

15 The Politics of Representation: Narratives of Automation in Twentieth-Century American Mathematics

Stephanie Dick

Abstract

This chapter explores narratives that informed two influential attempts to automate and consolidate mathematics in large computing systems during the second half of the twentieth century – the QED system and the MACSYMA system. These narratives were both political (aligning the automation of mathematics with certain cultural values) and epistemic (each laid out a vision of what mathematics entailed such that it could and should be automated). These narratives united political and epistemic considerations especially with regards to *representation*: how will mathematical objects and procedures be translated into computer languages and operations and encoded in memory? How much freedom or conformity will be required of those who use and build these systems? MACSYMA and QED represented opposite approaches to these questions: preserving pluralism with a heterogeneous modular design vs requiring that all mathematics be translated into one shared root logic. The narratives explored here shaped, explained and justified the representational choices made in each system and aligned them with specific political and epistemic projects.

If there is to be a bias, it is to be a bias towards universal agreement.

QED Manifesto

15.1 Introduction

Automation is all about representation and representation is always a political project. In order to hand off a given task to a computer, that task must first be reconceived and reformalized as something that a computer can do, translated into its languages, its formalisms, its operations, encoded in its

memory.¹ In service of those transformations, decisions have to be made about what is important, what will be lost in the translation, whose needs or goals will be prioritized. This chapter explores two influential attempts to automate and consolidate mathematics in the second half of the twentieth century – the QED system and the MACSYMA system – and the representational choices that constituted each: the languages of mathematics had to be translated into the languages and formalisms of computing; relatedly, mathematical procedures, like proof verification or algebraic simplification, had to be translated into computer-executable operations; and decisions had to be made about how best to formalize mathematics for automation, with what foundational logics, rules and premises.

MACSYMA and QED developers made very different representational choices and they used narratives to frame those choices. Marc Aidinoff has observed that historians often set out to unearth the ‘hidden politics’ of technological systems that are framed by their developers or users as value-neutral, objective, apolitical. He argues we should also ‘listen to people when they tell us what, and who, they prioritized’, we should attend to ‘the political, as it lies on the surface of technology, as actors directly described it’ (Aidinoff 2022). This chapter attempts to do just that by focusing on the narratives with which QED and MACSYMA were framed in order to make sense of the approaches to automation they represent, and the animating visions of mathematics and culture at work underneath.² These narratives were not just stories, extraneous and external to the systems. Nor were they post hoc, developed to explain choices that had already been made. They mapped directly onto and informed technical development and design decisions. They also mapped onto *practice* – the representational choices framed by these narratives corresponded with cognitive realities – how users would have to think about and do mathematics with these systems.³

¹ I use the term ‘reformatism’ to refer to the process of translating abstractions from one symbolic or material system to another – for example, the translation of logical relations from the symbol system in the pages of a logic text to encodings in computer memory (Dick 2014).

² Line Andersen (Chapter 19) explores the narrative qualities of formal demonstrations like mathematical proofs themselves. Here, I alternatively explore how seemingly ‘external’ narratives can *shape* and *direct* formalizing and reformatizing efforts within mathematics.

³ In introducing this volume (Chapter 1), Mary Morgan proposes that we ‘think of narrative as a “technology”, a ‘general-purpose technology’ at that, comparing it to steam power, electricity and computing. In fact, computers were not recognized as ‘universal’ or ‘general purpose’ in any obvious sense in their earliest decades of use. As we will see, narrative in fact played an outsized role in sounding out and establishing the limits and possibilities of computers, as well as explaining and justifying the decisions that researchers made while trying to make them useful in different domains.

As such, the narratives that framed each project were both *political* and *epistemic*.⁴ They were foundational myths that advocated for the consolidation and automation of existing mathematical knowledge so that the computer could take over certain elements of mathematical labour – from algebraic simplification to proof checking – and in so doing open up new possibilities for knowledge-making. Mathematicians in the future, it was proposed, would be able to see new things, solve new problems and ask new questions with automated repositories of what was already known in hand.⁵ Neither QED nor MACSYMA fulfilled their foundational myths, however. They were utopian narratives, at the intersection of political and epistemic imagination. Throughout the second half of the twentieth century, there was genuine uncertainty about what kind of tool the modern digital computer would turn out to be, what its epistemic and cultural limitations and possibilities were. The narratives explored here served to attribute meaning, possible futures and cultural values to mathematics as it would be made manifest in this new and undetermined technology.

15.2 Political Choices in Automation

The QED system, whose development began with an anonymously authored manifesto in 1994, was an attempt to combat the ‘tower of Babel’ its developers perceived in the automation of mathematics which had, throughout the 1970s and 1980s, involved a proliferation of ‘incompatible reasoning systems and symbolic computation systems’ that were inefficient, redundant, cacophonous, and that threatened mathematics’ traditional claim to universal truth (*QED Manifesto* 1994: 242). The *QED Manifesto* accordingly called for the translation of mathematics into a single formal and computational system, ‘that effectively represents all important mathematical knowledge and techniques’ and that conforms ‘to the highest standards of mathematical rigor, including the use of strict formality in the internal representation of knowledge and the use of mechanical methods to check proofs of the correctness of all entries in the

⁴ Certain aspects of the reforming ambitions presented here, particularly the emphasis on communal standards, and the values which research aspires to, also resonate with the motivations and goals of synthetic biologists as discussed in Dominic Berry’s chapter (Chapter 16).

⁵ Developers of the MACSYMA system, for example, proposed that the system would serve as a *laboratory* within which mathematical scientists would experiment, even with procedures and operations in mathematics they would not know how to execute by hand. Mathematics, usually characterized as inhering in a logico-deductive ‘style of reasoning’, following Ian Hacking, would instead become increasingly empirical and experimental, and users would develop knowledge *of the system* and its capabilities and behaviours, rather than of the underlying mathematics (Hacking 1992). See also Huss (Chapter 3), for another case discussing the new possibilities afforded by computation. Automation and its mathematical discontents was a theme present in Lorraine Daston’s narrative science public lecture ‘Annihilating Time: The Coup D’État and the Limits of Narrative’, given on 5 November 2019 at the London School of Economics: www.narrative-science.org/events-narrative-science-project-public-seminar-series.html.

system' (*QED Manifesto* 1994: 238). It was to be a 'monument', gathering together, verifying and unifying mathematics, the 'foremost creation of the human mind'. Writing in the wake of the Cold War, and amid the rise of American liberalism, the authors of the *Manifesto* proposed that the system would help 'overcome the degenerative effects of cultural relativism and nihilism' (*QED Manifesto* 1994: 239–240). They lamented the perceived loss of 'fundamental values' that the end of the Cold War and the rise of liberalism signalled and saw in mathematics a uniting and universalizing possibility.

QED would bring mathematics together by *making it all the same* – by formalizing it within one 'root logic', the same rules and foundations at work throughout. The *Manifesto* incorporated a narrative of 'Babel' and of the loss of shared cultural values in order to align their project with an ideological goal: they wanted to use the universality of mathematics in order to reinforce 'fundamental values', in the face of cultural difference. The home of the project was the Argonne National Laboratory (where some of the anonymous authors were based). This was an American government and military funded, Department of Energy hosted, effort to assert 'universal truth'. But their project highlights that 'the universality of mathematics' *is itself a construct*. QED would *make* mathematics universal, by demanding that different visions, approaches, logics and techniques be put into one formal and technological system. Anything that wasn't or couldn't be re-formalized in this way would be 'outside of mathematics', excluded from the centralized system, from the monument to truth. The corresponding commitment to shared fundamental cultural values is similarly normative – values will only be universal and shared when everyone has been convinced (or forced) to adopt them.

The authors of the *Manifesto* were right about Babel in mathematics automation. Since the early 1960s, there had been a proliferation of attempts to automate different parts of mathematics, and the resulting systems did not conform to shared formal or computational specifications. Some of the 'cacophony' resulted from the fact that system developers were building from scratch and without collaboration or communication with other system developers. Some differences were the result of direct competition between them. But some of the formal and representational pluralism was done by design, including in the second case to be explored in this chapter.

The MACSYMA system, developed at Massachusetts Institute of Technology (MIT) between the mid-1960s and the early 1980s during the Cold War, was among the most influential early computer algebra systems. It was designed with multiple representational schemes, multiple logics, *on purpose*, because the developers believed this would make it more useful to practising mathematicians and mathematical scientists. MACSYMA, too, was meant to be a centralized, consolidated, automated repository of existing mathematical techniques – a toolkit mathematicians could use in order to spare themselves the time and effort of learning and executing those techniques

for themselves. But MACSYMA developers believed that the best way to automate and consolidate mathematical knowledge was with as much heterogeneity and flexibility as possible. They wanted to bring mathematics together in pieces, stand-alone modules that each operated according to its own logic, its own internal design. This, they believed, would create a more accurate and more useful encoding of mathematical knowledge that would reflect and respect the pluralism of mathematical communities.

In an article explaining the representational choices one must make in the automation of mathematics, MACSYMA developers used political language. In a section called 'The Politics of Simplification', Joel Moses (a lead MACSYMA developer) described these choices in terms of how much freedom they afford the user, acknowledging that user freedom almost always adversely affects efficiency (Moses 1971). There are many different but equivalent ways that mathematical relations can be expressed, and mathematicians choose particular expressions because they are convenient to work with in a given context. But what is convenient for a mathematician on paper may not be efficient on the computer where very different constraints and economies, of memory and operations, are at stake.

For example, even simple addition can lead to trouble on the computer. Consider the sum of a series of numbers [1] $S = x_1 + \dots + x_n$. In computers, numbers are typically stored in memory using a fixed number of bits, and for 'real numbers', a format called floating-point is used to represent them. However, floating-point schemes struggle to represent both very large and very small numbers. As such, for the purposes of automation when very large numbers may be involved, it might be simpler to work in 'log space' where the computer stores and operates on the logs of numbers rather than the numbers themselves, because they require less memory. **Incidentally, the capacity to simplify problems by calculating in 'log space' is what made tables of logarithms so valuable in the nineteenth century before automatic calculators.**

Expression [2] $\log\left(\exp(x_1) + \exp(x_2) + \dots + \exp(x_n)\right)$ calculates the same value as [1], but works in log space, and so is often more efficient for computation. If you want to compute the log-space representation of the sum of x_1 to x_n , you can convert out of log space (by exponentiating), compute the sum of the regular representation of the numbers, and then take the log again, as in [2]. But, on the computer, it can be even more efficient to represent this expression as [3] $M + \log\left(\exp(x_1 - M) + \dots + \exp(x_n - M)\right)$ where $M = \max(x_1, \dots, x_n)$.

[2] and [3] are equivalent, but how could [3] possibly be more efficient than [2]? It has this extra term, M , added and subtracted throughout. [3] is called the 'log-sum-exp' trick and it is a way of computing the sum of a series of numbers in log space without having gigantic intermediate calculations that could

exhaust computer memory. While it complicates the *expression* by adding M , M simplifies the *computation* by ensuring that numbers are sufficiently small to be represented in available memory. But this way of looking at and working with sums may be counter-intuitive or difficult for a human user who may nonetheless be required to input expressions in this form or recognize and interpret them on the screen if sums have been implemented in this way in the system they are using. In this and so many other cases, what is easier and more efficient computationally may not be what is easiest for the mathematician.⁶

Typically, the more representational flexibility a user has, the more ‘under the hood’ processing needs to be implemented by developers to translate inputs into a form that the system was set up to manipulate. A ‘user-friendly’ system might allow a user to input simple expressions like [1] and, ‘under the hood’, the computer could convert them into the more computationally efficient forms in [2] or [3] before executing, and then convert back when displaying a result. But, these conversions also cost computing resources, so more rigid designs demand that the user become accustomed to working with, recognizing and generating computer-oriented representations themselves. This problem – how to implement and represent mathematical expressions and operations efficiently in memory, how users could input and work with mathematical expressions and operations, and how much work was needed to translate between the two – is a core problem for the automation of mathematics. These are the representational choices involved in any automation effort, and these are the choices MACSYMA developers framed through political narrative.

Moses surveyed the algebraic computing systems of the 1960s according to what he figured as the politics of their representational choices. There were the so-called ‘radical systems’ that could only ‘handle a single, well-defined class of expressions. [...] This means that the system stands ready to make a major change in the representation of an expression written by a user in order to get that expression into the internal canonical form’ (Moses 1971: 530). There was ‘the new left’, which ‘arose in response to some of the difficulties experienced

⁶ Throughout the history of mathematics, there is a related tension between aspirations fully to formalize mathematics, and a recognition of the convenience and productivity of working with informal, heuristic or more intuitive languages and representational systems. Formal systems can be cumbersome and tedious to work with, and actual mathematical practice tends not to adhere to strict formalization. Yet, without formalization, there is concern about the truth of mathematical conclusions and the foundations on which they rest (Livingston 1999). This tension was revisited in the context of computer automation (MacKenzie 2005) because computers require levels of formalization that may be unintuitive or difficult for human use. This tension also relates to Norton Wise’s comments on the distinction between formal and natural language narratives in his afterword (finale) (Chapter 22). In this case, natural language narratives used to make sense of and ascribe cultural meaning to formalisms.

with radical systems' and which operated like a radical system but with some alternative algorithmic simplification mechanisms. There were 'the liberals', equipped with 'very general representations of expressions', the 'conservatives', who 'claim that one cannot design simplification rules which will be best for all occasions. Therefore, conservative systems provide little automatic simplification capabilities. Rather, they provide machinery whereby a user can build his own simplifier and change it when necessary' (Moses 1971: 532). There were also 'catholic' systems that used 'more than one representation for expressions and have more than one approach to simplification. The catholic approach is that if one technique does not work, another might, and the user should be able to switch from one representation and its related simplification facilities to another with ease' (Moses 1971: 532). MACSYMA was a catholic system, incorporating elements of liberal, radical and conservative representational choices – 'The designers of catholic systems emphasize the ability to solve a wide range of problems. They would like to give a user the ease of working with a liberal system, the efficiency and power of a radical system, and the attention to context of a conservative system. The problem with a catholic system is its size' (Moses 1971: 532). MACSYMA, with its catholic design, reflected a narrative that highlighted *horizontal management* – the system's modules operated independently of one another – and *pluralism* – each module operated according to its own representational schemes and internal logic (Martin and Fateman 1971).

Any attempt to encode and automate mathematics requires an answer to a host of representational questions – how should mathematical objects be stored in computer memory? What will be included and what will be excluded? How should human practice be translated into computer operations? Whose needs and perspectives will be prioritized – the user or the developer? How and how much should these processes and representations be made visible to the user on a screen or printout? How must users formulate their problems and objects of interest such that they can be input to the system? QED and MACSYMA were designed with different answers to this set of representational questions, both framed with politico-epistemic narratives. QED embodied a vision of mathematics as a source of universal, shared truth and 'fundamental values' in the face of scorned 'cultural relativism'. MACSYMA instead embodied a commitment to pluralism and flexibility in both mathematics and culture. These narratives flag the cognitive freedom or discipline that accompanies different approaches to automation – they describe how users must discipline their relationship to mathematics and mathematical representation in order to use a system effectively. They imagine a different role for computers in the production of mathematical knowledge, and different 'styles of reasoning' to accompany them (Hacking 1992).

15.3 From Political Choices to System Building

But how (and how well) do these narratives relate to on-the-ground realities of these projects? How free are the developers of technological systems to decide what their politics will be? What is highlighted and what is left out in these narratives? Jonnie Penn, a historian of artificial intelligence (AI) has demonstrated that, in spite of all of their self-proclaimed differences, early AI practitioners were in fact united by key underlying logics and values (Penn 2020). While they disagreed about how intelligence might be manifested in the machine, or what intelligence was, different approaches to AI were nonetheless united by many shared commitments – most notably, he identifies military and industrial logics and funding at work across them. For all their purported differences, they in fact agreed as much as they disagreed, especially about unspoken assumptions. Similarly, on the face of it, QED and MACSYMA embodied opposite approaches to the same problem – both projects aimed to centralize and automate mathematics, MACSYMA by preserving difference and adopting representational flexibility, QED by translating all of mathematics into one ‘root logic’ by unifying it. The narratives adopted by the developers of each system correspond to these opposing visions of automation. However, in spite of those differences, both systems shared a more fundamental belief that the consolidation and automation of mathematics was possible. They shared an underlying goal – to extract mathematical knowledge from people and communities and put it into the machine. To do so, both projects had to accommodate computers, whose limitations and possibilities constrained the epistemological and political values they could realize. The next sections offer a closer look at each automated system, the narratives that surrounded them and the practices that accompanied them.

15.3.1 *MACSYMA*

The MACSYMA system (for Project MAC Symbolic Manipulator) was developed under the auspices of Project MAC at MIT, beginning in the 1960s. The system was meant to offer automated versions of much of what mathematicians know and do: ‘The system would know and be able to apply all of the straightforward techniques of mathematical analysis. In addition, it would be a storehouse of the knowledge accumulated about many specific problem areas’ (Martin and Fateman 1971: 59). The system could multiply matrices, it could integrate, it could factor and simplify algebraic expressions, it could maximize and minimize functions and hundreds of other numeric and non-numeric operations. This automated repository of knowledge was meant to free mathematical scientists from ‘routine mathematical chores’, and free them even from the process of acquiring much mathematical knowledge for

themselves (Engelman 1965: 413). With such a system at hand, one need only to know when different operations were useful in solving a particular problem, but not necessarily how to execute those operations by hand oneself. The system grew in popularity, especially among Defense Advanced Research Projects Agency (DARPA)-funded military, academic and industrial research centres throughout the 1960s and 1970s. The PDP-10 computer at MIT on which the system was housed could be accessed through the ARPANET and was, Moses recalled, one of the most popular nodes during the 1970s (Moses 2012: 4). MACSYMA grew popular enough, in fact, that by the mid-1970s, they shifted to a user consortium funding model rather than relying on DARPA funding alone. The initial consortium included the Department of Energy, NASA, the US Navy and Schlumberger, an oil and gas exploration company.⁷ Universities and academic research labs continued to access the system freely until the early 1980s, when the system outgrew the development and maintenance capacities of the MIT team, and it was privatized (controversially) and licensed to Symbolics Inc.

MACSYMA was developed in explicit opposition to two other trends in artificial intelligence and automated mathematics research at the time, and these differences help to situate the developers' framing narratives. First, MACSYMA developers were critical of the 'symbolic' approach to AI which was largely characterized by an 'information processing' model of human intelligence in which minds took information as input and manipulated it according to a set of rules, and then output decisions, solutions, judgements, chess moves and other 'intelligent behavior' (Cordeschi 2002).

Following Allen Newell and Herbert Simon, AI researchers using this approach looked for the information-processing rules that governed different problem domains and set out to automate these. Newell and Simon's ultimate goal in this field was the development of a 'general problem solver' (GPS) – a computer program equipped with sufficiently general rules of reasoning that it could solve problems in any domain, by applying those rules in a top-down fashion to whatever symbolic input it was given (Newell, Shaw and Simon 1959). GPS was based on a 'theory of problem solving' that suggested 'very general systems of heuristics [...] that allows them to be applied to varying subject matters' (Newell, Shaw and Simon 1959: 2). The idea was that people do the same sorts of analysis and planning when they solve problems in chess, or in mathematics, or in governance alike, and that if you could identify and automate those 'heuristics', they could be successfully applied 'to deal with different subjects' (Newell, Shaw and Simon 1959: 6). Attempts to produce a general problem solver in this way, however, were fraught with failure and overpromise throughout the second half of the century.

⁷ MIT Archives, Collection AC268, Boxes 22–24.

According to Moses, these failures were entirely unsurprising. He rejected both the belief that some one set of reasoning rules or heuristics was sufficient for problem-solving across domains, and the underlying vision of ‘top down’ control in automation. Reflecting in 2012, he wrote:

[. . .] I was increasingly concerned over the classic approach to AI in the 1950s, namely heuristic search, a top-down tree-structured approach to problem solving [. . .] There was Herb Simon [. . .] emphasizing a top-down hierarchical approach to organization. I could not understand why Americans were so enamored with what I considered an approach that would fail when systems became larger, more complex, and in need of greater flexibility. (Moses 2012 : 129)

Moses thought it was untenable to identify any set of top-down rules that would be effective in solving problems across domains in mathematics. He also believed that this was an inaccurate picture of how human minds work. He believed minds were modular as well, applying different tricks and methods here and there. He did not believe that there was a singular governing set of reasoning principles at work across all intelligent behaviour, not even in mathematics. The MACSYMA system was accordingly modular – one module to factor, another module to integrate, another module to find the Taylor expansion – and these modules did not operate according to a shared set of rules or a top-down governing principle. It fell to the user to chart a path through the available modules that would produce a solution to their problem, and this was based on experiment, intuition, trial and error.

Moses was born in Palestine in 1941 and found America to be more culturally homogeneous by comparison. He suggested that this cultural homogeneity explained the commitment to top-down hierarchical organizational structures, citing these as uniquely *American*. He believed that pluralist systems of organization had correlates both in other societies and in the branches of mathematics, and sought to reflect these in MACSYMA:

When I began reading the literature on Japanese management, I recognized ideas that I had used in [. . .] MACSYMA. There was an emphasis on abstraction and layered organizations as well as flexibility. These notions are present in abstract algebra. In particular, a hierarchy of field extension, called a tower in algebra, is a layered system. Such hierarchies are extremely flexible since one can have an infinite number of alternatives for the coefficients that arise in each lower layer. But why were such notions manifest in some societies and not so much in Anglo-Saxon countries? My answer is that these notions are closely related to the national culture, and countries where there are multiple dominant religions (e.g., China, Germany, India, and Japan) would tend to be more flexible than ones where there is one dominant religion. (Moses 2012)

Moses’ interest in ‘non-American’ forms of organization informed his approach to automation and AI throughout his career. His critique of top-down control infrastructure was not just that, empirically, it was brittle and

performed poorly, but also that it reproduced a commitment to homogeneity that he believed was characteristically American.

Moses recognized what historians of technology have long suggested – that culture and ideology can be reproduced in technical infrastructure – and the MACSYMA system was designed to reflect the political-technics of pluralistic places. MACSYMA's catholic modularity was intended to preserve pluralism, to allow for context, mixing radical, liberal and conservative elements. That modularity would, he believed, better meet the needs of mathematicians, avoid the brittleness and failings of top-down control hierarchies he perceived in other automation attempts and, he considered, in American culture overall.

15.3.2 QED

Where Moses sought to preserve pluralism in MACSYMA, the QED system, inaugurated in the 1990s, was meant to promote and even enshrine cultural homogeneity:

[P]erhaps the foremost motivation for the QED project is cultural. Mathematics is arguably the foremost creation of the human mind. The QED system will be an object of significant cultural character, demonstrably and physically expressing the staggering depth and power of mathematics. Like the great pyramids, the effort required may be great, but the rewards can be even more staggering than this effort. Mathematics is one of the most basic things that unites all people, and helps illuminate some of the most fundamental truths of nature, even of being itself. In the last one hundred years, many traditional cultural values of our civilization have taken a severe beating, and the advance of science has received no small blame for this beating. The QED system will provide a beautiful and compelling monument to the fundamental reality of truth. It will thus provide some antidote to the degenerative effects of cultural relativism and nihilism. (*QED Manifesto* 1994: 239–240)

The *QED Manifesto* was written by a collective of automated mathematics researchers, and anonymously published in the proceedings of the 1994 *Conference on Automated Deduction*, after the fashion of the mathematical collective called Nicholas Bourbaki.⁸ Like Bourbaki, however, the *Manifesto* had a primary author – Robert Boyer, a professor of computer science, mathematics and philosophy at the University of Texas at Austin. Boyer had many collaborators Argonne, the institutional home of QED, which had also been an important site of automated mathematics research since the 1960s. Readers of the 1994 *Manifesto* were directed to email ‘subscribe qed’ to majordomo@msc.anl.gov in order to subscribe to the Argonne-supported qed@msc.anl.gov mailing

⁸ ‘Nicholas Bourbaki’ was a pseudonym used by a group of primarily French mathematicians in the 1930s who collectively authored several texts aimed at modernizing mathematics through an emphasis on structure and abstraction (Corry 1998).

list. Argonne also hosted the first QED workshop, aimed at realizing the imagined project later in 1994.

Further reading of the *Manifesto* reveals which ‘civilization’ and whose values were perceived as under threat and in need of monumentalizing: they worked in the tradition of the European Enlightenment. The authors of the manifesto lamented the fact that ‘the increase of mathematical knowledge during the last two hundred years has made the knowledge, let alone understanding of all, or even the most important, mathematical results something beyond the capacity of any human’ (*QED Manifesto* 1994). In the late nineteenth century, during the so-called ‘foundations crisis’, similar concerns motivated efforts to consolidate and formalize mathematics, but in books and periodicals rather than computer systems (Corry 1998; Gray 2004). Logicians and philosophers like Giuseppe Peano, Gottlob Frege, Bertrand Russell and Alfred North Whitehead set out to develop logics whose premises and inference rules they hoped would be sufficient for the establishment of mathematical results from different fields, and they published lists of known theorems and proofs of foundational results within those systems. Their desire to consolidate emerged in part in response to concerns about the foundations of mathematics and the discovery of troubling paradoxes, but also in response to the professionalization and proliferation of mathematics, which developed distinct national cultures and schools during the nineteenth century.

If mathematics was to be the bedrock of ‘universal truth’, it wouldn’t do for it to diversify, proliferate and divide in this way, threatening the Enlightenment narrative in which mathematics, and its nineteenth- and twentieth-century bedfellows *reason* and *rationality*, respectively, were the foundations for universal truth.⁹ The *Manifesto* cites Aristotle on this point:

In the end, we take some things as inherently valuable in themselves. We believe that the construction, use, and even contemplation of the QED system will be one of these, over and against the practical values of such a system. In support of this line of thought, let us cite Aristotle, the Philosopher, the Father of Logic: That which is proper to each thing is by nature best and more pleasant for each thing; for man, therefore, the life according to reason is best and pleasantest, since reason more than anything is man. (*QED Manifesto* 1994: 240)

⁹ The late nineteenth-century anxiety about mathematics has been called the ‘Foundations Crisis’ (Corry 1998; Gray 2004). In the European Enlightenment context, ‘reason’ was cast as a universal faculty equipped to produce objective knowledge and seek out truth, and yet simultaneously it was denied to colonized people, people of colour and women. Reason was deemed ‘compromised’ in Eastern thinkers – it was a ‘universal’ faculty that European men reserved only for themselves (Terrall 1999; Mazzotti 2012). Mathematics had a central role to play in this history, since it was associated with ‘reasoning’ itself, especially in the American Cold War context (Erickson et al. 2013; Phillips 2014).

The narrative that an antidote to cultural relativism was required, in the form of a monument to fundamental truth, participated in that century-old impulse to gather together and render immutable – by logic and consolidation – what is known in mathematics. The enlightenment commitment to ‘reason’ as the bedrock of truth, as an imagined ‘universal’ faculty, and of mathematics as its purest manifestation, were the values perceived as under threat by ‘cultural relativism’ and in need of reinforcement by QED. The commitment to reason, like the commitment to formalization, may seem in tension or at odds with the use of narrative tools, and yet, in the context of QED, they work in entangled ways. While acknowledging that there would be biases and disagreements in the implementation of the system, their belief in universalism was not swayed – ‘If there is to be a bias, let it be a bias towards universal agreement’ (*QED Manifesto* 1994: 241). This statement captures the tension and political fantasy that supported the project.

The late nineteenth- and early twentieth-century attempt to consolidate and fully formalize all of mathematics largely failed. While significant subsections of mathematics were subjected to successful axiomatization efforts, much of mathematics remained and remains unformalized. There were also the incompleteness and decision problem results of Kurt Gödel, Alonzo Church, and Alan Turing, which demonstrated that formalization has intrinsic limitations. There was similarly the fact that most formal systems were too obtuse for actual use in practice, and most research mathematicians did not work strictly within them.

Boyer and his co-authors on the *Manifesto* believed that the modern digital computer put the full formalization of mathematics back on the table. *Human* limitations had impeded earlier efforts, but these were limitations that the computer did not share – ‘the advance of computing technology [has] provided the means for building a computing system that represents all important mathematical knowledge in an entirely rigorous and mechanically usable fashion’ (*QED Manifesto* 1994). Where early twentieth-century efforts at consolidation and formalization had fallen short, computer automation, they believed, could succeed – ‘The QED system we imagine will provide a means by which mathematicians and scientists can scan the entirety of mathematical knowledge for relevant results’. Mathematical knowledge would be redefined as that which was included in the system, and which adhered to its formal prescriptions, highlighting again that the field’s ‘universality’ was *constructed* through inclusionary and exclusionary choices. Mathematicians would not need, they went on, ‘minute comprehension of the details’ of the knowledge they would find, use and build upon in the centralized database. In this way, human understanding of that knowledge was displaced in favour of machine-consolidation. Human understanding was further displaced by the

QED commitment to machine-verification. Results would be accepted, not if they were convincing to mathematicians, but if they were automatically verifiable by the system.

QED, as earlier projects projecting universalism in mathematics, largely failed to achieve its lofty goals. Although it led to the development of the Mizar library which currently holds the largest database of fully formalized and verified mathematical results, and projects are ongoing, no system has achieved the consolidation and automation they imagined.¹⁰ The Manifesto itself pointed to numerous obstacles – ‘social, psychological, political, and economic’, not to mention technical and mathematical – that would need to be overcome (*QED Manifesto* 1994: 250). They imagined a vast number of people would be needed to achieve this project and suggested that credentialing systems and individualism in mathematics might also impede their vision (*QED Manifesto* 1994: 249). They noted even that QED should avoid ‘any authorship or institutional affiliation’ since these could undermine the universalism that QED sought to construct. Universalism would be the product of a particular social and labour organization, central planning, shifts in credentialing and motivations, as well as technical consolidation.

The *Manifesto* acknowledged that the establishment of leadership, and the cultivation of agreement about the priorities and plans that would guide the project, would be difficult. What they described, essentially, was a centrally planned economy – you need a central planner to make a centrally planned universal mathematics, to ‘establish some “milestones” or some priority list of objectives’, to ‘outline which parts of mathematics should be added to the system and in what order. Simultaneously, an analysis of what sorts of cooperation and resources would be necessary to achieve the earlier goals should be performed’ (*QED Manifesto* 1994: 249). The *Manifesto* proposed that, ideally, the ‘root logic’ with which mathematics would be represented in the system would be widely accepted: ‘It is crucial that the “root logic” be a logic that is agreeable to all practicing mathematicians’ (*QED Manifesto* 1994). However, they also acknowledged that no such ‘root logic’ was, as yet, universally accepted, and leadership and agreement would remain difficult. In practice, the QED project was guided by the perspectives of a small number of automated reasoning researchers and descendent efforts remain adjacent to both mainstream mathematics and computer science. In spite of continually running up against the realities of pluralism and individualism in mathematics, part of QED’s foundational myth was that a ‘root logic’ could be established, that reasonable people would no doubt agree on it, and mathematical labour could be reorganized accordingly. The *Manifesto*’s acknowledgement of obstacles highlighted the fact that the unity and universalism of mathematics would have

¹⁰ The Mizar Project: <http://mizar.org/>.

to be *constructed* – disagreements erased, a ‘root logic’ selected and then all of mathematics reformatized and implemented within it by labourers willing to eschew individual recognition for collaborative achievement. Although QED inspired significant efforts in this direction, no such fully formal, automatically verified, comprehensive consolidation of mathematics yet exists.

In spite of consistent failures, the *belief* that full formalization and consolidation of mathematics could be achieved, *just around the next corner*, with the next advancement, has been remarkably powerful and persistent in the history of mathematics. The authors of the *QED Manifesto* suggested that paper, pencil and human minds had simply been too limited for the task but the technological advances of computing had, by the mid-1990s, made it possible to achieve. Over the next several decades, mathematicians reflecting on the QED project proposed that it had failed because of limited interest and limited technical capacity but that *now* it might be possible. In 2007, Freek Wiedijk asked, ‘Why the QED manifesto has not been a success (yet)’, and concluded that ‘I myself certainly believe that the QED system will come. If we do not blow up the world to a state that mathematics will not matter much anymore, then at some point in the future people will formalize most of their proofs routinely in the computer. And I expect that it will happen earlier than we now expect’ (Wiedijk 2007: 132). In 2016, success still had not come, but Italian computer scientists Michael Kohlhase and Florian Rabe proposed that ‘Even though [QED] never led to the concrete system, communal resource, or even joint research envisioned in the QED manifesto, the idea lives on and shapes the research agendas of a significant part of the community’ (Kohlhase and Rabe 2016). Again, in 2014, Ittay Weiss proposed that ‘two decades later it is safe to say the dream is not yet a reality’. But he, too, believed that success was just around the corner (Weiss 2014: 803). Weiss suggested a new approach to the complete automation of mathematics, which he named ‘Mathropolis’ – an imagined polity, *just over the next hill*, in which the monument to universal truth will be built, the pluralism of mathematics united in one formal system, the economy of mathematical labour centrally planned, the limited human mind and social vetting of truth replaced by the robust and reliable machine. His proposed system, named as a city, reflected the entanglement of politics, governance and epistemology at work within the QED project.

15.4 Conclusion

This vision – that mathematics will be fully consolidated, automated and formalized just around the next social or technical corner, that its universality will be made materially manifest – gained much traction in the late nineteenth and early twentieth centuries. Responding both to the discovery of several troubling paradoxes and to the proliferation of mathematical fields and centres of research, mathematicians around the turn of the twentieth century wanted to get all of

mathematics into one place, they wanted to represent it all in the same formal system, the same symbolism and in the pages of one book. They were unable to do so, for formal, social and material reasons. With the perceived possibilities of modern digital computing in the 1960s and 1970s, many, including the developers of the MACSYMA system, believed that, *finally*, consolidation would be possible, especially through pluralism and horizontal management. It wasn't. Again, in the 1990s, the anonymous authors of the *QED Manifesto* proposed that *finally* the cost of computing and the intellectual will were such that it would be possible to gather up all of mathematics in one place, in one formal system. It wasn't. In revisiting the QED Manifesto two decades later, several mathematicians proposed that the time had *finally* come for the full and final consolidation of mathematics. It hadn't. This story – that mathematics will be fully unified, consolidated, formalized *just around the corner* – now that the conditions of past failures have been overcome – shapes whole research projects, and scaffolds belief in the universalism of mathematics.¹¹

In spite of their different approaches to automation, and the different narratives that accompanied them, QED and MACSYMA both participated in that shared goal of consolidating mathematical knowledge and automating it, putting it in the machine. Moreover, both received initial funding from the same organizations – DARPA and the Office of Naval Research (ONR), especially. Both projects were undertaken at powerful hubs of military–industrial–academic research, MIT and Argonne National Laboratory, whose power grew out of the post-war American context. Both ascribed to ideologies of efficiency and logics of industrial planning in their imagining of automated mathematics, but to serve two different ideologies. Both projects rested on the belief that, whether pluralistically or not, knowledge could be extracted from human knowers, that it could and should be 'put into the machine'. And both set out to redefine, transform and encode mathematical knowledge with computer-oriented representations and processes. QED and MACSYMA have more in common than their framing narratives may suggest.

MACSYMA was meant to preserve pluralism and empower mathematicians for new programs of problem-solving. It was meant to free time and energy for new questions and explorations by handing over much mathematical labour to the machine. However, the freedom afforded by MACSYMA required users to work with and within highly disciplined and often counter-intuitive computer-oriented representational schemes, and that freedom cultivated dependency on the system, once a user came to rely on the system for the execution of techniques they did not themselves understand (Dick 2020). The developers conceded the

¹¹ Further chapters in this volume making use of the scaffold notion include Kranke (Chapter 10), Teather (Chapter 6) and Miyake (Chapter 5). On examples of background knowledge active in cases, see Crasnow (Chapter 11).

point that MACSYMA required mathematicians to reconceive what they know for the purpose of automation, and even encouraged users to transform their own knowledge into automated modules for inclusion in the system. The modularity that was meant to serve a pluralistic and modular vision of mathematical practice also made it easier for mathematicians to take what they knew and ‘put it in the machine’. Users could contribute to a SHARE Directory – an ever-growing repository of new modules, user-generated, that expanded the system’s capabilities and made more ‘knowledge’ available to more people. The claim that MACSYMA freed mathematicians and that it preserved pluralism of practice betrayed the fact that incredible accommodation to the machine was first required and that the system was primarily useful and usable to elite and defence-funded institutions. When MACSYMA was privatized in 1981, and licensed to Symbolics Inc., the users who had worked so hard to learn and accommodate and even contribute to the system were then transformed into a set of buyers in a market who had to now pay for the privilege of consuming the goods they had in part made themselves. MACSYMA wasn’t the materialization of freedom and pluralism that its narrative suggests.

Lewis Mumford cautioned, in opposition to strong theories of social construction, that there are technological systems that cannot be aligned with any politics whatever, but rather operate according to fundamental logics that cannot be overcome through creative use, alternative intention or new narrative. Mumford suggested that computers are essentially authoritarian technics, centralized command and control technologies, no matter how often people have tried to align them with democracy, freedom, counter-culture and pluralism (Mumford 1964; Turner 2008). Even if one doesn’t accept Mumford’s analysis in its entirety, it would still be safe to suggest that no American militarily funded effort to extract knowledge from knowers and communities and make it efficiently and automatically available to defence-funded research institutions, can be aligned with the politics of pluralism.

Both QED and MACSYMA were supposed to serve a dual purpose. First, both were meant to automate mathematics, and in this they differed – the former meant to automate by representing all of mathematics in a shared ‘root logic’, the latter, automating mathematics modularly, attempting to preserve logical and methodological pluralism, as well as offer users flexibility. In this difference, the narratives the developers attached to the projects suit. But both projects were also meant to consolidate all of mathematical knowledge, efficiently and automatedly. Both entailed and in fact *celebrated* the displacement of human understanding – users need not understand that which the system can do. For MACYSMA, users would be spared the need to learn mathematical techniques for themselves because of having an automated system available to execute them instead. In QED, the fundamentally social project of establishing mathematical truth was displaced in favour of

automatically verified results. Both entailed theories of knowledge that did not require a subjective knower, only a machine encoding. And in this regard both displace human understanding, social processes and the pluralism these entail. And both projects consolidated resources and decision-making power, as well as the automated mathematical knowledge itself, in the hands of a small number of institutions, also limiting pluralism. Both projects also minimize the productive capacity of friction, miscommunication, disagreement, misunderstanding and difference. While MACSYMA preserved *logical* pluralism in its modularity, all modules still had to accommodate the constraints of a single arbiter: the PDP-10 computer on which they ran. We might call this *computational-pluralism*, and it was only as plural as those constraints permit. The politics of technology go beyond the technical design choices made within them to include the context in which they are developed, who pays for them, profits from them, and how much freedom or discipline users and contributors have in their engagement with technical systems.

In these histories of mathematics automation, narratives map onto design and implementation decisions, they acknowledge the representational choices involved in accommodating the machine and the user and they reflect beliefs about mathematics' relationship to culture. But the narratives that developers use to frame their technological systems may also serve to direct our gaze away from certain institutional realities and unspoken assumptions. These epistemic–political narratives highlight entanglements between mathematics and culture, and conformity and freedom, in the representational choices that automation always involves.¹²

References

- Aidinoff, M. (2022). 'Centrists against the Center: The "Jeffersonian" Internet as Policy and Politics'. In J. Abbate and S. Dick, eds. *Abstractions and Embodiments: New Histories of Computing and Society*. Baltimore, MD: Johns Hopkins University Press.
- Cordeschi, R. (2002). *The Discovery of the Artificial: Behavior, Mind, and Machines Before and Beyond Cybernetics*. New York: Springer.
- Corry, L. (1998). 'The Origins of Eternal Truth in Modern Mathematics: Hilbert to Bourbaki and Beyond'. *Science in Context* 12: 137–193.
- Dick, S. A. (2014). 'After Math: (Re)configuring Minds, Computers, and Proof in the Postwar United States'. PhD thesis, Harvard University.
- (2015). 'Of Models and Machines: Implementing Bounded Rationality'. *Isis* 106.3: 623–634.
- (2020). 'Coded Conduct: Making MACSYMA Users and the Automation of Mathematics'. *BJHS Themes* 5: 205–224.

¹² *Narrative Science* book: This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 694732). www.narrative-science.org/.

- Engelman, C. (1965). 'MATHLAB: A Program for On-Line Machine Assistance in Symbolic Computations'. *AFIPS (American Federation of Information Processing Societies) Conference Proceedings*. vol. 27. Fall Joint Computer Conference. London: Macmillan: 413–423.
- Erickson, P., J. L. Klein, L. Daston, R. Lemov et al. (2013). *How Reason Almost Lost Its Mind: The Strange Career of Cold War Rationality*. Chicago: University of Chicago Press.
- Gray, J. (2004). 'Anxiety and Abstraction in Nineteenth-Century Mathematics'. *Science in Context* 17.2: 23–47.
- Hacking, I. (1992). "'Style" for Historians and Philosophers'. *Studies in the History and Philosophy of Science A* 23.1: 1–20.
- Kohlhase, M., and F. Rabe (2016). 'QED Reloaded: Towards a Pluralistic Formal Library of Mathematical Knowledge'. *Journal of Formalized Reasoning* 9.1: 201–234.
- Livingston, E. (1999). 'Cultures of Proving'. *Social Studies of Science* 26.6: 867–888.
- MacKenzie, D. 2005. 'Computing and the Cultures of Proving'. *Philosophical Transactions of the Royal Society A* 363: 2335–2350.
- Martin, W. A., and R. J. Fateman (1971). 'The MACSYMA System'. In *Proceedings of the Second ACM Symposium on Symbolic and Algebraic Manipulation*. New York: Association for Computing Machinery: 59–75.
- Mazzotti, M. (2012). *The World of Maria Agnesi, Mathematician of God*. Baltimore, MD: Johns Hopkins University Press.
- Moses, J. (1971). 'Algebraic Simplification: A Guide for the Perplexed'. *Communications of the ACM* 14.8: 527–537.
- (2012). 'MACSYMA: A Personal History'. *Journal of Symbolic Computation* 47.2: 123–130.
- Mumford, L. (1964). 'Authoritarian and Democratic Technics'. *Technology and Culture* 5.1: 1–8.
- Newell, A., J. C. Shaw and H. Simon (1959). 'Report on a General Problem-Solving Program'. Revised version. *RAND Technical Report P-1584*. typescript. http://bit.savers.informatik.uni-stuttgart.de/pdf/rand/ipl/P-1584_Report_On_A_General_Problem-Solving_Program_Feb59.pdf.
- Penn, J. (2020). 'Inventing Intelligence: On the History of Complex Information Processing and Artificial Intelligence in the United States in the Mid-Twentieth Century', PhD thesis, University of Cambridge. www.repository.cam.ac.uk/handle/1810/315976.
- Phillips, C. (2014). *The New Math: A Political History*. Chicago: University of Chicago Press.
- QED Manifesto* (1994). In 'Automated Deduction: CADE 12'. *Lecture Notes in Artificial Intelligence* 814: 238–251.
- Terrall, M. (1999). 'Metaphysics, Mathematics, and the Gendering of Science in Eighteenth Century France'. In W. Clark, J. Golinski and S. Schaffer, eds. *The Sciences in Enlightened Europe*. Chicago: University of Chicago Press, 246–271.
- Turner, F. (2008). *From Counterculture to Cyberculture: Steward Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago: University of Chicago Press.
- Weiss, I. (2014). 'The QED Manifesto after Two Decades: Version 2.0.' *Journal of Software* 11.8: 803–815.
- Wiedijk, F. (2007). 'The QED Manifesto Revisited'. *Studies in Logic, Grammar, and Rhetoric* 10.23: 121–133.