6

A Hybrid Subject Matter

'Computer-related inventions are a subject matter with "two or more faces"'.[1]

INTRODUCTION

In some ways, the situation with software-related subject matter in the 1960s and 1970s was similar to the position in relation to microbiological inventions in the 1940s and 1950s – where the lack of an agreed taxonomic framework made defining and identifying patented microbiological inventions problematic – and with isolated genetic sequences in the early years of the twenty-first century – where the law was called upon to decide whether isolated genes should be seen in chemical or genetic terms (the outcome of which determined subject matter eligibility). While software shared things in common with both microbiological inventions and isolated genetic material, it differed in one important respect. This was because while with microbiological inventions and isolated genes, the problem that the law faced in dealing with the new subject matter was how it was to be characterised, with software-related subject matter the problem was more fundamental: there was no clear idea of what the subject matter was, let alone how it should be interpreted. As a patent examiner wrote in 1969, the most prevalent problem in the debate over the patentability of computer programs was the 'lack of effective communication between the parties involved'. This was primarily because there was no 'concrete, workable definition set forth by the computer or software industry for even the most basic of terms'.[2]

One of the consequences of the fact that the computing industry was divided about patentable subject matter was that in contrast to organic chemistry where the question of what the subject matter was and once this was decided how it was to be characterised, defined, and described was largely resolved by the relevant scientific communities and then adopted in the law, with software-related subject matter

[1] Morton C. Jacobs, 'Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyright and Patents' (1970) 1 *Rutgers Journal of Computers and Law* 50, 69.

[2] T. Buckman, 'Protection of Proprietary Interest in Computer Programs' (1969) *Journal of the Patent Office Society* 135, 138.

these questions were aired in legal fora. That is, instead of the industry or scientific community agreeing on what the subject matter was and how it was to be characterised, with software the industry attempted to work out its ontological issues *through* the law. As a result, the task of determining the nature and characteristics of software-related subject matter was treated as a *legal* problem to be resolved using *legal* terms. As a patent examiner wrote in 1969, 'any system that interfaces law and technology, such as the patent system, must necessarily use the sometimes burdensome language of the law. The computer industry should be no exception!'.[3]

The legal response to the question of how to approach computer-related subject matter can be broken down into two periods. The first, which spans the 1960s and 1970s and which is the subject of this chapter, saw patent law attempting to reconcile the conflicting views about what the subject matter was and how it should be interpreted. The situation changed in the early 1980s, however, as patent law took a more active role in thinking about computer-related subject matter. More specifically, the 1980s saw patent law come to view computer-related subject matter through the lens of 'abstractness'. As we will see in Chapter 7, it was here that we see the influence of materiality and its absence most clearly.

EARLY LEGAL RESPONSES TO SOFTWARE PATENTING

In 1961, two Mobil Oil Corporation engineers, Charles D. Prater and James Wei, lodged a patent application for 'improvement in the art of mass spectography'. The application, which was based on their discovery of 'a new way to analyze the output that is produced when a mass spectrograph measures a sample containing an unknown mixture of gases',[4] was examined and 'allowed' by the Patent Office on 22 September 1961. Before paying the final fee that would have triggered the grant of the patent Prater and Wei's patent attorney discovered a number of minor typographical errors in the application. To correct these mistakes and to add additional data, a continuation-in-part application (which is effectively a revised application) was filed. While the revised application was only 'imperceptibly' different from the original application, nonetheless it was rejected by the Patent Office.[5] In effect what had happened was that between the time when the initial application was filed in August 1960 and when the revised application was filed in November 1961, the approach of the Patent Office had changed. By the time the revised application was examined, the 'Patent Office had become concerned about the new technology of computer programming, especially if such applications were about to descend upon the Office in great numbers'.[6]

[3] Ibid.

[4] *In re Prater* (Prater I) 415 F.2d 1378 (CCPA 1968). See also *Application of Prater and Wei* (Prater II) 415 F.2d 1393 (CCPA 1969).

[5] Howard R. Popper, 'Prater II' (1970) 19 *The American University Law Review* 25.

[6] Ibid., 26.

The approach taken by the Patent Office to Prater and Wei's revised application was indicative of a trend that would continue across the 1960s and beyond.[7] While there had been what was described as 'encouraging dictum' at the Patent Office Board of Appeals for those seeking patent protection for computer programs,[8] the Patent Office consistently rejected software-related applications. As the lawyer who represented Applied Data Research, Morton Jacobs, said in 1965, when an application was seen to claim a computer program, the Patent Office examining staff tended to classify the application as non-statutory on the 'basis that they were for a system of knowledge (like mathematics), rather than an industrial process'.[9]

While the approach of the Patent Office towards software patents in the early half of the 1960s was (fairly) consistent, there was still some confusion. In order to clarify the standing of software in patent law, Patent Office Examination Guidelines were drafted in 1966, which distinguished between software as a process and software as a device.[10] According to the draft Guidelines, as a process, computer programs 'are written in terms of algorithms rather computer component changes and, therefore, are not statutory subject matter'.[11] Building on the idea that algorithms 'are conclusions based upon a precise or mathematical premise and line of reasoning'[12] and the uncontroversial proposition that mathematical process, discoveries, and mathematical formula were not patentable, the Guidelines proposed that as a process computer programs were not patentable because they were mere mathematical or mental steps. While these processes may have been useful and important, nonetheless they were non-patentable on the basis that they were 'merely expressions of an algorithm'.

In contrast, the draft Guidelines proposed that programs that (i) controlled the changes in state of components of the computer itself and (ii) transformed a specific machine from a general-purpose to a specific-purpose device should be potentially patentable. That is, as a device for controlling the operation of a general purpose computer that dealt with 'tangible things and substances' (the later were called patentable 'utility processes'), programs were patent eligible.[13] In explaining the Guidelines the Commissioner of Patents, Edward J. Brenner, said that while 'program' had been defined loosely by the parties, the Office did not think it was necessary to define program (or computer) since these were merely 'adaptions of the concept of inventions of "automatic control", which

[7] *In re Prater* 415 F.2d 1378, 1390 (CCPA 1969). Rich J. dissenting from grant of rehearing noted in relation to patentability of software that 'the Patents Office's policy of refusing to follow what this reviewing court has now declared the law to be and to have been, at least since 1952'.

[8] Michael I. Rackman, 'The Patentability of Computer Programs' (1963) *New York University Law Review* 891, 893–94.

[9] Morton C. Jacobs, 'Patent Protection for Computer Programs' (1965) 47 *Journal of the Patent Office Society* 6, 10.

[10] 'Guidelines to Examination of Programs' (9 August 1966) 829(2) *Official Gazette of the United States Patent Office* 441.

[11] Ibid., 441–42.

[12] Ibid., 441.

[13] Ibid., 442.

the patent office already considered as patentable subject matter';[14] these included the Jacquard looms (class 139 Weaving, subclass 59) which have 'presented for many years the concept of processes and apparatus that include a program'. Subtly shifting the focus of attention from the program to the programmed computer, Brenner said that a machine that includes a 'program device' that causes a machine as a whole to function falls within the patent statute the same as any other special purpose machine. The fact that portions of the completed machine take the form of a replaceable program is of no moment'.[15] It did not matter 'whether a "program device" is termed a Jacquard card belt, a player piano roll, a plug-board or a magnetic tape and the corresponding "program" is termed a weaving design a musical composition, a switching scheme or a document listing a series of instructions which a machine will execute'.[16] In all cases, the special purpose machine was patent eligible subject matter.

The draft Guidelines were discussed by over a hundred people at a public hearing held at the Patent Office in October 1966 to ascertain the 'present law on patenting of programming'. All of the speakers at the hearing 'opposed adoption of the proposed Guidelines'. In reflection of the different approaches taken towards the patenting of software they were divided, however, 'on whether the Guidelines would or should authorise the issuance of patents on computer programs'.[17] While the Guidelines had been written to clarify the state of the law, it was said that the only effect of the guidelines, which had 'raised a small storm of protest in Washington', was that they 'succeeded in riling both the proponents and the proponents to the patenting computer programs'.[18] While 'Bell Laboratories felt the proposed guidelines were too restrictive … IBM felt the proposed guidelines were too broad'.[19] Given that no one supported the Guidelines,[20] it is not surprising that they never came into force.[21]

At the same time as the Patent Office was attempting to develop Guidelines for the examination of programs, the standing of computer programs was also being

[14] Ibid., 441.
[15] Ibid., 442.
[16] Ibid., 441.
[17] Anon, 'Patent Office Holds Hearing on Computer Programming Patents' (November 1966) 6(2) *The New York Patent Law Association Bulletin* 6.
[18] Elmer W. Galbi, 'Software and Patents: A Status Report' (1971) *Communications of the ACM* 274.
[19] Ibid. For some, the draft Guidelines provided a limited form of protection for programs. Statement of Richard C. Jones, President, Data Research, Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate (Nineteenth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 2 (1–2 January, 1 February 1968), 751).
[20] Morton C. Jacobs, 'Commissions Report (re: Computer Programs)' (1967) *Journal of the Patent Office Society* 372.
[21] Elmer W. Galbi, 'Software and Patents: A Status Report' (1971) *Communications of the ACM* 274. Congressman Brooks (from Texas), who sat on the Judiciary Committee which was planning to holding hearings in 1967 to revamp patent law and who was reportedly 'alarmed at the prospect of patents on computer programs' convinced the Department of Commerce to set aside the draft Guidelines (pending the more wide-ranging review). James P. Titus, 'Pros and Cons of Patenting Computer Programs' (February 1967) 10(2) *Communications of the ACM* 126.

looked at by the *President's Commission on the Patent System*, which had been established by President Lyndon Johnson in July 1965 to undertake a wide-ranging review of patent law in the United States. The report of the Presidential Commission on the Patent System was released in December 1966. One of the findings of the Commission was that patents should *not* be granted for computer programs (which were defined as a 'series of instructions which control or condition the operation of a data processing machine').[22] A number of reasons were given for the decision, including the administrative difficulties that the Patent Office would have experienced if protection was allowed. Specifically, it was thought that as patent review searches would neither be feasible nor economical given the amount of prior art and that the 'current classificatory techniques and search files were inadequate' that the patent examination process would be put under great stress if protection was allowed. The Commission also noted that programs had grown satisfactorily in the past without protection and that, in any case, copyright protection was available.[23]

In reviewing the existing law, the Commission noted that attempts to patent programs per se had been rejected on the ground that they were not statutory subject matter. They also noted that '[i]ndirect attempts to obtain patents … by drafting the claims as a process, or a machine or components thereof programmed in a given manner rather than as a program itself, have confused the issue further and should not be permitted'.[24] To avoid this confusion, the Commission said that programs should not be patentable whether claimed as an article, a process described in terms of the operation performed by a machine pursuant to a program, or one of more machine configurations established by a program.[25]

The recommendation of the Presidential Commission in relation to software found its way into the *Patent Reform Bill* of 1967, which expressly excluded computer programs from patentable subject matter. Specifically, section 106 of the Bill provided that a 'plan of action or set of operating instructions, in whatever form presented, to cause controllable data processor or computer to perform selected operations shall not be patentable'. While discussions of software patenting at the President's Commission had been dominated by hardware manufacturers,[26] the draft

[22] *Report of the President's Commission on the Patent System* (1967), 20.

[23] Ibid. The Justice Department, saw program patent as being inherently anticompetitive, argued against protection. They also reminded the Patent Office that monopolies were the exclusive business of the Justice's Antitrust Division. James P. Titus, 'Pros and Cons of Patenting Computer Programs' (February 1967) 10(2) *Communications of the ACM* 126.

[24] *Report of the President's Commission on the Patent System* (1967), 21.

[25] Ibid., 20. Harold L. Davis, 'Computer Programs and Subject Matter Patentability' (1977–78) 6 *Rutgers Journal of Computers and Law* 1, 9 n 46.

[26] While the hardware manufacturers were represented on the committee by James Birkenstock (IBM, Vice President, Commercial Development) and by Bernard Oliver (Hewlett-Packard, Research and Development), the software industry was not represented. Software manufacturers also did not make submissions to the President's Commission. Brief Amicus Curiae for the Association of Data Processing Service Organisations, Software Products and Service Section, *Gottschalk* v. *Benson*, Supreme Court of the US, Oct Term, 1971, No 71–485, 19.

Act attracted the interest of the proponents of patent protection who argued against the proposed changes.[27] Section 106 was opposed by a number of software firms, the American Patent Law Association, the Electric Industries Association, the American Chemical Society, and the National Small Business Association.[28] It also seems that the administration had a change of heart, which was reflected in the fact that neither the President's Science Advisor nor the Assistant Attorney General testified in support of the proposed new law.[29] Interestingly, although the Patent Office had helped to prepare the legislation[30] and section 106 was said to have codified Patent Office practice,[31] nonetheless the Commissioner of Patents, Edward J. Brenner, also argued against section 106. While he noted that computer programs 'were not patentable under the present law, and we shall continue to deny applications for patents on computer programs per se', the Patent Office felt that there were 'substantial difficulties in finding an adequate definition for computer programs'. On this basis, Brenner said that it was 'premature to enact legislation at the present time'.[32] Similar complaints about the breadth of section 106 and the difficulties of defining computer program were made by a range of other parties. Following this hostile reaction, section 106 was removed from the Patent Reform Bill of 1967.[33]

Despite the concerns that had been raised about the decision to classify software as patent ineligible subject matter, the Patent Office reconfirmed its earlier anti-software approach in October 1968 when it issued new Examination Guidelines that said 'computer programming per se, whether defined in the form of process or

[27] Discussed as part of the wide-ranging review of patents that took place by the Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 1 (17, 18 May 1967), Part 2).

[28] William D. Smith, 'Fighter for Computer-Program Patents' (29 December 1968) *The New York Times* 19.

[29] Morton C. Jacobs, 'Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyright and Patents' (1970) 1 *Rutgers Journal of Computers and Law* 50, 58.

[30] Ibid., 57.

[31] Edward J. Brenner, (Commissioner of Patents), *Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary, United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 1 (17, 18 May 1967), 137).

[32] Edward J. Brenner (Commissioner of Patents), *Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 2 (1–2 January, 1 February 1968), 394). Brenner asked that it be recorded that the 'omission was not intended to pass judgement on the question of the patentability of computer programs' (ibid). The Electric Industries Association also counselled against the adoption of a legislative because the definition 'posed extreme uncertainty' (ibid., 516). While the American Chemical Society 'took no position on the question of the patentability of inventions with computer programs', they suggested that the definition used in the section 'would appear to prohibit patents on any chemical process which is ordinarily carried out with automated equipment'. Robert W. Cairns, President American Chemical Society, ibid., 533–3.

[33] This, in turn, gave rise to further debate about whether the removal of section 106 was an indication of positive support for patent protection or that it was simply too early to make normative judgements about the fate of software.

apparatus, shall not be patentable'.[34] (Embarrassingly, Goetz's patent for Autoflow, which was heralded at the time as the first software patent, was granted shortly after).[35] Two related arguments were used by the Patent Office to reject software-related claims.[36] First, programs were denied protection on the basis that they could not satisfy the 'change of state' doctrine, which specified that to be patentable processes needed to operate physically on substances.[37] Programs – which were seen as a series of mental, mathematical steps – were also refused protection on the basis of the mental steps doctrine, which denied patents to processes that could be performed by or required the use of human intellect.[38]

While the reason given by the Patent Office for excluding software was that it lacked the requisite physical indicia and that it was a non-patentable mental process, the anti-software approach of the Office was also motivated by another concern: namely, a concern that if patent protection for software was allowed it would have increased the workload of the Patent Office at a time when the Office was facing a public backlash because of the time it was taking to process patents. As a patent examiner said, it was feared that software patenting would have 'imposed a tremendous burden at a time when [the Patent Office was] 'desperately trying to decrease its backlog'.[39] Interestingly, the backlog at the Office also played a role in determining the basis on which software would be denied protection (namely, subject matter ineligibility). As a patent examiner explained in 1968, the 'need for a more summary treatment of software (program) claims arose with the advent of the computer, since

[34] 'Guidelines to the Examination of Programs' 855 Official Gazette of the United States Patent Office 829–30, 33 Fed Reg 15609, 15610 (1968) (22 October 1968). See William D. Smith, 'Fighter for Computer-Program Patents' (29 December 1968) *The New York Times* 19.

[35] Robert F. Brothers and Alan M. Grimaldi, 'Prater and Patent Reform Proposals' (1969) 17 *Catholic University Law Review* 389.

[36] Elmer W. Galbi, 'Software and Patents: A Status Report' (1971) *Communications of the ACM* 274, 275.

[37] Howard R. Popper, 'Prater II' (1970) 19 *The American University Law Review* 25, 28. This was based upon the Supreme Court in *Cochrane* v. *Deener* that a 'process is a mode of treatment of certain materials to produce a given result. It is an act, or a series of acts, performed upon the subject matter to be transformed and reduced to a different state or thing'. 94 U.S. 780, 788 (1876).

[38] T. Buckman, 'Protection of Proprietary Interests in Computer Programs' (1969) *Journal of the Patent Office Society* 135, 144 (Buckman was a Patent Office examiner). On the role of mental steps in relation to software, see Samuel J. Sutton Jr, 'The "Mental Steps" Doctrine: A Critical Analysis in the Light of Prater and Wei' (1969–70) 13 *Patent, Trademark and Copyright Journal of Research and Education* 458; Virgil E. Woodcock, 'Mental Steps and Computer Programs' (1970) 52 *Journal of the Patent Office Society* 275.

[39] Robert W. Wild, 'Computer Program Protection: The Need to Legislate a Solution' (1969) 54(4) *Cornell Law Review* 586, 604. On the backlog see Robert A. Choate, 'Backlog' (1966) 48 *Journal of the Patent Office Society* 274; *Official Gazette Patent Office* 668 (1968), 187; W. Scott Railton, 'The Examination System and the Backlog Problem' (1965–66) 9 *Idea* 487. The backlog had risen to 216,00 applications in July 1964, and had been going backwards at around 10,000 applications annually (at 493). By 1967, the delay had been reduced from 'about 3.5 years down to about 2.5 years'. Edward J. Brenner, (Commissioner of Patents), *Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary, United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 1 (17, 18 May 1967), 128).

rejection for lack of novelty became entirely too involved'. In contrast, 'non-statutory holding was concluded to be an applicable basis for summary, no-examination treatment (along the lines of methods of doing business, printed matter, mathematics, etc) … There was always a lack of capacity (personnel shortage, lack of training, lack of prior art files) to perform the conscientious, meaningful examination'.[40]

For the most part, courts across 1960s accepted the Patent Office's argument that software should not be patentable. In a series of decision at the end of the decade, however, the Court of Customs and Patent Appeals overturned the approach that had been adopted at the Patent Office towards the patenting of software. The judicial change in approach was heralded by the 1968 decision of *Prater and Wei*, which concerned the application by the Mobil Oil Corporation engineers, Prater and Wei, for an 'improvement in the art of mass spectography'. As we saw above, while the initial application was allowed, the revised application was denied. After the Patent Office Board of Appeals affirmed the Patent Office decision to reject the application (on the basis that as the applicant's claims could be performed mentally they were not patentable subject matter), an appeal was made to the Court of Customs and Patent Appeals. In what was heralded as a landmark decision written by Judge Arthur M. Smith, the court reversed the decision of the Board of Appeals and upheld the validity of the patent.[41] While the decision was hailed by software companies as a 'magna carta'[42] that allowed them to compete with hardware manufacturers, the Patent Office feared that it would lead to the 'demise of an effective patent system'.[43] In order to end the 'path of destruction' that had been started by *Prater*,[44] the Patent Office petitioned for a rehearing, which was granted.[45] The matter was reheard and the case was decided anew (*Prater* II).[46] To the annoyance of hardware manufacturers, the court in *Prater* II followed the court in *Prater* I and upheld the validity of the apparatus claim (using a style of drafting that I look at in Chapter 7). The court also said that the mere fact that a process could be carried out by mental steps or performed in the mind was not a bar to patentability.[47]

[40] L. Smilow, 'Comments on Computer-in-Law Institute's First Annual Conference' (November 1968) 50 *Journal of the Patent Office Society* 779, 780–81 (Smilow was a primary examiner at the Patent Office).

[41] *In re Prater* (Prater I) 415 F.2d 1378 (CCPA 1968). Judge Smith died the day after the opinion was handed down.

[42] Stacy Jones, 'Computer Programs Are Held Patentable: An Appellate Court Decides Case Concerning Software' (16 August 1969) *The New York Times* 35.

[43] T. Buckman, 'Protection of Proprietary Interests in Computer Programs' (1969) *Journal of the Patent Office Society* 135.

[44] Ibid.

[45] 160 USPQ 230, 415 F.2d 1390 (CCPA 1969).

[46] *Application of Prater and Wei* (Prater II) 415 F.2d 1393 (CCPA 1969).

[47] *In re Prater* (Prater I) 415 F 2d 1378, 1389 (CCPA 1968). Described as an anthropomorphic view of the new equipment and processes. James B. Gambrell and Irving Kayton, 'Patent Law in Perspective 1967' (1967) *The George Washington Law Review* 545, 551. See also Brief Amicus Curiae for Applied Data Research, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 11 n 13. ("Technical jargon in the computer field, which is quite often anthropomorphic in suggesting human

One of the notable things about *Prater* I and II is that they removed much of the 'semantic thicket surrounding the "mental process" and non-statutory subject matter lines of rejection'[48] by holding that the mental step and change of state cases were not bars to the patentability of software. Specifically it was said that to be patentable a process did not need to physically operate upon substances.[49] The rejection of the change of state and mental step cases as potential grounds of objection paved the way for a more positive approach to software patenting. This was reflected in the comment by the court in *Prater* II that '[n]o reason is now apparent to us why, based on the Constitution, statute, or case law, apparatus *and* process claims broad enough to encompass the operation of a programmed general-purpose digital computer are necessarily unpatentable'.[50] The court went on to say '[i]n one sense, a general purpose digital computer may be regarded as but a storeroom of parts and/or electrical components'. However, 'once a program has been introduced, the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit with or without electro-mechanical components) which, along with the process by which it operates, may be patented subject, of course, to the requirements of novelty, utility and non-obviousness'.[51]

Following *Prater II*, in October 1969 the Patent Office begrudgingly withdrew the Examination Guidelines that provided that computer programming was not patentable.[52] While the Commissioner announced that the Patent Office would henceforth look at applications for computer programs on a case-by-case basis,[53] the Office continued to reject applications for process claims (either on the basis of the mental steps doctrine[54] or in a shift away from subject matter on the basis that the invention was not adequately disclosed in the patent).[55]

Frustrated with the ongoing rebuffs by the Court of Customs and Patent Appeals and a belief that the Court was inappropriately attempting to legislate

characteristics where none exist draws upon the only available terminology we have … The hardware manufacturers arguments about "mental processes" are largely and speciously based on this anthropomorphic terminology').

[48] Howard R. Popper, 'Prater II' (1970) 19 *The American University Law Review* 25, 27.

[49] *In re Prater* (Prater I) 415 F.2d 1378, 1388 (CCPA 1968). T. Buckman, 'Protection of Proprietary Interests in Computer Programs' (1969) *Journal of the Patent Office Society* 135, 146.

[50] *Application of Prater and Wei* (Prater II) 415 F.2d 1393; 162 USPQ 541, 549, n 29 (CCPA 1969).

[51] Ibid. The Court of Customs and Patent Appeals went one step further shortly after in *re Bernhart and Fetter*: 'If a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program, its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should not tempt us to conclude that the machine has not changed'. 163 USPQ 611 (CCPA 1969).

[52] 34 Fed Reg 15724 (1969) (Commissioner William E. Schulyer).

[53] Ibid.

[54] Pauline Wittenberg, 'Computer Software: Beyond the Limits of Existing Proprietary Protection' (1973) *Brooklyn Law Review* 116, 131 n 118.

[55] Howard R. Popper, 'Prater II' (1970) 19 *The American University Law Review* 25, 27.

the patentability of computer programs,[56] the Patent Office decided to appeal the question of software patenting to the Supreme Court. The decision that the Patent Office selected for appeal was Benson and Talbot's 1963 application for a process where a general purpose digital computer was programmed with an algorithm that converted binary coded decimals to pure binary numbers. Given that the process could be performed manually using pen and paper, the Patent Office and the Patent Office Board of Appeals rejected the application on the basis that it was a mere mental process. Following the approach in *Prater*, the Court of Customs and Patent Appeals reversed the finding of the Board of Appeals, holding that while the process could be performed mentally, as no mental steps were required in the proposed method it was patentable. In an attempt to overturn the pro-patent stance that had been adopted at the Court of Customs and Patent Appeals, the Patent Office appealed the decision to the Supreme Court. As with *Prater* II, the appeal attracted a lot of industry interest (including 14 amicus curie briefs).[57]

While it had been hoped that the Supreme Court would have provided much-needed clarity about whether or not, and if so in what circumstances software might be patentable, when the 1972 decision of *Gottschalk* v. *Benson* was handed down, it readily became apparent that the decision only served to reinforce the existing confusion.[58] In part, this is because the court sidestepped the question of whether software was patentable and focused instead on the specific facts of the case – on the question of whether a process claim directed to a numerical algorithm was patentable. In relation to this point, the court found that 'the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself'. In response to the request that had been made by the hardware manufacturers in their amicus curia for the court to declare that 'the decision precludes a patent for any program servicing a computer', the court responded: 'we do not so hold'. In so doing the court left open the general question of whether software was patentable.

While it is sometime suggested that *Gottschalk* v. *Benson* decided that software did not qualify as patentable subject matter,[59] at the time the decision was handed down it was not clear what the outcome of the decision was. Indeed, one of the

[56] T. Buckman, 'Protection of Proprietary Interests in Computer Programs' (1969) *Journal of the Patent Office Society* 135, 147.

[57] It was said that *Benson* was an experiment in the limits of patent drafting to test whether embodied software was necessary (patent did not disclose a machine – program was designed only to manipulate numbers). See Robert D. Nimtz, 'Computer Applications and Claim Drafting under Current Law' in (ed) Irving Kayton, *Software Protection* (Washington: Patent Resources Group, 1969), 242, 252. For discussion see Gerardo Con Diaz, 'Embodied Software: Patents and the History of Software Development' (July–September 2015) *IEEE Annals of the History of Computing* 8, 16.

[58] See Kenneth Nichols, *Inventing Software: The Rise of Computer-Related Patents* (Westport, CT: Quorum Books, 1998), 16.

[59] See G. A. Stobbs, *Software Patents* (New Jersey: Wiley Law Publications, 1995); Robert P. Bigelow, 'Infosystems, the Law and Patents' (1973) *Jurimetrics* 129.

things that is apparent from contemporaneous accounts of the decision is that there were very different understandings of what was decided. While some interpreted the decision as 'effectively deny[ing] patent protection to all software claims',[60] others suggested that all that the Supreme Court had decided was that the particular algorithm in question could not be patented as a process.[61] The different ways in which the decision could be read was highlighted by articles in successive editions of the trade magazine *Computerworld*, which said that *Benson* precluded[62] and allowed[63] patent protection for computer software. In a decision that seemed to have something for everyone, it was suggested that there was text in *Benson* to support both of these incompatible views.[64] The confusing nature of the decision was reiterated in a comment by Justice Rich who said that he was 'probably as much – if not more – confused by the wording of the *Benson* opinion as many others … I have no idea what was in the collective mind of the … Court'.[65]

The Supreme Court decision set the tone for the way patent law approached computer-related subject matter for the remainder of the decade. As in the past, the law's response to the computer-related subject matter remained inconsistent, unclear, and unsettled. At times, such as in the *Presidential Commission on the Patent System*, the 1967 *Patent Reform Bill*, and the 1968 *Patent Office Examination Guidelines*, patent law embraced aspects of the hardware manufacturer's way of construing the subject matter. At other times, there was support for the approach favoured by software companies. For example, in several cases, including *Prater*, the Court of Custom and Patent Appeals sided with software companies and accepted the equivalence of a programmed general-purpose computer with a unique single purpose machine. While there was scattered support for both approaches, there was no overall agreement either about what the subject matter was nor about how it should be interpreted. As is often the case when the law first grapples with a new subject matter, the language used to describe computer-related subject matter was fluid and changing. The confusion was exacerbated by the widespread use of 'software' as a catch-all term for computer-related subject matter, even when talking about very different things. One of the consequences of this was that people often talked at cross-purposes[66] and 'read and understood patents and judgements

[60] Mary Jane Gaskin, 'In re Johnston: New Output by the CCPA on the Patentability of Computer Software' (1975) 36 *University of Pittsburgh Law Review* 739.

[61] See, e.g., Robert P. Bigelow, 'Infosystems, the Law and Patents' (1973) *Jurimetrics* 129, 130.

[62] *Computerworld* (29 November 1972), 1, col 3.

[63] *Computerworld* (13 December 1972), 37.

[64] Robert M. Milgrim, 'Software, Carfare and Benson' (1973) *Jurimetrics* 240.

[65] *In re Johnston* 502 F.2d 765, 773–4 (CCPA 1974). The reasoning is 'monstrously bad'. Donald S. Chisum, 'The Patentability of Algorithms' (1986) *University of Pittsburgh Law Review* 959, 977–78.

[66] In response to the 1967 testimony by Commissioner of Patents, Edward J. Brenner, to the House Judiciary Committee that the 'Patent Office has taken the view that computer programs are not patentable under present law, and no patent has been issued on a computer program per se', it was said that since it was not clear what was meant by 'computer' and 'program' that the claim was 'highly

differently'.[67] These problems were exacerbated by the fact that, at least from software producer's perspective, the patents that made their way to the courts for review in the 1960s and 1970s were the wrong type of subject matter. As Martin Goetz complained, the cases before the Court of Customs and Patent Appeals in the 1960s 'were not representative because they were industrial companies that had filed for patents and as part of the patent there was a computer. But they were not software companies that were filing for patents. These patents were usually controlling the machine, using an industrialized process … that included a computer program'. The patents had 'nothing to do with the software business, except one of the claims … involved software'.[68] The upshot of this was that patent law was unable to reach agreement about what the subject matter was, let alone how it should be dealt with.

Given this confusion, it is not surprising that commentators began to look elsewhere to regulate computer-related innovations. While a number of options were mooted, the most prominent and important change that occurred at the time was that the 'computer program *per se*' came to take on a special role in intellectual property law. The growing attention given to the computer program in patent law was primarily a consequence of changes in copyright law. In part this was a result of the fact that US, foreign, and international copyright practice adopted the computer program as the archetypical subject matter.[69] At the same time, there was also a growing expectation that the computer program would operate as a boundary object that regulated the divide between copyright and patents. In part this built upon the fact that it was accepted that computer programs, as descriptions of sets of machine instructions, were not and should not be patentable subject matter. Instead, they belonged if anywhere within copyright law. In effect what occurred was that the computer program was separated out, almost fetishized, and given pride of place as a discrete and distinct object amongst the myriad of things that fell within the field of computer technology. While the shift was never complete – software companies, in particular, consistently spoke about programmed machines and commentators still talk today about patenting software – the computer program came to occupy a

speculative'. Robert O. Nimtz, 'Computers, Programs and the Patent Laws' (1966–67) *Idea* 199, 207. Brenner testified that 'the Patent Office has not issued patents for computer programs per se'. Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary, United States Senate (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 1 (17, 18 May 1967), 137).

[67] Brief Amicus Curiae for Mobil Corporation, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 19.

[68] ADAPSO History Program, Interview with Martin Goetz (3 May 2002) (interviewed by Jeffery R. Yost), 14–15.

[69] In 1964 the Copyright Office defined a computer program as 'either a set of operating instructions for a computer or a compilation of reference information to be drawn upon by the computer in solving problem'. Copyright Office Announcement SML-47, May 1964; Copyright Office Circular 31D (January 1965).

special place within discussions about computer-related subject matter. As discussions were re-orientated to focus on the computer program, the computer program became the lens through which discussions about computer-related subject matter were framed (at least for a time).

Within patent law, the pride of place given to computer programs was reflected in the way that the Patent Office spoke about computer-related subject matter, in official inquires (such as the *Presidential Commission on the Patent System* and the 1967 *Patent Reform Bill*), and in the way many doctrinal accounts of the field were organized. In these accounts, the computer program (or the algorithm that was thought to underpin it) typically came to be seen as the 'intellectual heart of computer operation'.[70] The computer program also became the lens through which patents were viewed. As Martin Goetz, who is often credited with being granted the first patent on a computer program said, there was 'a lot of confusion because people thought of it as getting a patent for a program, which was not the case, because my patent … was for a sorting process'.[71] At times, the focus on the computer program became so dominant that software-related patents that did not expressly claim a computer program were criticized for obfuscating and disguising the 'true nature of their contribution by garbing the patent claims with recitations that appear to be directed to hardware components of digital computers and digital computer operations'.[72] The preoccupation with the computer program also came to shape the way that the history of software patenting has been viewed. To the extent that these histories move beyond the personal computer they tend to reach back to 1968, the year in which the 'first' software patent was granted to Martin Goetz. While this reading has been challenged on two fronts – first, by Goetz himself who (at least in some situations) has questioned whether his patent was for a computer program and more recently by Gerardo Con Diaz – nonetheless the history still centers on computer programs and software.[73]

While subject matter eligibility in US patent law was and remains a creature of jurisprudence, over the 1960s and 1970s the computer program took on a life of its own as it was entrenched in a network of formal and informal legal settings; a process that reinforced the expectation that the computer program would operate (at least ostensibly) as a boundary object to police the limits of software-related subject matter. This occurred as a result of a series of institutional, bureaucratic, and juridical changes in the United States, in other countries (notably in Europe), and at the international

[70] Gabriel P. Katoma, 'Legal Protection of Computer Programs' (1965) 47 *Journal of the Patent Office Society* 955, 956.

[71] Martin Goetz, ADAPSO Reunion Workshop, 'Intellectual Property' Computer History Museum, CHM Ref No. X4589.2008 (Recorded 4 May 2002), 5.

[72] Brief Amicus Curiae for Burroughs Corporation, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 4.

[73] From here program-focused histories often reach back to the Jacquard loom as providing the earliest antecedents. For example see Gerardo Con Diaz, 'Embodied Software: Patents and the History of Software development, 1946–1970' (July–September 2015) *IEEE Annals of the History of Computing* 8.

level: changes which gradually enmeshed the place of the computer program within intellectual property law. These included the 1970 *Patent Cooperation Treaty* (which allowed Member States to exclude computer programs from the examination process),[74] the 1973 *European Patent Convention* (which specifically excluded 'computer programs per se' from the scope of patentable subject matter), and a range of other efforts (such as the joint initiative of the National Bureau of Standards, the American Patent Law Association, and the Association of Computer Machinery Patent to classify computer programs to help with prior art searches).[75]

While this can be seen as a victory of sorts for hardware manufacturers, they were only partially successful. This was because while the computer program did operate (relatively) successfully as a boundary object to police the overlap between copyright and patents, it was much less successful in regulating patentable subject matter.[76] The reason for this, which is also a reason why patent law experienced so many problems in the 1960s and 1970s in dealing with computer-related subject matter, was that from a technological perspective the computer program and the programmed computer were inextricably connected and intertwined. While hardware and software companies had presented the choice of subject matter as a choice between non-patentable computer programs and patentable programmed machines, the fact that they were technologically intertwined meant that it was not easy to separate and distinguish them in this way.[77]

One of the consequences of this was that it was difficult to define a computer program in a way that did not bleed into and exclude other technologies that were considered to be patent eligible. (As we will see in Chapter 7, this is something that patentees exploited in drafting their patents to secure protection for their computer-related innovations.) These definitional problems led to the suggestion that any attempt to exclude computer programs from patentability was 'doomed to failure'. This was because any 'attempt to define software for the purposes of excluding it', such as in the *Patent Reform Act* of 1967, 'led to a definition that necessarily excluded other control devices or systems which, as machines or parts thereof, have always

[74] Rule 39(1), *Patent Cooperation Treaty*. For background see *Draft records of the Washington Diplomatic Conference in the Patent Cooperation Treaty: 1970 Conference Documents*, PCT/DC/3, (11 July 1969), item 32.

[75] G. Knight Jr, *Hierarchical Descriptor Classification System for Documents Related to Computer Software: With Scope Notes* (1970) (prepared for the Administrator, Office of Systems and Search Documentation, US Patent Office). Michael Duggan, 'Patents and Programs: The ACM's Position' (April 1971) 14(4) *Communications of the ACM* 278, 279. (Duggan was chairman of the ACM Committee on Copyrights, Patent and Trademarks).

[76] Thomas Haigh, 'Software in the 1960s as Concept, Service, and Product' (January–March 2002) *IEEE Annals of the History of Computing* 5.

[77] This ambiguity was captured in an article arguing for patent protection for algorithms which qualifies a statement that mathematical algorithms as such do not constitute patentable subject matter 'in theory' with a footnote that says that 'in fact a large number of patents are currently being obtained on what are essentially computer programming concepts'. Donald S. Chisum, 'The Patentability of Algorithms' (1986) *University of Pittsburgh Law Review* 959, 960–61, n 3.

been patentable',[78] such as 'built-in programs in special propose computers'[79] and 'programmable devices, such as an automatic dishwasher having certain predetermined cycles'.[80] The problem here was that while it may have been possible to demarcate and differentiate a computer program as an object of commerce, it was much more difficult to differentiate a program when the subject matter was seen from a more functional (or engineering) perspective.[81] In many ways these definitional concerns built upon the fact that using a computer program to run (or softwire) a computer was the engineering equivalent of hardwiring a computer. While there were physical differences, from an engineering or technological perspective the hardware and software forms of programming were functionally the same.[82]

One of the challenges that patent law faced when discussing patentable subject matter was that it had to deal with the fact that numerous patents had been granted for hardwired-programmed computers since the late 1940s.[83] The reason why this was important was because hardwiring was one of two ways by which special-purpose computers, that is computers programmed to perform specific tasks, could be constructed. Hardwiring was a permanent or semi-permanent solution that involved

[78] Morton C. Jacobs, 'Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyright and Patents' (1970) *Rutgers Journal of Computers and Law* 50, 58.

[79] B. M. Oliver, 'Major Recommendations of the US Presidents Patent Commission' (February 1967) *IEEE Spectrum* 57, 60.

[80] Statement by Philadelphia Patent Law Association, *Patent Law Revision: Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary, United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 1 (17, 18 May 1967), 259).

[81] Morton C. Jacobs, 'Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyright and Patents' (1970) *Rutgers Journal of Computers and Law* 50, 52. '[M]any, many patents issue which disclose the hardware embodiment by which contain claims broad enough to cover the software equivalent'. Richard E. Kurtz, 'Examples of Inventions Embodying Software, Types of Disclosure and Claims' in *Software Protection by Trade Secret, Contract, Patent: Law, Practice, and Forms* (Washington, DC: Patent Resources Group, 1969), 188.

[82] Brief Amicus Curiae for Applied Data Research, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 4. A machine containing a programmed control system is the same in all features as that containing special purpose hardware controls'. Statement of Richard C. Jones, President, Applied Data Research, *Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 2 (1–2 January, 1 February 1968), 751–52). If a patent application is filed disclosing only the claimed sequence of steps (in a flow chart for example), there is no conceivable way in which the Patent Office, or anyone else, can ascertain with certainty whether the applicant had in mind a computer program or a wired circuit. It is not surprising therefore that the Patent Office and certain patentees disagree as to whether or not a "computer program" has been patented'. 'A Case History: Benson and Talbot: Appellant's Position: Computer Programs in General', Appendix C, appended to Robert O. Nimtz, 'Computer Application and Claim Drafting under Current Law' in *Software Protection by Trade Secret, Contract, Patent: Law, Practice, and Forms* (Washington, DC: Patent Resources Group, 1969), 261.

[83] In *Ex Parte King and Barton* 146 USPQ 590 (1964) 'the Examiner took note of the engineering equivalence of hardware implemented inventions and those implemented by software and general-purpose hardware'. See Brief Amicus Curiae for the Association of Data Processing Service Organisations, Software Products and Service Section, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 16 n 38.

modifying the hardware of a computer to perform certain specified tasks. (A modern example would be a TV remote control.) The second way of developing a special-purpose computer, which was the technology that was under discussion in the 1960s and 1970s, was to use special (external) programs to control one of the standard general-purpose computers that were sold or leased by hardware manufacturers such as IBM.[84] Rather than adopting a different set of hardware connections each time a new purpose was desired, a computer engineer could use 'special software to achieve an equivalent softwire change in the connections and the general purpose hardware'.[85] That is, they could use different (soft-wired) programs to allow the computer to perform different tasks. In 'the place of the "hard-wire" of special purpose hardware, the software uses the "soft-wire" of recorded electrical signals which have the physical effect when placed in the general-purpose computer of setting thousands (or even millions) of electronic switches in unique combinations'.[86] 'Soft-wiring' was the term used by 'engineers in the industry to indicate that the recorded signal combinations of software achieve the same effects as actual "hard-wire", but the advantages of modifications and replacement without rewiring are also achieved'.[87]

The fact that a special-purpose computer could be created either by hardwiring or softwiring a general-purpose computer, combined with the fact that hardware manufacturers had been patenting hardwired computers since the 1940s, influenced the way hardware and software companies portrayed the subject matter in the 1960s and 1970s. On the one hand, hardware companies presented the 'new' technology in such a way that it allowed for the continued patenting of hardwired computers but, at the same time, excluded softwired computers operated by computer programs. In contrast, software companies presented the technology in such a way that allowed them to argue that softwired computers should be given the same type of protection as had been bestowed on hardwired computers. Specifically software companies argued that given that the decision to either hardwire or softwire a computer was based on economic and practical rather than engineering considerations that 'consequently there should be no legal difference since the two forms of the invention are engineering equivalents'.[88] To hold otherwise would have meant

[84] Morton C. Jacobs, 'Patents for Software Inventions: The Supreme Court's Decision' (1973) *Journal of the Patent Office Society* 59.

[85] Ibid., 60.

[86] See Brief Amicus Curiae for the Association of Data Processing Service Organisations, Software Products and Service Section, *Gottschalk* v. *Benson*, Supreme Court of the US, No. 71-485 (Oct. Term, 1971), 10.

[87] Ibid., 10 n 27. Computer engineers who recognize the equivalence of software and hardware, 'speak of the software techniques for the building of special-purpose computers as "softwiring."' Morton C. Jacobs, 'Patents for Software Inventions: The Supreme Court's Decision' (1973) *Journal of the Patent Office Society* 59.

[88] Statement of Richard C. Jones, President, Applied Data Research, *Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 2 (1–2 January, 1 February 1968), 751–2).

discriminating against 'inventors who chose a program as the preferred embodiment in favour of a hardware embodiment for the same inventive concept'.[89] If this was allowed it would, so the argument went, have created an unfair situation that arbitrarily favoured one segment of the computer industry over another.[90]

The upshot of this was that any attempt to exclude computer programs would have also excluded other 'control mechanisms; such as the 'circuitry embodiment of a machine invention' which was 'an engineering equivalent of the program embodiment'.[91] These problems were reinforced by the fact that despite the claims of the hardware companies, no one was really interested in patenting computer programs (algorithms, or mathematical methods) per se. As Martin Goetz said in 1970, '[a]t no time did' any of the software producers 'or any other advocate of "software patents" ever ask to protect computer programs. Rather, our goal is not to have the patentability of an "inventive machine process" denied solely because the inventor arbitrarily chose to embody that machine process in software (usually because of the prohibitive costs of embodying the same invention in hardware)'.[92] Robert Nimtz made a similar point when he said, the 'overall issue has never been the patentability of computer programs, as such. On the contrary, the issue has always centered around the patentability of processes carried out in response to programmed instructions in a computer, and to the patentability of apparatus configurations resulting from the execution of programmed instructions in a computer'.[93] Nimtz summed up these arguments when he said the 'program, as writing, has never been the subject matter of a patent claim, at least as far as this author is aware'.[94]

Since there was no real interest in patenting computer programs as ends in themselves and it was very difficult to define computer programs for the purpose of excluding them from protection in a way that did not also exclude subject matter

[89] Anon, 'Computer Programs: Are They Patentable?' (29 December 1968) *The New York Times* 1.

[90] 'Those who manufacture the programs but not the machines should have the same rights to patent protection as those who manufacture the machine'. Computer engineers who recognize the equivalence of software and hardware, 'speak of the software techniques for the building of special-purpose computers as "softwiring."' Morton C. Jacobs, 'Commissions Report (re: Computer Programs)' (1967) *Journal of the Patent Office Society* 372, 376.

[91] Ibid. The overlap was recognized in the comment, in relation to the suggestion that one of the reasons why computer programs should not be patentable was because of the problems facing the Patent Office that 'even if some defined area of computer programming technology were to be made "non-statutory", the Patent Office would still have the burden of classifying and searching the computer programming literature because of the close interplay between the software and hardware technologies'. George Metcalf, (US Chamber of Commerce), *Patent Law Revision, Subcommittee on patents, trademarks, and copyrights of the Committee on the Judiciary United States Senate* (Ninetieth Congress, First Session Pursuant to S Res 37 on S. 2, S. 1042, S. 1377, S. 1691, Part 2 (1–2 January, 1 February 1968), 454).

[92] Martin Goetz, 'A Different Viewpoint on the Benson Talbot Decision' (May 1973) 16 *Communications of the ACM* 334.

[93] Robert O. Nimtz, 'The Patentability of Computer Programs' (1970) 1 *Rutgers Journal of Computers and Law* 38.

[94] Ibid., 38 n 4.

considered to be patent-worthy, it became clear that the computer program could not operate as an effective way of policing computer-related subject matter. While the decision in Europe to use the 'computer program per se' as a way of regulating patentable subject matter meant that European patent law was forced to work out a way of distinguishing computer programs per se from computer-related inventions, patent law in the United States went in a different direction. As we will see in Chapter 7, after struggling to reconcile the 'contested ontologies of software' for over two decades, in the 1980s US patent law shifted its focus of attention to develop a more legal approach to computer-related subject matter.