The Work of Proof in the Age of Human–Machine Collaboration

By Stephanie Dick*

ABSTRACT

During the 1970s and 1980s, a team of Automated Theorem Proving researchers at the Argonne National Laboratory near Chicago developed the Automated Reasoning Assistant, or AURA, to assist human users in the search for mathematical proofs. The resulting hybrid humans + AURA system developed the capacity to make novel contributions to pure mathematics by very untraditional means. This essay traces how these unconventional contributions were made and made possible through negotiations between the humans and the AURA at Argonne and the transformation in mathematical intuition they produced. At play in these negotiations were experimental practices, nonhumans, and nonmathematical modes of knowing. This story invites an earnest engagement between historians of mathematics and scholars in the history of science and science studies interested in experimental practice, material culture, and the roles of nonhumans in knowledge making.

INTRODUCTION: THE NEGOTIATIONS BEGIN

In 1950, Alan Turing predicted “that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.”¹ It is difficult to ascertain whether Turing’s prediction was correct, given that what it means for a computer to think has been up for debate. It is not often remarked, however, that human thinking has also been reconfigured through interactions with computing machinery. This essay traces one such transformation in mathematical thought that occurred in the field of Automated Theorem Proving (ATP).

ATP was a field of computation research aimed at producing logical and mathematical proofs by automated means. The field emerged in the late 1950s in the United States

* Department of the History of Science, Science Center 371, Harvard University, Cambridge, Massachusetts 02138.
This research was supported in part by a Social Sciences and Humanities Research Council of Canada doctoral award.

* Isis*, 2011, 102:494–505
©2011 by The History of Science Society. All rights reserved.
0021-1753/2011/10203-0006$10.00

494
following the advent of the reprogrammable digital computer. This essay focuses on one ATP engine—the Automated Reasoning Assistant (AURA)—and the transformation in mathematical thought and practice that materialized around it. AURA—implemented and employed during the late 1970s and 1980s by a team at the Argonne National Laboratory—was designed according to a specific understanding of human theorem-proving work: namely, that mathematical thought and discovery involve certain irreducible moments of insight and intuition that cannot be automated. This vision originated with Larry Wos, a pioneer of ATP at Argonne, and expanded to encompass most ATP work at the lab. In accordance with this vision of human mathematical thought, the Argonne team eschewed the project of total automation and instead aimed to implement a reasoning assistant that could collaborate with human users. They would provide their unautomatable insights and intuitions about a proof search, and AURA would perform the lesser task of running the search within the frame of those contributions. In practice, however, the work of the humans and the work of the AURA were far more entangled than this model of labor division might suggest. AURA was full of surprises. Although the program could only do precisely what it was programmed to do, the programmers usually could not know in advance what the consequences of their instructions would be. Indeed, if they could, they would have had little need for a fast computer to do this work for them. They manually studied printouts of thousands of clauses from run after run on AURA in order to understand what was going on when the engine looked for proofs. To accommodate their a posteriori revelations, the Argonne group committed to an experimental paradigm for ATP in which AURA was constantly improved and redesigned on the basis of results from previous runs. Those prized human insights and intuitions that the Argonne researchers reserved for the

2 ATP was in the spirit of Charles Babbage and George Boole, who shared the conviction that human thought, and especially mathematical thought, could be automated. A more specific call to arms for ATP research came from John McCarthy, who coined the term “Artificial Intelligence” in 1955. In 1958 he proposed that researchers attempt to develop automated theorem-proving programs. See Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence (Natick, Mass.: Peters, 2004), p. 51. ATP research was not limited to traditional academic centers. Some important early engines were developed at the RAND Corporation, the Argonne National Laboratory, and IBM research labs. Early ATP research sites, engines, and researchers are well documented in Donald Loveland, “Automated Theorem Proving: A Quarter Century Review,” in Automated Theorem Proving: After Twenty-five Years, ed. W. W. Bledsoe and Loveland (Contemporary Mathematics, 29) (Providence, R.I.: American Mathematical Society, 1984), pp. 1–46.

3 Some members of the team, among them Steve Winker and Ross Overbeek, were initially employed at Northern Illinois University. According to Overbeek, a quarrel with the university administration in the early 1980s regarding the trajectory of their then rapidly growing Department of Computer Science provoked his (and many others’) departure. Until that time, the Northern group was quite separate from, and even in competition with, the Argonne group. Many Northern researchers eventually took up positions at Argonne, however, essentially merging the two teams. This information comes from a telephone interview I conducted with Ross Overbeek, 10 Nov. 2010. The gesture made in the title of this essay to Walter Benjamin’s seminal paper “The Work of Art in the Age of Mechanical Reproduction” is not motivated merely by the fact that AURA is so named. Benjamin’s concept “aura” refers to a historically grounded notion of authenticity for art objects that cannot be (re)produced by mechanical means. The creators of AURA similarly believed that something of human mathematical practice could not be translated into mechanical processes. See Walter Benjamin, “The Work of Art in the Age of Mechanical Reproduction” (1936), in Illuminations, ed. Hannah Arendt, trans. Harry Zohn (London: Harcourt Brace Jovanovich, 1968), pp. 217–242. In response to the prospect of automation, ATP researchers articulated various beliefs and assumptions about what human mathematical work and thought consists in. As such, ATP is a “powerful disclosing agent” for images and understandings of mathematics. This concept comes from Lucy Suchman, who proposes that emotive robotics is a “powerful disclosing agent” for assumptions about the nature of human emotions. See Lucy Suchman, Human–Machine Reconfigurations: Plans and Situated Actions (Learning in Doing: Social, Cognitive, and Computational Perspectives) (Cambridge: Cambridge Univ. Press, 2006), p. 226.
human, rather than materializing from the cognitive ether, emerged from intimate and prolonged experimental work with the engine. Experimentation determined the character, form, and relevance of human contributions to AURA proof searches. By privileging and isolating a traditional notion of human mathematical thought in their design, the Argonne team in fact made possible radically new forms of intuition and insight grounded in experiential knowledge of computational behavior.

The surprising result of this feedback process was that the hybrid humans + AURA system was able to solve some open problems from pure mathematical fields about which none of the parties had any expert knowledge. Wos and Steve Winker, another ATP pioneer at Argonne, claimed that “researchers involved in the effort knew and know almost nothing of the fields from which the questions were selected. Rather than indicating the triviality of the questions (some of which are far from easy to answer), this fact shows the potential value of having access to an automated assistant—a colleague in the form of a theorem-proving program.” Although the problems humans + AURA solved were not thought to have particular mathematical significance, early ATP research nonetheless inaugurated certain negotiations between humans and nonhumans in the work of mathematics and produced new and surprising reconfigurations of mathematical problem solving.4 What follows is an investigation of how the hybrid humans + AURA system emerged at Argonne and how its unconventional contributions to mathematical knowledge were made.

Much has been written about what computers can and cannot do, whether computers can be intelligent, and whether computers can be part of the social networks to which historians of science usually attribute the primary agency of knowledge making. Many of these discussions assume that humans and computers are clearly bounded creatures whose essential abilities can be discussed and compared in the abstract. However, the character and capacities of computer work and human work depend on the nature of human–computer interactions and their mutual enabling and/or inhibiting in specific systems. Material tools are almost never the passive receptacles of socially predetermined knowledge, values, and practices but are actively and inextricably involved in the negotiation of those elements. And so I make no general claim about machine intelligence or about mathematical thought. I am interested only in the specific collaboration between the Argonne ATP research team and the AURA computer program and in the reconfiguration of mathematical knowledge production that materialized in their interactions.

This approach to the history of mathematics is grounded in a wealth of literature from the history of science and science studies that focuses on practice, experimentation, material culture, and the role of nonhumans in scientific knowledge making.5 Some

---

4 Larry Wos and Steve Winker, “Open Questions Solved with the Assistance of AURA,” in Automated Theorem Proving, ed. Bledsoe and Loveland (cit. n. 2), pp. 73–84, on p. 74. I propose that “mathematically uninteresting” developments ought to be studied in the history of mathematics for many reasons parallel to those offered by pioneers of the sociology of scientific knowledge who favored adopting a symmetry between the “winners” and the “losers” of scientific controversies. See esp. Steven Shapin and Simon Schaffer, Leviathan and the Air-Pump: Hobbes, Boyle, and the Experimental Life (Princeton, N.J.: Princeton Univ. Press, 1985). Limiting the history of mathematics to those problems now considered interesting reinforces the vision of mathematics as continuous and cumulative. What mathematicians consider interesting and significant is part of what needs to be explained, as is the appearance of continuity.

historians of mathematics have adopted similar approaches. Given that mathematics is not traditionally associated with experimentation and its tools are often relegated to the status of pedagogical supplements, this work in the history of mathematics has been groundbreaking. These historians of mathematics have successfully subverted traditional images of mathematics as an a priori, cumulative, and continuous discipline whose tools are mere incidentals. By elevating the materialities and associated practices of mathematical research to a vital position in historical investigation, these historians of mathematics invite earnest engagement with the history of science and science studies. This discussion of the AURA is a small contribution to that effort to enrich our understanding of how mathematical knowledges are made and made possible.

DESIGNING MATHEMATICS

AURA, among the earliest ATP engines, was developed by an eclectic team of researchers—the core of which was constituted by Wos, Winker, Ross Overbeek, Brian Smith, and Ewing “Rusty” Lusk—in the Applied Mathematics Division at Argonne, just outside of Chicago, throughout the 1970s and 1980s. Given that Argonne was established primarily as a nuclear reactor testing laboratory for the Manhattan Project, operated by the United States Department of Energy (DOE), it might be surprising to discover a team of ATP researchers there. Its existence seems to have been the achievement of William Miller, the director of the Applied Mathematics Division at Argonne from its founding in 1956 until 1965. The division was originally intended to provide information processing services, computing services, and mathematical analysis to other groups at Argonne. Donald MacKenzie reports, however, that Miller had expansive interests in automation and computing research for its own sake and that he was influenced by ATP practitioners elsewhere. Miller therefore founded an ATP group and garnered many resources for it, in spite of the fact that these researchers did not provide immediate and obvious services to the other divisions.7 Jack Holl, a historian of Argonne, indicated that Miller “was not modest and believed that his applied mathematics division was engaged in the most important, profound, and far-reaching activity of any group at Argonne.”8 Miller’s interest...


7 MacKenzie, Mechanizing Proof, p. 78. AURA did eventually provide useful applications in optimizing circuit design and program correctness proving, but this was not the original goal of the project.

8 Jack Holl, Argonne National Laboratory, 1946–96 (Chicago: Univ. Chicago Press, 1997), p. 125. Argonne continued to host a highly successful and robust team engaged in enrolling computers in the production of mathematical knowledge until about 2006. Around that time, the DOE cut funding to the group and several key researchers either retired or accepted positions elsewhere: Ross Overbeek to Stephanie Dick, email, 6 Jan 2011.
in exploring the full potential of digital computers, along with the hiring of the mathematically trained Wos for the division, prepared the way for ATP at the lab.

The basic insight behind ATP research was that sets of logical or mathematical axioms and permissible inference rules could be coded and stored within a digital computer. The computer could then be programmed to apply the inference rules to the axioms in order to deduce any logical consequences. Users could then input a mathematical proposition, P, and run the computer to see if any permitted sequences of inference led to P. If so, the series of steps taken by the program in that sequence would constitute a proof of P. In spite of the incredible speed and efficiency with which computers could execute instructions, ATP researchers quickly discovered that this method of proof seeking on its own was so inefficient as to be unusable. Not only did it lead to an exponential explosion of data, given how many inferences could be made, but there was also no way to know when a proof would be found. This exhaustive method guaranteed that every provable proposition P would eventually be proven (or its opposite would be proven), but it was not clear that this would happen during the lifetime of the mathematicians who actually cared about the problem. Wos and his colleagues indicated that the central problem with the exhaustive method was its inclusion of “numerous unprofitable inferences” that did not bring AURA any closer to the desired consequence of a given run.

Consequently, ATP research turned to the development of heuristics. Researchers sought to develop and program tricks, shortcuts, and strategies that would cut down on the number of “unprofitable inferences” produced by exhaustive search. Donald Loveland, himself a pioneer of ATP, divided the early heuristics into two design philosophies: he called one Logic and the other Human Simulation. Each approach was informed by an image of mathematics and a set of beliefs about how humans searched for proofs. Those working within the Human Simulation paradigm believed that much of what human mathematicians do could be articulated, formalized, and automated. They therefore consulted their understanding of what limitations, tools, and strategies informed human theorem-proving practices in order to design heuristics for their engines.

Alternatively, the Logic design philosophy, within which the Argonne group was firmly grounded, held that human thinking and computational calculation were qualitatively different processes, and so the latter ought not conform to our understanding of the former. They aimed to capitalize on what computers were uniquely good at in order to limit exhaustive deduction without the restriction of human resemblance. Wos’s belief that

9 The axioms and propositions were expressed in terms of Boolean formulas—called “clauses”—with disjunction as the primary logical operator.


11 Loveland, “Automated Theorem Proving” (cit. n. 2). The debates about how much of mathematical knowledge and practice could be articulated and formalized in order to permit automation resonate with Harry Collins’s discussion in Artificial Experts: Social Knowledge and Intelligent Machines (Cambridge, Mass.: MIT Press, 1992). Collins proposes that only those social practices that have themselves been sufficiently stabilized as to resemble mechanical processes can be exported to machines as intelligent purveyors of the associated knowledge.

12 Most, if not all, of the Logic camp ATP engines were based on a powerful inference rule called “resolution,” developed by Alan Robinson, who often visited Argonne as a researcher. Resolution was not based on any known human practice and was in fact difficult and counterintuitive for humans to understand. See J. A. Robinson, “A Machine-Oriented Logic Based on the Resolution Principle,” Journal of the Association for Computing Machinery, 1965, 12:25–41; and MacKenzie, Mechanizing Proof (cit. n. 6), pp. 77–83.
human mathematical practice could not, even in principle, be fully automated reinforced and extended the Logic paradigm. His conviction was likely related to the fact that, unlike many other ATP practitioners (who came from applied mathematics, philosophy, software engineering, or electrical engineering), Wos studied pure mathematics throughout his university career and had not studied computers at all before arriving at Argonne.13

In Wos’s experience, and that of other mathematicians with whom he spoke, the human mind works on mathematics even when the mathematician is not aware of it, and many crucial insights for mathematical problem solving emerge as unexpected “eureka” moments.14 This unconscious percolation could not be reduced to any mechanizable algorithm. Wos and Winker summarized this position as follows in their contribution to a 1983 conference on ATP: “Proving theorems in mathematics and in logic is too complex a task for total automation, for it requires insight, deep thought, and much knowledge and experience.” Overbeek has indicated that not everyone in the group held this position, at least not as strongly as Wos. Regardless, AURA’s design was premised on the conviction that moments of human insight that could not be mechanized were required for the work of proof.15

The prospect of automating the work of proof motivated groups of practitioners to articulate their beliefs about the nature of mathematics, the characteristics of human mathematical thought, and the nature of computers.16 This is one very preliminary sense in which the history of mathematics is tied to its material conditions. Encounters with new tools, objects, and abilities provide motivation and context for conversations about mathematics. Rather than simply revealing preformulated assumptions and beliefs, however, these materially motivated conversations can also reconfigure and create new images of the discipline. For example, although the Argonne team privileged and cordoned off human intuition as an unautomatable part of proof searches, those intuitions had to be put into the language and context of computation in order to be useful: intuitions had to be communicated to AURA. As will be discussed at length in the next section, one way that intuition was imparted to the computer program was as a weighting mechanism for privileging certain inference paths over others.17 Intuition was rethought in terms of a

13 He studied first at the University of Chicago and received his Ph.D. from the University of Illinois at Urbana-Champaign. Wos’s mathematics training was, however, not traditional in one important respect: he has been blind since birth. In order to pursue his mathematics education, he and his professors had to develop a Braille system with which he could encounter visual and textual mathematical objects. Wos’s blindness was also a factor in his turn from traditional mathematics to ATP research at Argonne. In spite of offering to share his salary with someone who could do the “board work” in his classes, and feeling fully supported by the Department of Mathematics at the University of Illinois at Urbana-Champaign, Wos was denied a teaching position by a dean at the university because of perceived obstacles presented by his blindness. See Tim A. Obermiller, “Top of His Game,” University of Chicago Magazine, 1997, 89:4 (online edition); further information comes from a telephone interview I conducted with Larry Wos, 4 Nov. 2010.

14 Wos recounted one story in which the primary realization for his development of quad arithmetic came to him while he was sleeping: “It just came to me. So if you want to give credit, if you want to get pedantic as hell—I’ve got a big imagination that works when I’m asleep!” Wos interview, 4 Nov. 2010. Wos’s position resembles that of Jacques Hadamard in The Psychology of Invention in the Mathematical Field (Princeton, N.J.: Princeton Univ. Press, 1945).

15 Wos and Winker, “Open Questions Solved with the Assistance of AURA” (cit. n. 4), p. 74 (quotation); and Overbeek interview, 10 Nov. 2010. AURA is therefore somewhat ironically related to Benjamin’s “aura”—the former denotes an engine that captures everything but the unmechanizable essence of human practice.

16 Different positions appear to be, at least in part, motivated by the different backgrounds from which ATP practitioners came. Further research is required before more specific claims about these positions can be made.

technical practice—weighting—in order to be computationally relevant, despite not being produced by automated means. Thus material tools and conditions can both reveal and reconfigure beliefs various practitioners have about mathematics. ATP practitioners not only formulated and articulated different images of mathematics in response to the possibility of automation that accompanied the advent of the computer; they also built those images right into their programs. Since the Argonne team didn’t think the work of proof could be fully automated, AURA was designed to collaborate with human users who could input their intangible intuitions and insights. However, AURA turned out not to be merely the passive receptacle of that image of mathematics: the very notions of “insight” and “intuition” that the Argonne team cordoned off for the human were reconfigured through extensive programming for and collaborating with AURA.

“THE QUICKEST AND SUREST WAY TO INSIGHT”

The Argonne group (and the Logic heuristic paradigm in general) adhered to a more experimental regime than the Human Simulation approach. Rather than applying theoretical models of the human mind, the Argonne group experimented extensively with different test problems to observe how AURA behaved in different conditions. They looked for patterns and search paths in the output clauses, hoping to isolate what looked to them like interesting or promising behavior. They would also identify what they thought were repetitive and uninteresting trajectories in order to retard the engine’s pursuit of them. Most of the human input to AURA runs was of this kind—the perception of important information or promising search paths. The capacity to identify what was “promising” or “interesting” was precisely one of those unautomatable human abilities, since, without guidance, computers indiscriminately produce all consequences. However, such value judgments did not come from nowhere: the Argonne practitioners decided what was important on the basis of extensive experimenting with AURA.

Further, intuitions about what was important weren’t confined to the mind for long. All human input had to be rendered usable by the computer—it had to be programmed. AURA was implemented on an IBM 360/370 mainframe computer. For the most part, the researchers used the punch card–based Assembly Language for those machines to write the program. As such, all of the commands, instructions, and information—and, of course, all of the human intuitions—had to be represented as holes punched in cards that were fed into the machine. The human ideas deployed in AURA runs both originated from and were translated into the material specificities of AURA. That material infrastructure was amenable to very different human insights and valuations of importance than traditional paper-and-pencil-and-human mathematics. The remainder of this section deals with the design details of AURA to demonstrate how mathematics was “thought” in that infrastructure.

The engine was constituted of several so-called modules or environments, each of which was a programmed subroutine that performed a certain task in the search for a


19 Assembly Language is a low-level (close to the hardware) symbolic representation of instructions and data that users wish to communicate to the computer. See Ned Chapin, 360/370 Programming in Assembly Language, 2nd ed. (New York: McGraw-Hill, 1973) (the first edition appeared in 1968).
There was one special module whose function was to read in all of the initial input information (called the SYSIN file). In the *Reference Manual for the Environmental Theorem Prover*, Smith calls on users to impart their preliminary predispositions and intuitions in the SYSIN file through a weighting mechanism, developed by Overbeek in the early 1970s, that restricted the order in which AURA executed inferences. This description highlights the intimate integration of computing and human insight central to AURA:

Weighting is the process that assigns measures of complexity to clauses in the clause space. The definition(s) of complexity can be chosen by the user to reflect some predisposition that may be, for example, based on an intuitive notion of how to direct the proof search. To create new inferences, clauses that seed the next inference . . . are selected by least weight. Derived inferences that are too complex (too heavy) are rejected.

Weighting allowed human collaborators to provide their preliminary sense of what kind of information would matter for a given proof. “Weighting templates,” included in the SYSIN file, assigned a value to each produced clause. For example, users could indicate that addition was more important than multiplication or that the sum of two sums was more important than the sum of two products. The values assigned by the templates would lead AURA preferentially to pursue those inferences whose ancestral clauses have the highest concentration of relevant information. The users’ sense of what information to weight was based on previous experiences of which algorithms cut down clause development in what ways and what patterns were visible in AURA’s output on various inputs: experimentation with and knowledge of the machine provided the content, the structure, and the sources of weighting insights.

For example, one of the earliest strategies developed at Argonne—the Unit Preference Strategy—prioritized shorter clauses over longer ones. Wos presented the strategy in direct response to the nature of clause generation from engines that use the resolution principle. Resolution permits the inference of new clauses only when it is known in advance that the resulting clause was not derivable from either parent clause alone. After extensive experimentation with resolution-based machines, the Argonne group decided that “it seemed worthwhile to orient the program to produce shorter and still shorter clauses in preference to other possible inferences”: more productive contradictions were formed between shorter clauses. The human intuition was based on a sense of “productive contradictions” that was, in turn, grounded in experimentation and the *a posteriori* analysis of resolution-based computational behavior.

A second form of intuition could be imparted to AURA in the SYSIN file through the Set of Support Strategy. This mechanism allowed human users to restrict what sequences

---


21 The weighting mechanism was originally developed by Overbeek for his doctoral dissertation in 1971; it was later published in Ross Overbeek, “A New Class of Automated Theorem-Proving Algorithms,” *J. Assoc. Computing Machinery*, 1974, 21:191–200. In addition to extensive work on the general implementation, the weighting mechanism was Overbeek’s main contribution to the AURA project.


23 In this context, two clauses were called “contradictory” if one contained at least one literal (atomic logical formula) that was the negation of a literal in the other. That is, if one clause contained L and the other contained not-L, these clauses were considered contradictory. See Wos et al., “Unit Preference Strategy in Theorem Proving” (cit. n. 10), p. 20.
of inferences AURA could make. Users could input a list of inferences, at least one of which must precede every subsequent inference, meaning that AURA couldn’t indiscriminately infer anything permissible. Through this mechanism, the human user could encode intuitions about what combinations, patterns, or sequences of inferences produced types of clauses that might be relevant to the theorem in question and prevent what was perceived as the unproductive swapping of uninteresting equalities. Even more than weighting, such intuitions were tied to extensive experimental knowledge of AURA’s runs and of which search paths and decision trees resulted from the application of various sequences of inference rules and what kinds of clauses they produced. Weighting strategies and Set of Support strategies were obviously not products of traditional mathematical “eureka” moments. Rather than providing some human insight about the mathematical problem at hand, users contributed insights about computation and AURA’s behavior.

In a sense, the Argonne team used AURA to produce the preliminary “scratch work” that traditional mathematicians often use to approach a new problem. They try several cases, construct examples, and search for patterns or useful analogies in order to guide their approach to a proof. However, offloading that part of the work to AURA fundamentally changed the type of insight garnered by the humans. The resulting human insights were not about the mathematical problem at hand, but about the behavior of the computer program. Changing the materials from pencil and paper to computer program and clause output, far from incidentals in the work of proof, reconfigured the practices and the insights of the humans—and this in spite of the team’s intention to preserve intuition from traditional mathematical practice.

A third form of human–AURA interaction was not part of the SYSIN file. Steve Winker believed that proof seeking involved more than deductive inferences. Humans, equipped with their unautomatable intuitions, also construct models of problems and search for counterexamples. He therefore devised a way for humans+AURA to build simple models of problems to structure proof searches even further. It was the addition of this mechanism that led to AURA’s successes in answering open problems. To understand what is meant by “model,” consider an example from ternary Boolean algebra; such algebras are defined by a set of five axioms built with five variables and two functions (one giving the product and the other giving the inverse). Whether any of the axioms were independent—meaning that they could not be derived from any combination of the other four—was an open question. The fourth and fifth axioms were already known to be dependent—that is, they could be derived from the others—but the question remained open for the rest. In order to show the independence of, for example, the first axiom, it was thought sufficient to find values for each of the five variables that satisfied all of the axioms except the first: if the first axiom could be derived from

25 The humans+AURA system qualifies as what Andy Clark and David Chalmers call a “coupled system” of extended cognition (as does the traditional humans+pencil system): “If, as we confront a task, a part of the world functions as a process which, were it done in the head, we would have no hesitation in recognizing as part of the cognitive process, then that part of the world is (so we claim) part of the cognitive process. . . . In these cases, the human organism is linked with an external entity in a two-way interaction, creating a coupled system that can be seen as a cognitive system in its own right.” Andy Clark and David Chalmers, “The Extended Mind,” in The Extended Mind, ed. Richard Menary (Cambridge, Mass.: MIT Press, 2010), pp. 27–42, on p. 29.
the others, whenever they were true, the first must also be true. In this example, Winker wanted humans+AURA to develop a model consisting of a revealing domain of values that could be checked for each axiom to see if this was the case.

Winker’s model-producing method capitalized on precisely the feature of humans+AURA that I have identified here: he emphasized that “no properties of the field of mathematics to which the method is applied are used in the method for” producing the model.29 Humans could have mathematically relevant insights about a problem based on interactions with AURA without necessarily having traditionally mathematical knowledge of a problem. The method consisted of three phases, each separated by an exchange of information between the human and the AURA. The shift in the character of mathematical intuition toward pattern recognition and a posteriori structural experimentation is most clear in this design feature.

First, AURA was run on a problem according to the standard protocols to produce many clauses, giving the human user more information to work with. This phase was based on the insight that “mathematical axiom systems are near-minimal . . . sets that express only the bare necessities to define the mathematical structure under study. Such axiom systems certainly do not include large numbers of extra ‘interesting’ identities.”30 As such, in the first phase AURA provided the user with thousands of printed clauses to peruse in search of ones that appeared interesting; again, these value judgments were based on experimental knowledge of AURA’s prior behavior and pattern recognition with the printouts. The human took the insights and equalities gleaned from the first phase and returned to AURA a partial model for testing. AURA then performed a second run, this time incorporating the incomplete model and outputting all those equalities or formulas that pertained to the missing parts. Again, the user studied thousands of printed clauses looking for patterns and potentially useful identities in order to complete the model. AURA then, without further human input, checked every instance of the model. For the case of ternary Boolean algebra, AURA checked that every instance of the model satisfied all axioms except one. If the verification run was successful—and in this case it was—the desired theorem would be proved. Therefore, in 1978 humans+AURA proved that, indeed, the first three axioms of ternary Boolean algebra are independent, thus making a novel contribution to mathematical knowledge.31

Each instance of human input based on the Argonne team’s idea of what was interesting, what was potentially important, and what might work—weighting, Set of Support, and model construction—was based on extensive experimentation and intimate knowledge of the lists of clauses AURA output on various previous runs. Overbeek made a revealing statement about the Argonne group’s methodology:

I began to realize that Wos took experimentation very seriously. . . . Most of our work together focused on laying out a set of problems that could not be solved, examining thousands of output clauses to determine what was going wrong, designing algorithms to correct the situation, and then observing the result. This is the first and most central aspect of Argonne culture: . . . Rule 1. Run experiments and observe what is going wrong. It’s the quickest and surest way to insight.

30 Ibid., p. 278.
Furthermore, Overbeek acknowledged that the kind of intuition needed to collaborate with AURA successfully was not simply logical or mathematical intuition. In fact, he proposed that “the trained logicians, with the exception of George Robinson, really had little idea of how the proposed algorithms actually worked (or why they did not work).”32 This was not traditional mathematical intuition; it was technological and computational intuition, grounded in experimental experience and a posteriori revelation. And yet, within the humans+AURA system, it led to solutions of difficult open problems in pure mathematics.

CONCLUSIONS: PERSPECTIVES ON THE AFTERMATH

The Argonne team was by no means unaware that AURA created new ways for working through mathematical problems and contributing to mathematical knowledge, in spite of their belief that the cognitive heavy lifting always fell to the humans. For example, in the conference proceedings from an American Mathematical Society annual meeting focused on ATP, Wos and Winker related two anecdotes as a preface to their+AURA’s solutions to open mathematical problems. The anecdotes, taken from work on problems in finite semigroups and equivalential calculus, were designed to demonstrate that AURA could enable and surprise its users in ways beyond what traditional mathematicians might expect. In the case of finite semigroups, the Argonne team knew so little that they misunderstood an open problem completely and in the first instance ended up proving the wrong thing altogether. After this was pointed out, they+AURA went on to solve the actual problem in 1981.33 In the case of equivalential calculus, after many unsuccessful runs Smith attempted to force AURA to recreate a known proof of some theorem using weighting. Instead, AURA went ahead and “found a proof half as long.” Even the heavy-handed imposition of the user’s knowledge could lead to surprising results.34 The Argonne team, in spite of reserving the most important work for the humans, proposed that AURA’s assistance made possible new and surprising possibilities for the discovery of proofs. Wos has even indicated that, if it weren’t for the fact that he may not have been taken seriously by journal editors, he would have liked to include the names of his automated reasoning engines, like AURA, as coauthors of the proofs that were produced.35

The Argonne group did not justify AURA’s value merely in terms of the theorems they proved with it. Similarly, AURA’s relevance to the history of mathematics is not in its contributing grand theorems to the corpus of mathematical truths. Instead, the Argonne team proposed that a new way of contributing to mathematical knowledge was made possible through collaboration with their unconventional assistant. AURA’s value to the history of mathematics is similarly grounded in its revealing how the image of mathe-

32 Overbeek, “Wos and Automated Deduction at ANL” (cit. n. 18), pp. 2–3 (emphasis added); and Ross Overbeek to Stephanie Dick, email, 11 Nov. 2010.

33 The problem, suggested by the mathematician Irving Kaplansky, was “Does there exist a finite semigroup admitting a nontrivial anti-automorphism but admitting no nontrivial involutions?” Wos and Winker, “Open Questions Solved with the Assistance of AURA” (cit. n. 4), p. 75. In their first attempt to solve the problem, the group misunderstood the meaning of “involution” (demonstrating their lack of knowledge about finite semigroups). Despite their lack of knowledge, however, the team successfully solved the problem with AURA in their second attempt. See Larry Wos and Steve Winker, “Semigroups, Anti-automorphisms, and Involutions: A Computer Solution to an Open Problem,” Mathematics of Computation, 1981, 37:533–545; and Wos and Winker, “Open Questions Solved with the Assistance of AURA,” pp. 75–77.

34 Wos and Winker, “Open Questions Solved with the Assistance of AURA,” p. 78. Hans-Jörg Rheinberger’s concept of an “experimental system” captures how even very disciplined scientific instruments can surprise their users. See Rheinberger, Toward a History of Epistemic Things (cit. n. 5).

35 Wos interview, 4 Nov. 2010.
matics, the practice of mathematics, and the way in which mathematical problems are understood and solved are constantly negotiated with and within the material culture of mathematical communities. AURA was not merely a receptacle of prenegotiated social practices but, rather, surprised and resisted its users, participating in the negotiations that precipitated a change in mathematical thinking and problem solving. I hope that historians of mathematics will continue to investigate more instances of such negotiation.

The history of science, because of its long interest in practice, technology, experiment, nonhuman agency, and material culture, offers an incredible wealth of tools and interlocutors to enroll in these investigations. In return, the resulting histories of mathematics offer a new image of humanity’s supposedly most abstract knowledge. Mathematics is not a continuous, cumulative, unchanging construct that grounds and shapes other sciences. Mathematical knowledge and practice are historically specific and constantly in dialogue with other disciplines, modes of knowing not traditionally classed as “mathematical,” and its many tools. At Argonne, mathematical knowledge was produced through extensive experimentation, engineering, programming, and collaboration with a nonhuman. Each element of that production subverts traditional images of mathematics, and none should be written out of the story later. Automated Theorem Proving is only one, albeit a very visible, case of such subversion. New images of mathematics open countless doors for historians of science to involve themselves with mathematics, to look for mutual influences between mathematics and other disciplines, and to trace the negotiations between even the most theoretical of knowers and their material tools, nonhuman assistants, and unconventional colleagues—even the ones clothed in electrical circuits, solder, dials, punched cards, core memory planes, wires, and thousands of printed clauses.