausing for a moment to recognize that, while using a relative definition of structure in the section 112(f) threshold analysis may yield an administrable rule that reduces the permissible scope of software claims from their pre-*Williamson* norm, it is awkward in another way. It places a formalistic, rather than substantive, limit on permissible patent scope.[145] A relative definition of structure does not cap permissible claim scope at any particular level of generality. Rather, it mandates that every functional claim limitation be narrowed to a disclosed algorithm and its equivalents, regardless of the level of specificity at which those limitations are drafted.[146] 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. However, 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, section 112(f) would require that each limitation in this claim, too, be restricted in scope to more granular, algorithmic steps (or sub-algorithmic steps, depending on your perspective) disclosed in the specification. For example, the limitation reciting function 1 would need to be restricted to the "structure" of steps 1a, 1b, and 1c.[147] The relative nature of the Federal Circuit's definition of structure is again on display: every set of functions that are corresponding structure when they are 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 disappearance of the structural nature of a description of a

[145] This is awkward in part because of the goals of *Williamson* was arguably to reduce the formalism of section 112(f) by shifting the threshold test from intent theory to scope theory. *See supra* __.

[146] Collins, *supra* note __, at __.

[147] If an algorithm were defined in a stand-alone manner and the functions 1, 2, and 3 remained an algorithm regardless of their context, the specification would have to disclose sub-algorithmic steps.

technology is not a harmless party trick. It reveals that section 112(f) operates in formalistic manner when it curtails permissible patent scope. 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 sets of technologies—should be subjected to the same restrictions because they raise identical policy concerns. However, when operating on the basis of a relative definition of structure, section 112(f) does not achieve this goal. The second claim construed under the default rules of *Phillips* would be coextensive with the first claim construed under scope-narrowing section 112(f).[148] Yet, the relative definition of structure means that the second claim, too, must be subject to section 112(f) and narrowed in scope to the more granular, (sub-)algorithmic steps disclosed in the specification. The Federal Circuit's relative definition of an algorithm formalistically narrows the scope of every functional software limitation, even if that limitation already recites a highly specific function.

B. The Pre-*Williamson* Threshold Test as a Model?

Before *Williamson*, the strong presumptions animated by intent theory did most of the work in the section 112(f) threshold test in software cases.[149] Nonetheless, the Federal Circuit did occasionally entertain arguments that there was sufficient structure in a claim limitation that recited "means" or insufficient structure in a claim limitation without the word "means," and these arguments were sometimes, although rarely, sufficient to carry the day. Because a

[148] Actually, the literal scope of the second claim would be narrower because it does not include the equivalents of steps 1, 2, and 3.

[149] Section 112(f) software cases only became pervasive after intent theory had become dominant, so the strong presumptions governed most of them.

relative definition of structure is useless in the section 112(f) threshold test,[150] these cases must have employed a stand-alone definition of structure of some kind. As such, they should, in theory, provide fodder for the stand-alone definition of structure that is needed after *Williamson* to administer a threshold test based on scope theory.

In actuality, however, the pre-*Williamson* cases addressing the threshold test offer little guidance for how to proceed in post-*Williamson* cases. Some opinions use conclusory reasoning, baldly asserting that what amounts to a programmed computer is actually sufficient structure.[151] Where there is more than conclusory reasoning, the reasoning is either misguided or protean. Section IV.B.1 addresses a first line of cases that derives from the use of the term "circuit" as a claim limitation and that unhelpfully reverts to the notion of software's physical structure. Section IV.B.2 turns to a second, and much more sparse, line of cases suggesting that functional claim limitations describing an operation's "input, output, and connections" or "how its output may be achieved" should count as logical structure under the section 112(f) threshold test. This definition of software's structure is radically underdeveloped, but it at least hints at what would be needed to formulate a stand-alone definition of structure.

1. Circuit as Physical Structure

A number of the pre-*Williamson* cases in which the Federal Circuit found sufficient structure in a claim limitation that lacked the term "means" to overcome the presumption against the application of section 112(f) derive from a single precedent: *Apex Inc. v. Raritan*

[150] *See supra* Section IV.A.2.

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

Comp.[152] *Apex* involved a patent on "computerized switching systems, known in the art as keyboard, video, mouse … switches, that allow centrally located network administrators to operate multiple server computers without requiring a complicated wiring scheme."[153] The claim contained, *inter alia*, a "programmable logic circuit" limitation for performing a variety of functions, including "control[ling] the programmable switch in response to the keyboard or cursor control device signals detected."[154] The district court held that this limitation, in addition to a number of others that also used the term "circuit," were governed by section 112(f).[155] The Federal circuit vacated the district court's holding, stating that the presumption against section 112(f) had not been overcome.[156] The Federal Circuit's reasoning relied on the fact that a circuit has physical structure because devices must be connected in a path:

> 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." *Dictionary of Computing,* 75 (4th ed.1996). In light of this definition, it is clear that the term "circuit," by itself connotes some structure. In the absence of any more compelling evidence of the understanding of one of ordinary skill in the art, the presumption that § 112, ¶ 6 does not apply is determinative.[157]

The Federal Circuit did "not find it necessary to hold that the term 'circuit' by itself always connotes sufficient structure," but rather held that "the term 'circuit' with an appropriate

[152] Apex Inc. v. Raritan Comp., Inc., 325 F.3d 1364, 1373–74 (Fed. Cir. 2003).

[153] *Id.* at 1367.

[154] *Id.* at 1368.

[155] *Id.* at 1369.

[156] *Id.* at 1371-75.

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

identifier such as 'interface,' 'programming' and 'logic,' certainly identifies some structural meaning to one of ordinary skill in the art."[158]

Based on its holding in *Apex* as precedent, the Federal Circuit proceeded in future cases to find sufficient structure in a number of software limitations lacking the word "means." Some of these limitations used the term "circuit," just as *Apex* did.[159] Other limitations, however, did not, and the Federal Circuit only analogized a software limitation to a hardwired electrical circuit structured in a path.[160] Whatever merit this invocation of the physical structure of a circuit has in relatively simple hardware devices, it has little to no relevance for software limitations in a post-*Williamson* world. The word "circuit" simply launders function into structure.

The oddity of the Federal Circuit's circuitry doctrine is on full display in the Federal Circuit's opinion in *MIT v. Abacus Software*.[161] The patent at issue claimed a system for producing color copies. Two of the claim's limitations were "aesthetic correction circuitry for interactively introducing aesthetically desired alterations into [the] appearance signals to produce modified appearance signals," and "colorant selection mechanism for receiving said modified appearance signals and for selecting corresponding reproduction signals … to produce

[158] *Apex*, 325 F.3d at 1373.

[159] *Linear Tech.*, 379 F.3d at 1320–21; MIT v. Abacus Software, 462 F.3d 1344, 1353–55 (Fed. Cir. 2006); Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc., 711 F.3d 1348, 1365 (Fed. Cir. 2013).

[160] Inventio AG v. ThyssenKrupp Elevator Americas Corp., 649 F.3d 1350, 1358 (Fed. Cir. 2011) ("In this case, the claims indicate that the 'modernizing device' functions as an electrical circuit that receives signals, processes signals, and outputs signals to other components in the patented system.").

[161] MIT v. Abacus Software, 462 F.3d 1344 (Fed. Cir. 2006).

… a colorimetrically-matched reproduction."[162] The Federal Circuit held that the former limitation connoted sufficient structure for the presumption against section 112(f) not to be overcome because "the term 'circuitry,' by itself, connotes sufficient structure" and that, again laundering function into structure, the "aesthetic correction" language, although functional, "adds further structure by describing the operation of the circuit."[163] In contrast, the presumption against section 112(f) was overcome for the latter limitation because the limitation failed to connote sufficiently definite structure. The Federal Circuit held that "[t]he term 'mechanism' standing alone connotes no more structure than the term 'means,'" and that the "colorant selection" modifier did not have a generally understood meaning in the art that connoted structure, either.[164] Here, the circuitry doctrine is simply an example of "the doctrine of magic words" in action.[165] There is nothing that is meaningfully more structural—at least, when structure is understood as physical structure—about the "aesthetic correction circuitry" limitation than the "colorant selection mechanism" limitation.

2. Inputs and Outputs as a Clue to Logical Structure

A second theme in the Federal Circuit's pre-*Williamson* cases addressing the section 112(f) threshold test offers a more promising starting point for developing a stand-alone

[162] *Id.* at 1348.

[163] *Id.* at 1355–56. The dissent argued that the limitation was a section 112(f) limitation. [] In response, the majority's rejoinder is that "the dissent appears to misapprehend the strength of the presumption that applies when the term 'means' does not appear in the claim." *Id.* at 1356.

[164] *Id.* at 1353–55.

[165] Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Cal. 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"). The legal significance of the "circuit" term could be explained by intent theory, but that is not a rationale that the Federal Circuit has ever offered. Rather, the Federal Circuit has insisted that "circuit" connotes physical structure.

definition of software's logical structure. This theme was first overtly articulated in Judge Reyna's majority opinion in *Apple v. Motorola*, issued only a year before *Williamson*.[166] In the course of explaining why the strong presumption against applying section 112(f) to a limitation not using the word "means" was not rebutted, Judge Reyna asserted that limitation recited sufficient structure because it described the limitation's "input, output, and how its output may be achieved."[167] In so doing, Judge Reyna laid some preliminary groundwork for a stand-alone definition of logical structure in software.

Among the many patents and issues addressed, *Apple* upheld several apparatus claims in Patent No. 7,479,949 on the use of finger contacts change the visual field or select files a touchscreen computer.[168] Each of the claims at issue describes one or more "heuristics" for determining that certain finger contacts correspond with certain commands. For example, 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 finger contact with respect to the touch screen display."[169]

[166] Apple Inc. v. Motorola Inc., 757 F.3d 1286 (2014).

[167] *Id.* at 1300.

[168] *Id.* at 1294–1304. The *Apple* opinion is much better known for its implications for injunctions for standard-essential patents. *Id.* at 1331–32.

[169] *Id.* at 1295. Another limitation recites the inverse heuristic that works in tandem with the heuristic in the text: "a two-dimensional screen translation heuristic for determining that … one or more finger contacts correspond to a two-dimensional screen scrolling command rather than a one-dimensional vertical screen translation command based on an angle of initial movement of finger contact with respect to the touch screen display." *Id.* A final limitation recites "a next item heuristic for determining that … one or more finger contacts correspond to a command to transition from displaying a respective item in a set of items to displaying a next item in the set of items." *Id.*

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."[170] Applying the section 112(f) threshold test, the district court then held the heuristics limitations to be subject to section 112(f), despite the then-present strong presumption that a limitation not using the world "means" is not a means-plus-function limitation, because the claims described functions "without describing the structure necessary to perform the functions."[171] According to the district court, the limitation would encompass any rule for distinguishing touches that generate vertical, one-dimensional screen scrolling from touches that would generate diagonal, two-dimensional screen scrolling.[172] Finally, the district court identified the corresponding structure: "a heuristic that uses as one input the initial angle of the user's finger swipe gesture and

[170] *Id.* Judge Posner sat by designation as the district court judge. *Id.* at 1294.

[171] *Id.* at 1295. At least in part, Judge Posner's reasoning in the threshold test seems to employ the very category error noted above that can be implicated by using the Federal Circuit's definition of an algorithm as a multi-step process to identify sufficient structure in a singular claim limitation. *See supra* __. When assessing whether each functional heuristic limitation in the claim individually was subject to section 112(f), Judge Posner identified an algorithm as the structure that must exist in a claim limitation to avoid section 112(f). Apple Inc. v. Motorola, Inc., No. 1:11-cv-08540 at 14–15 (N.D. Ill. March 19, 2012) (positioning an "algorithm, or 'step-by-step process,' for performing [the claimed] function" as "the 'structure' required by statute" in the claim to avoid the strictures of section 112(f)). Judge Posner concluded that "[t]he claims in patent '949 gesture toward such a step-by-step process, but don't describe one." It is not clear what it means for a single function recited in a claim limitation to be a multi-step algorithm (let alone what it means to "gesture toward" one). Apple Inc. v. Motorola, Inc., No. 1:11-cv-08540 at 7 (N.D. Ill. Jan. 16, 2012).

[172] In its reasoning explaining why the limitation is purely functional and lacks sufficient structure, the district court technically over-states the breadth of the limitation:

> It is inherent in the concept of a touch screen computer that the user would want to enter commands to tell it to do things, that he would enter those commands by moving his fingers on the touch screen, and that the device would need to apply some set of rules to a given finger input to determine which command was intended. Apple's patent cannot cover every means of performing the function of translating user finger movements into common computer commands on a touch-screen device—that would be a patent on all touch-screen computers.

Apple Inc. v. Motorola, Inc., No. 1:11-cv-08540 at 14–15 (N.D. Ill. March 19, 2012). The claim does not "cover every means of performing the function of translating user finger movements into common computer commands on a touch-screen device." *Id.*

determines whether that angle is within a predetermined range of being perfectly vertical, as shown for example in Figure 39C."[173]

In the Federal Circuit's majority opinion, Judge Reyna reversed the district court, holding that the claim limitation was not a section 112(f) limitation.[174] Judge Reyna concluded that the limitation connoted structure because it recited "the heuristics' operation within the context of the invention, including the inputs, outputs, and how certain outputs are achieved."[175] This inputs-and-outputs line of reasoning is problematic in some ways. Its support in Federal Circuit precedent is weak.[176] Furthermore, the claim of the '949 patent at issue in *Apple* arguably did not include inputs and outputs as limitations on claim scope. Judge Reyna arguably relied excessively on features of the preferred embodiments disclosed in the specification that do not actually limit claim scope to find that the claim language connoted structure.[177] Nonetheless,

[173] Apple Inc. v. Motorola, Inc., No. 1:11-cv-08540 at 7 (N.D. Ill. Mar. 29, 2012).

[174] *Apple*, 757 F.3d at 1294–1304.

[175] *Id.* at 1302; *see also id.* at 1299 ("Structure may … be provided by describing the claim's operation, such as its input, output, or connections."); *id.* at 1300 ("We find … that the '949 patent … describes the limitation's operation, including its input, output, and how its input may be achieved."); *id.* at 1303 (The limitation "describes [the invention's] operation, including its input, output, and how its output may be achieved."). Judge Reyna also argued "that 'heuristic' has a known meaning" that connotes structure. *Id.* at 1300. The opinion eventually concludes 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 section 112(f) had not been rebutted. *Id.* at 1301.

[176] *Apple* cites two cases to support its looking to the "input, output, and connections" of a program as a clue for identifying structure in a claim limitation. *Id.* at 1299. First, *Linear Tech. Corp. v. Impala Linear Corp.*, 379 F.3d 1311, 1320–21 (Fed. Cir. 2014), held that the presumption against section 112(f) was not overcome principally because the term "circuit" connoted tangible structure. *See supra* __ (discussing the section 112(f) "circuit" cases). However, *Linear* did 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 1310. Second, *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1361–62 (Fed. Cir. 2004), held that the presumption against section 112(f) was not overcome principally because the term "connector" had a generally understood meaning in the art as a name for physical structure.

[177] The claim language clearly recited the heuristic's "objectives." *Id.* 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 of the majority is that the majority relies too heavily on the preferred

despite these shortcomings, Judge Reyna's opinion is interesting because, unlike the rest of the Federal Circuit's pre-*Williamson* jurisprudence on the threshold test, it suggests one factor that might be relevant in developing a stand-alone definition of software's structure.[178]

The gist of Judge Reyna's reasoning seems to reside in the notion that functional claim language should count as a description of the logical structure of software if the function recited is sufficiently specific that describes how the invention works—that is, which it describes "inputs, outputs, and how certain outputs are achieved"—rather than what the invention does for its user.[179] This approach to answering the threshold test's definitional question is underdeveloped, but it at least hints at what is needed to move forward after *Williamson*, and its shortcomings are arguably endemic to any answer to the definitional question that formulates a stand-alone, rather than relative, definition of structure in software limitations.

---

embodiments in the specification in the section 112(f) threshold test. *Id.* at 1335 (Prost, J., concurring-in-part and dissenting-in-part) ("In effect, what the majority has done is imported the second step of the analysis (where you define the scope of a means-plus-function claim based on the corresponding structure in the specification) into the first step (where you identify whether the term is drafted in means-plus-function format."). This type of disagreement should not be surprising given that the Federal Circuit has not been clear about whether *Phillips* governs the search for sufficient structure in section 112(f) threshold test. *See supra* __.

[178] 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 section 112(f) status. Finjan, Inc. v. Proofpoint, Inc., 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.").

[179] The language used to convey this reasoning is a bit awkward. As the opinion puts it, "[s]tructure may also be provided by describing the claim's operation, such as its input, output, or connections. The limitation's operation is more than just its function; it is how the function is achieved in the context of the invention." *Apple*, 757 F.3d at 1299. This language is misleading because it suppresses 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* __. A more accurate way of describing the needed line in a stand-alone definition of structure in software is that the logical function that serves as structure is function, too, but that it is just a type of function that is sufficiently specific that it should count as metaphorical structure.

V. Towards a Stand-Alone Definition of Software's Structure

Williamson mandates a revolution in the law of section 112(f) as applied to software, but it is as-of-yet unclear which of two possible revolutions it will yield.[180] First, if the Federal Circuit were to import its already developed, relative definition of the structure of a software invention from the search corresponding structure in the specification into the threshold test, then the outcome, but not the doctrine, would be revolutionary. All functional software limitations would be subject to section 112(f), regardless of whether they employ the term "means."[181] Second, if the Federal Circuit were to develop a stand-alone definition of structure for the threshold test, then it would need a revolution in its section 112(f) doctrine. To date, the Federal Circuit has not sought to develop the stand-alone definition of structure that is needed for the threshold test to draw a principled line that subjects some, but not all, functional software limitations to section 112(f), although the discussion of inputs, outputs and connections in *Apple v. Motorola* hints at factors that could be considered as part of such a definition.[182] The remainder of this essay looks down the path that the Federal Circuit would have to follow if it were to meet *Williamson*'s challenge by revolutionizing its section 112(f) doctrine and developing a stand-alone definition of software's structure for the threshold test.

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

[180] These two options ignore the misguided possibility that physical structure is a relevant consideration outside of exceptional cases. *See supra* __.

[181] *See supra* __.

[182] *See supra* __.

the twentieth century require a threshold test motivated by scope theory.[183] When there is enough structure recited in the limitations of a claim, 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 overbroad patents.[184] The same policy concerns should frame the 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/abstraction, 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 this ladder at which a granular, functional description of a software program that constitutes logical structure transitions into a highly general, structureless description of a software program.[185] The descriptions both above and below the line are technically functional descriptions in a linguistic sense, but descriptions at or below the line

[183] *See supra* __.

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

[185] 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. Unlike a comparison to the baseline provided by a claim, the baseline does not shift dramatically from one section 112(f) analysis to the next.

count as metaphorical structure for the purposes of section 112(f). To reiterate, this line on the ladder of functional abstractions is not needed in the mechanical arts because section 112(f) can put all functional descriptions in the same bucket and require a dollop of physical structure as a limitation on permissible claim scope. However, because the physical, structural properties of a software invention are not relevant when describing what a software inventor has accomplished, the ladder-of-abstraction approach that discriminates among functional descriptions is the only way of developing the stand-alone definition of logical structure that is needed if section 112(f) is to subject some, but not all, software limitations to its scope-narrowing rule of claim construction.

In his work on functional claiming in software, Mark Lemley argues that section 112(f) must make a distinction between a "goal" (not structure) and a "way of implementing a goal" (structure),[186] or a "problem" the patentee solved (not structure) and "what the patentee … actually did" to solve the problem (structure).[187] The difficulty with this distinction is, of course, that goals and problems can be expressed in functional language couched at many different rungs on the ladder of abstraction. Your goal is likely to be 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.[188]

[186] Lemley, *supra* note __, at 947

[187] *Id.* at 963.

[188] *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 interaction of 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 patent describes a claim "purport[s] to improve the functioning of the computer itself" and thus 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.

What is needed in the post-*Williamson* world is a stand-alone definition of what constitutes a structureless problem or goal and, inversely, what constitutes a way of implementing a goal or a patentee's actual solution to a problem that should count as logical structure.

Importantly, once formulated, a stand-alone definition of structure reaches the same conclusion about whether a functional description counts as logical structure regardless of whether the description is a claim limitation (and the question is whether the claim connotes sufficient structure to avoid section 112(f)) or text in a specification (and the question is whether the specification discloses corresponding structure). The exact same definition of structure could apply in both contexts.[189] Furthermore, when used to identify corresponding structure, it should not make the existence of structure contingent on the claim limitation at issue. Some specifications that disclose corresponding structure under the Federal Circuit's definition of an algorithm will fail to disclose corresponding structure under the new stand-alone definition of software's structure. Even though a functional description in the specification is more granular than the functional description recited in the claims, the specification's functional description may still be too high up on the ladder of generality. Conversely, some specifications that fail to disclose corresponding structure under the Federal Circuit's definition of an algorithm because they simply repeat the functions that were recited as claim limitations will disclose structure under the stand-alone definition of software's structure. The functional description recited in both the claims and the specification may be sufficiently far down the ladder of generality to count as logical structure. In these cases,

[189] In theory, the Federal Circuit could adopt a stand-alone definition of structure for the threshold test and continue to use its relative 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 doctrinal contexts.

however, the presence of corresponding structure in the specification will technically be irrelevant because the presence of structure in the claims means that the limitations will not be subject to section 112(f) in the first place.

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.[190] In fact, it is the difficulty of a parallel task that led the Federal Circuit to rely on a threshold test motivated by intent theory to shelve the project of bringing section 112(f) to bear on step-plus-function limitations in method claims, despite the statutory requirement to do so.[191] In method claims, there is no clear, intuitive difference between a "step" and an "act," just like there is no clear, intuitive difference between functionally defined software limitations that should, and should not, be subject to section 112(f).[192] The only difference is a functional description's position on a ladder of generality. An "act" describes a more specific functional task than a "step" does, and several acts must be strung together to explain how to perform a step. 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, leaning heavily on intent theory to create an even stronger than usual presumption in

[190] Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930). The actual copyright doctrine that has developed in software copyright cases is helpful. 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. Computer Associates Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992). In patent law, section 112(f) must draw a line between claims to functionality that are sufficiently specific and those that are excessively general.

[191] *See supra* notes __ and accompanying text.

[192] 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 and Court Const., 172 F.3d 836, 848–49 (Fed. Cir. 1999) (Rader, J, concurring).

the section 112(f) threshold test that method claim limitations without the (rarely used) term "step" are not step-plus-function limitations.[193] Yet, this question parallels precisely the question that *Williamson* has now teed up for the Federal Circuit to answer in order to create a stand-alone definition of software's structure.[194]

The line marking a binary legal distinction on a continuous spectrum has no "natural" position on the ladder of abstractions. Nor is it feasible to ask the ultimate economic question—At what level of abstraction does a patent claim provide sufficient, but not excessive, reward to a software innovator?—as part of each and every patent examination procedure and invalidity defense. What is needed is an administrable proxy to replace the proxy of physical structure that is used in mechanical technologies but that is not available in software. The author has suggested elsewhere that one possible proxy uses consumer preferences as an anchor.[195] 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 particular ways of satisfying those desires should be seen as ways of achieving those goals and thus as logical structure. This proxy is far from straight-forward, but it at least provides an objective reference to which the stand-alone definition can be pegged.

---

[193] *Id.* at 849 ("The difficulty of distinguishing acts from functions … places a significant burden on the claim drafter to choose language with a definite and clear meaning. To invoke a presumption of § 112, ¶ 6 application, a claim drafter must use language that expressly signals the recitation of a function as distinguished from an act.").

[194] In fact, if *Williamson* applies to both apparatus and method claims, the Federal Circuit can no longer rely on a strong presumption against the application of section 112(f) to avoid having to draw a distinction between functions and acts in method claims.

[195] Kevin Emerson Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 Wash U. L. Rev. 1399, 1421–23, 1466 (2013).

The next step required to develop a stand-alone definition of structure for software under section 112(f) involves an interdisciplinary conversation among patent lawyers, computer scientists, and economists.[196] The desired output of such a conversation is a menu of levels of generality at which software can be described that is grounded in computer science, and an analysis of the implications of selecting any one of these levels—or a combination of them in different contexts—as a stand-alone definition of logical structure in software for the purpose of section 112(f). With a menu of options identified, and the consequences of choosing the different options clarified, the Federal Circuit will be well positioned to undertake the revolution in the section 112(f) law of software patents that *Williamson* mandates.

To be clear, the proposed conversation may not flush out any possible distinctions grounded in computer science that provide an acceptable answer to the definitional "What is structure?" question. It may be that, at the end of the day, the lesson learned from the conversation is that there are no viable distinctions upon which to build a reformed law of software patents that produce anything close to the desired amount of clarity. The purely functional nature of software may mean that section 112(f) just cannot regulate software patents in the way that it regulates mechanical patents.[197]

[196] Collins, *supra* note __, at __. Pam 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 Samuelson undertook. Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2323 (1994).

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

Conclusion

Williamson v. Citrix Online altered the section 112(f) threshold test for determining whether a functional claim limitation that not use the term "means" should be construed using the scope-narrowing rules of means-plus-function claiming. More specifically, *Williamson* expands the reach of section 112(f) by replacing a formalistic, strong-presumption test with a substantive, sufficient-structure test.

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 section 112(f). However, with respect to software patents in particular, *Williamson* actually signals a revolution because it will finally force the Federal Circuit to address the definitional "What is structure?" question in software.

One understandable, initial reaction to *Williamson*'s impact on software in the patent community might be a collective shrug of the shoulders. Before *Williamson*, there was already a well-established answer to the definitional question of software's structure in the context of identifying corresponding structure in the specification: software's structure is an algorithm, or step-by-step procedure, for performing the function recited as a claim limitation. However, this definition of an algorithm is relational, and a relational definition of structure is of little use in the *Williamson* threshold test. A reader cannot examine a series of functions in isolation and say that it is or is not an algorithm. Whether a specification recites an algorithm can only be determined by comparing the series of functions disclosed in the specification to the baseline

functionality recited as a claim limitation. Corresponding structure is defined only in relation to a particular claim; the exact same disclosure may be structure in relation to a more general claim and not structure in relation to a more specific claim. As a definition of corresponding structure, this relative definition of structure in software provides workable rule. However, when searching for structure in the claims, it is highly problematic because there is no extant baseline to which to compare the functional limitations.

Thus, *Williamson* will require a significant shift from the pre-*Williamson* conceptions of what constitutes the structure of a software invention. This shift may take either one of two different forms. Recognizing that an algorithm only provides a metric for identifying software's structure in the specification, the Federal Circuit could decide that software claims themselves never connote structure and that all functional software limitations are thus subject to section 112(f). Alternatively, the Federal Circuit could undertake the difficult task of articulating a novel, stand-alone definition of software's structure by drawing a line on a ladder of abstractions much like the idea/expression dichotomy does in copyright law. Either way, however, *Williamson* marks a revolution in the law of section 112(f) as applied to software.