Strategic Thinking:
Educational Use & Cognitive Foundation*

Wilfried Sieg

Abstract. Strategic thinking is at the intellectual core of *the calculus for the 21st century*, which is not the mathematical calculus that emerged in the 17th century, but rather the logical calculus that was conceived in the same period by Leibniz, one of the two inventors of the mathematical calculus. Leibniz put great emphasis on a universal language to organize concepts, on rules to guide thinking and on mechanical algorithms to solve problems. The idea of the logical calculus came to theoretical fruition in the first half of the 20th century; it became absolutely vital during the second half of that century in the context of the computing revolution.

Proofs, functions and computations are the fundamental components of the logical calculus, but are scattered in logic, mathematics and computer science. Thorough familiarity with these concepts is no longer a privilege of a first rate education cutting across the boundaries of the three disciplines. On the contrary, it is a practical necessity for computer scientists and for students whose subject involves computational modeling, be they biologists, psychologists or economists. For students who reflect on the social impact of computers or conceive of mental processes as computations, it is equally central.

Our work aims to contribute to educational practice and research. It contributes to the former by expanding an innovative introduction to logic with a focus on proofs through two deeply integrated parts on functions and computations. It contributes to the latter by using this expanded, fully web-based course, *Computational Logic*, as a Learning Laboratory to evaluate the efficacy of pedagogical approaches, in particular, our central one of teaching and tutoring strategic thinking. Our approach has its cognitive foundation partially in the computational model of goal-directed logical reasoning AProS; the broader reflective use of logic in mathematical problem solving can, we conjecture, be modeled in extensions of AProS. Thus, the Learning Laboratory also allows us to test, refine and extend the cognitive foundation for strategic thinking.

0. Computational Logic. We are pursuing two central educational objectives: To give students logical tools for rigorous argumentation and conceptual organization, and to convey substantive knowledge of basic mathematical and computational notions, together with historical and philosophical perspectives. These objectives are to be reached in *Computational Logic* by carefully integrated and scaffolded material on the one hand, and by its effective and individualized presentation on the other. The presentation is computer-based, highly interactive and supported by sophisticated cognitive tutors.

---

* This is an essay attempting to articulate the strategic direction of the AProS project. (April 2006)

Computational Logic exposes students to the intellectual foundations of computer science anchored in logic and mathematics. However, it is also a unique Learning Laboratory, which allows us to examine restricted issues (for example, the effectiveness of special cognitive tutors) as well as our broad pedagogical approach that emphasizes strategic thinking. As the latter has its cognitive foundation partially in the computational model of goal-directed reasoning provided by AProS, we can investigate AProS's adequacy to model a meta-cognitively informed approach to problem solving. We have done extensive prior work through developing the web-based course *Logic & Proofs*[1] and the proof search method AProS. This provides crucial support for parts of our project we describe in greater detail under the headings: logical integration, dynamic tutoring, empirical assessment and natural reasoning.

I. Logical integration. The notions of proof, function and computation are core notions that evolved to their rigorous contemporary form from work in the foundations of mathematics beginning in the late 19th century. This work had enormous impact on the actual development of mathematics and created, in the 1930s, the theoretical notions for computer science. The material expanding *Logic & Proofs* is integrated to support the "evolution" of strategic thinking: we move from deductions in logic through proofs in set theory to arguments in computability theory.

Logic & Proofs gives a thorough introduction to modern logic. As it focuses on proofs, students gain insight into the validity and invalidity of arguments and the facility to construct proofs. The material on functions and computations is elementary, presupposing only what is contained in *Logic & Proofs*. At the same time it is rigorous, systematic and sophisticated; the last module in computability theory, for example, presents Gödel's Incompleteness Theorems. This material is

[1] The course has been offered with great success at Carnegie Mellon and at IUPUI (Indiana University and Purdue University at Indianapolis) since the fall of 2003 to approximately 300 students each year; we have refined the course iteratively, to a large extent in response to student feedback. - Various URLs for crucial software components can be found at the beginning of the list of references.

of deep interest, as any informed perspective on controversial issues concerning artificial intelligence and cognitive science must take it into account.

Proofs in set theory are no longer purely logical, but use definitions and theorems; ultimately, they are grounded in the axioms of Zermelo and Fraenkel. Through the systematic development that lead to central theorems involving functions, students learn to understand the axiomatic method and become familiar with the conceptual apparatus for most of contemporary mathematics. Set theory is mathematics' *lingua franca*. Dedekind emphasized the centrality of the function concept by saying that functions reflect "the ability of the mind to relate things to things, to let a thing correspond to a thing, or to represent a thing by a thing, an ability without which no thinking is possible."

The first three modules move from Boolean operations and Cartesian products to relations and functions; the crucial theorems include those of Cantor and Bernstein; cf. [14]-[15]. Except for Module 4, which introduces axioms for set theory, all modules are discussing functions. Module 5 defines natural numbers and justifies definition of functions by primitive recursion, whereas Module 6 transitions to the second expansion by introducing Gödel's "recursive functions." The latter are those number-theoretic functions that satisfy recursion equations and whose values can be calculated via rules.

Having moved from logical arguments to mathematical proofs and from general functions to calculable number-theoretic ones, we develop computability theory with "Turing machine computations" at its center; cf. [16]-[17]. As in set theory, the presentation is supplemented by historical and philosophical readings. Students gain insight into the nature, but also the limitations of computational and formal methods. Module 1 sets the stage by establishing the decidability of monadic logic and formulating the decision problem. It was this problem, the *Entscheidungsproblem*, that drew Turing to logic.

Module 2 introduces Turing machines to characterize computations. These machines initially seem so simple, but calculate the values of all recursive functions as is shown in Module 3. The construction of a universal machine allows us to establish, in Module 4, the unsolvability of the halting problem. As that problem is reducible to the decision problem, the undecidability of predicate logic can be inferred in the next module. In Module 6, this part culminates with an abstract proof of Gödel's Theorems. This is classical material and still at the center of any discussion concerning the foundations of computer and cognitive science.

II. Dynamic tutoring. *Logic & Proofs* is delivered online, through the Internet. Sophisticated mini-tutors and interactive learning environments (ILEs) enhance the presentation in accord with insights from cognitive psychology; cf. [30]–[32]. Every course module is structured in a similar way. A brief lecture introduces the topics of the module and formulates its learning objectives. That is followed by a richly interactive presentation of the core material, which is carefully scaffolded and divided into manageable sub-modules. The introduction of every major concept or technique is accompanied by an ILE, and particularly important matters are explained in mini-lectures. Every module ends with a reminder of the learning objectives, a quiz that serves as a self-assessment and a variety of exercises. Altogether, the course is a well-tested setting for the individualized instruction of analyzing and constructing arguments.

Strategic thinking is fostered theoretically by the explicit formulation of effective strategies in the text, and practically by having students work in the Carnegie Proof Lab (CPL). The CPL is a sophisticated Java applet with backend database support allowing students to construct and visualize proofs. It permits both backward and forward steps; in addition, it provides informative feedback by diagnosing mistakes and opening links to online help material. It cannot, however, provide hints on how to proceed when students are stuck. For that we are constructing a *Proof Tutor* that gives students dynamically generated advice. The advice-enabling component is the strategically guided proof search system AProS. It is a major innovation in automated theorem proving. (It has been designed and implemented mainly by Sieg and Ramsey; cf. [8]-[11]. The method is complete in the relevant logical sense: if there is a proof leading from premises to a conclusion, then AProS will find it.)

The Proof Tutor connects AProS with the CPL; when a student requests help in finding a proof, the Tutor asks AProS to complete the argument and analyzes the finished AProS proof. Based on its analysis, the Tutor provides hints: the first is a strategic one, whereas subsequent ones provide more concrete advice on how to proceed, until a particular next proof step is recommended. To provide tutoring for set theory, we have to expand CPL's functionality and AProS's strategic approach to proof search beyond logic. Support of mathematical problem solving is obtained by heuristics, which are conceived as in Polya [18] but are also informed by work on proof planning; cf. [19]-[21]. The crucial task is to isolate the "leading idea" for a part of mathematics and to formulate it procedurally. That requires tough analytic work and was accomplished for the proofs of Gödel's Incompleteness Theorems and the Bernstein theorem in [11] and [12]. - To make the presentation of computability theory interactive we will adapt *Turing's World* developed by Barwise and Etchemendy for [13]. We have Etchemendy's permission to turn the program into a web-based application.

III. Empirical assessment. The course is a computer-based, remarkable Learning Laboratory: it provides an astounding array of data, as our logging covers all interactions of students with text, ILEs and CPL. The CPL activities are recorded in such detail that the proof-process of any student can be literally reconstructed. We also collect the standard data concerning the background of students (major, year, relevant courses, SAT scores) and their performance measured by assignments and classroom tests.

It is of paramount interest for us to investigate how the explicit teaching of strategic thinking impacts proof construction in logic, and how it affects proof skills in set theory and computability theory. We will design experiments, and collect and analyze data to answer more specific questions concerning:

i) The effectiveness of computer-based instruction with rich ILEs and, in particular, dynamic tutoring;

ii) The transfer of strategic skills from finding formal proofs in logic to constructing informal arguments in mathematics;

iii) The intellectual accessibility of the material to students from differing academic backgrounds.

As to i), we are carrying out preliminary experiments this spring with two groups of about twenty students each. One group receives enhanced strategic instruction in their version of the material; the other group does the same proof construction exercises without the enhanced instruction. We want to see whether this impacts proof construction: Is there a closer fit between recommended and pursued moves, are searches shorter and lead to more direct proofs, are more difficult problems solved, and can students better articulate their strategic considerations? Of great interest is the question: Does a deeper theoretical understanding of proof search play a significant role? (These issues have been discussed in the literature; see [22]-[25].)

Once the Proof Tutor is implemented, we will explore the effects of dynamic tutoring. A pilot experiment will help us determine how students can be encouraged to use the advice as a scaffold, rather than a crutch. We conjecture that the systematic layering of advice - from broad strategic help to the suggestion of the next step - will have the desired effect, and that the emphasis on strategic thinking will facilitate transfer. Question ii) requires the development of the material on functions and computations; it will no longer be explored just by the analysis of logging data, but by means of detailed Think-Aloud-Protocols.

As to iii), it is clear to teachers of formal material that in a traditional classroom setting differing backgrounds of students require a delicate balancing act between "losing to boredom" those students who have some mathematical background and "losing to confusion" those who do not. The basic question, "Does the individualized instruction make complex formal material intellectually more accessible?" seems to have a positive answer: students with widely differing backgrounds have been taking *Logic & Proofs* very successfully. (The course has attracted a very diverse audience, as it satisfies at Carnegie Mellon and at IUPUI critical reasoning and mathematics requirements for students in the Humanities and the Arts.) The issue must be analyzed carefully.[2]

In the background is another extremely important question, "What is the role of human instruction and of discussion sections in the context of a computer based course?" Here we have to wait for the course's wider use. After all, the web-based material can be used in a variety of ways, for example, as a sophisticated, interactive e-book in a lecture course, as the exclusive vehicle for distance education or as the source for material in a discussion-based class. *Logic & Proofs* has been developed as part of Carnegie Mellon's *Open Learning Initiative* and, technically, can be offered to thousands of students. The course has been used outside of Carnegie Mellon only at IUPUI, as we have focused on improving the core material and its presentation. With the enhancements planned for spring and summer, the course will be highly developed, and we are making efforts to expand the user base for Fall 2006.

IV. Natural reasoning. The notions of proof, function and computation should be integral parts of any rigorous educational program: they are the fundamental components of the calculus for the 21st century, which in turn is central for the problem solving necessary for our social and economic well-being, as well as for a deeper understanding of the human mind. We suggested, here, a particular approach to teaching this material effectively, which is supported by and ultimately strives to foster strategic thinking.

The detailed CPL data allow us to analyze in the most refined way a central intellectual activity, rational argument construction, and compare it to the computational model AProS. Such an analysis is directly connected with work in

---

[2] In the fall of last year, two students from universities in New Orleans enrolled in *Logic & Proofs* (through the Sloan Foundation's project of bringing online courses to students displaced from colleges shut down by Hurricane Katrina). Sieg interacted with them once a week via e-mail, grading their pen & paper exercises, answering questions and making new assignment. Both students received an A in the course; one of them wrote to Sieg after having completed the course: "Actually, I worked harder in your class than I have ever worked in any class, … . But, in doing so, I learned so much. The logic that I learned helped me to get an "A" in my English class "Argument Writing". I was able to use some of the same ideas and write really well formed, logical arguments. I am also looking forward to taking the next logic course at Xavier as an elective or possibly go for the minor in logic. It's hard work, but I really like logic."

cognitive psychology. Rips developed in [26] a theory of mental proofs given by an automated theorem prover; that prover, PSYCOP, has deep structural similarities to ours. AProS's search procedure is production-rule based and can consequently be embedded into the cognitive architecture of Anderson's ACT-R, as formulated for example in [28]-[29]. Thus, above and beyond the logical and empirical support for the claim that AProS reflects deep structural features of human argumentation, there is a profound theoretical integration.

The extended AProS, we conjecture, will provide an effective and empirically supported characterization of the cognitive operations involved in the task of proving a mathematical theorem. Thus the reasoning involved here is not only natural from the quasi-empirical standpoint of logic, but also from a genuinely empirical psychological perspective. Such a result would undoubtedly topple the common wisdom assumption that logic has no significant role in finding proofs; it would also have a profound effect on the general teaching of mathematics.

References

The first three items are URLs that connect directly to:
i) The AProS site and thus the automated theorem prover, which is the major component underlying the *Proof Tutor*,
http://www.phil.cmu.edu/projects/apros/index.php

ii) The latest version of the *Logic & Proofs* as it is currently offered to about 40 students at Carnegie Mellon and around 80 students at IUPUI,
http://www.cmu.edu/oli/

iii) More comprehensively, to the course but also to the *Carnegie Proof Lab* (CPL), in which students construct proofs, and to particular interactive learning environments (ILEs)
http://www.phil.cmu.edu/~cpldemo/

[1] G. W. Leibniz: *Leibniz Selections* (P.P. Wiener, ed.); New York, 1951. [Essays 3, 4, and 5 from Part I *On Method*]

[2] A.M. Turing: Collected Works – Mechanical Intelligence (D.C. Ince, ed.); North-Holland, 1992. [The essays: Lecture to the London Mathematical Society; Intelligent Machinery; Solvable and Unsolvable Problems}

[3] R. Penrose: *The Emperor's New Mind*; Oxford University Press, 1989.

[4] P. Suppes (ed.): *University-level computer-assisted instruction at Stanford: 1968-1980*. Stanford, 1981.

[5] W. Sieg and R. Scheines: Searching for Proofs (in sentential logic); in: L. Burkholder (ed.), *Philosophy and the Computer*, Westview Press, 1992, 137-159.

[6] R. Scheines and W. Sieg: Computer Environments for Proof Construction; Interactive Learning Environments 4 (1994), 159-169.

[7] W. Sieg: *Mechanisms and Search* – Aspects of proof theory; AILA Preprint, 1992/3, 79 pp.

[8] W. Sieg and J. Byrnes: Normal Natural Deduction Proofs (in Classical Logic); Studia Logica 60 (1998), 67-106.

[9] J. Byrnes: *Proof Search and Normal Forms in Natural Deduction*; Ph.D. Thesis, Carnegie Mellon, May 1999.

[10] D. Prawitz: *Natural Deduction – A proof-theoretic study*; Stockholm, 1965.

[11] W. Sieg and C. Field: Automated search for Gödel's proofs; Annals of Pure and Applied Logic 133, 2005, 319-338.

[12] I. Kash: A partially automated proof of the Cantor-Bernstein Theorem; Senior Honors Thesis, Adv. W. Sieg; Computer Science Department, 2004.

[13] J. Barwise and J. Etchemendy: *Turing's World – an introduction to computability theory*; CSLI, Stanford, 1993.

[14] K. Devlin: *Sets, functions, and logic – an introduction to abstract mathematics;* Chapman & Hall, 2004.

[15] D.J. Velleman: *How to Prove it – a structured approach*; Cambridge University Press, 1994.

[16] M. Davis: *Computability and Unsolvability*; New York, 1958.

[17] D. Mundici and W. Sieg: Computability, in: Routledge Encyclopedia of Philosophy, 1998.

[18] G. Polya: Mathematical Discovery – On understanding, learning, and teaching problem solving; Wiley & Sons, 1981.

[19] E. Melis and J. Siekmann: Concepts in Proof Planning; in: *Intellectics and Computational Logic*; Kluwer, 1998, 249-264.

[20] E. Melis: Why proof planning for Maths education and how?; in: *Mechanizing Mathematical reasoning*, LNAI 2605, 2005, 364-378.

[21] A. Bundy: Planning and patching proof; in: *Artificial Intelligence and Symbolic Computation* (B. Buchberger and J.A. Campbell, eds.); LNAI 3249, 2004, 26-37.

[22] R. Elio and P.B. Scharf: Modeling novice-to-expert shifts in problem-solving strategy and knowledge organization; Cognitive Science, 14, 1990, 579-639.

[23] K. vanLehn, R.M. Jones and & M.T.H. Chi: A model of the self-explanation effect; The Journal of the Learning Sciences, 2(1), 1992, 1-59.

[24] J.I. Heller and F. Reif: Prescribing effective human problem-solving processes: Problem descriptions in physics; Cognition and Instruction, 1(2), 1984, 177-216.

[25] K. vanLehn, D. Bhembe, M. Chi, C. Lynch, K. Schulze, R. Shelby ea: Implicit vs. explicit learning of strategies in a non-procedural skill; in ITS 2004. Berlin: Springer.

[26] L. J. Rips: *The Psychology of Proof*; MIT Press, 1994.

[27] P.N. Johnson-Laird: *The Computer and the Mind*; Harvard University Press, 1988.

[28] J.R. Anderson: Human symbol manipulation within an integrated cognitive architecture; Cognitive Science, 29(3), 2005, 313-341.

[29] J.R. Anderson, *Rules of the Mind*, Erlbaum, Hillsdale. N.J., 1993.

Our general approach to computer-based instruction is informed by the rich literature; we mention in particular:

[30] J.R. Anderson, A.T. Corbett, K.R. Koedinger and R. Pelletier: Cognitive tutors: Lessons learned; The Journal of the Learning Sciences 4(2), 1995, 167-207.

[31] R.E. Eberts: Computer-based instruction; in: M. Helander, T.K. Landauer, and P. Prabhu (eds.), *Handbook of Human-Computer Interaction*, Elsevier Science, 1997, 825-47.

[32] K.R. Koedinger, Toward evidence for instructional design principles: Examples from Cognitive Tutor Math 6; in: *Proc. 24th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, Columbus, 2002, 21-50.