TURING IN COMPUTATIONAL THEORY: AN APPROACH TO BROADEN INTEREST IN COMPUTER SCIENCE CLASSES

Ken Abernethy
Furman University
Dept. of Computer Science
Greenville, SC 29613, USA
ken.abernethy@furman.edu

Paula Gabbert
Furman University
Dept. of Computer Science
Greenville, SC 29613, USA
paula.gabbert@furman.edu

Kevin Treu
Furman University
Dept. of Computer Science
Greenville, SC 29613, USA
kevin.treu@furman.edu

ABSTRACT

The authors have instituted curricular changes at their institution that attempt to address the issue of declining enrollments in computer science courses. These changes involve a deliberate attempt to better demonstrate the inherent multi-disciplinary nature of computing. More specifically, we illustrate how an auxiliary study of Alan Turing's original writings as well as his life has been used to connect the traditional computational theory course to the broader non-computing world.

Keywords

Computational theory, cognitive science, Turing machines, Alan Turing

1. INTRODUCTION/BACKGROUND

The decreasing numbers of students studying computer science in the USA has been well-documented in the Computer Research Association's Annual Taulbee Survey of Ph.D.-granting CS Departments [11]. In addition, survey results from the Higher Education Research Institute at the University of California at Los Angeles (HERI/UCLA) [12] indicates that the popularity of computer science among incoming freshmen at *all* undergraduate institutions has dropped significantly in recent years. More specifically, the HERI/UCLA surveys conclude that the percentage of incoming undergraduates indicating that they would major in CS declined by over 60 percent between the Fall of 2000 and 2004, and is now 70 per cent lower than its peak in the early 1980s.

There has been no shortage of suggested reasons for these declines, including the dot-com bust, reliance on off-shore programmers, and other economic realities. However, a growing number of observers have offered some non-economic possibilities as well. For example in [8], Morris and Lee argue that the decline in undergraduate computer science enrollment may be because computer science hasn't chosen to emphasize its grand challenges. Rather than touting the excitement of trying to magnify the power of the human mind through machines, the field has perhaps become overly focused on more narrow applications. In other words, Morris and Lee suggest that an approach that reduces the current focus on the low-level details of the discipline and emphasizes instead the many connections that computer science has to larger fundamental questions in the world might indeed have some positive impact on the declining enrollment picture.

This theme of the importance of connecting the teaching and learning of CS to the broader world has also been present in other works. For example, in their book about how we might attract more women to the field, Margolis and Fisher [6] make a number of suggestions. Some of them relate directly to this theme. The study revealed that while both male and female students rank enjoyment of computing as an important factor for choosing a computing major, over 40% of females studied ranked the versatility of the field as an important factor for choosing the major while less than 5% of males did. Consequently, Margolis and Fisher recommend showing links between theoretical computing and the applications of computing from the beginning of the curriculum.

To some extent, the success of computing has perhaps become its own worst enemy in attracting young people to the discipline. The World Wide Web and its associated technologies have made computing ubiquitous and have ironically removed some of its "gee whiz" impact. In other words, the more success computing has had in delivering truly revolutionary *practical results*, the less inherent appeal the technology *per se* can have. In a sense, the mystery has been removed; so the characteristics of the discipline that appealed to a sizable percentage of students in 1980s appeal to a much smaller percentage now. This is in fact another facet of the argument that Morris and Lee have made. In [8], they ask the question "Are we about a machine or an idea?" Their conclusion is that we have allowed ourselves to become too much about the machine and not enough about ideas.

The authors have long believed that tempering our excessive focus on narrow applications would improve the CS curriculum. In this paper we discuss one way we have attempted this in the upper-level computational theory course. In the past, we have made analogous attempts in lower-level classes as well. [2, 3]. In the final section we discuss future efforts to extend this theme to other parts of the curriculum, specifically those with a significant chance of influencing students at a time when they have not decided about their majors.

2. A SYSTEMIC APPROACH

The typical computational theory course [7] provides an excellent opportunity to experiment with approaches to teaching computer science that are less about the machine and more about ideas. To begin with, the standard material – which includes basics of discrete mathematics, proof techniques, finite state machines, pushdown automata, Turing machines, intractable and unsolvable problems – is already less about (physical) machines than most other courses in the curriculum. It nevertheless tends to suffer significantly from the problems noted previously in that the traditional approach to the material can be too self-contained, and too focused on very specific theoretical problems for which any connection to bigger, significant ideas is often difficult to make clear to the students. It's not that the connections aren't there, it's just that they can get lost or obscured in the mathematics. The subject matter is viewed by the majority of students as arcane and disconnected. It is therefore widely regarded as an unpopular class, perhaps the least anticipated in the curriculum. As such, we believe that computational theory serves as a microcosm of the computer science discipline as a whole. That is, it is viewed as being rather irrelevant, focused on trivial details and something to be avoided when possible.

Also like the discipline as a whole however, the opportunity is there in computational theory for a *very* rich consideration of ideas and connections to the broader world. Over the course of 9 years (since the fall of 1997), in 7 offerings of computational theory we have experimented with various approaches to seizing this opportunity. Our efforts in recent years have revolved around a study of the life and work of the Turing machine's namesake, Alan M. Turing.

In the U.S., students beginning a study of computational theory have almost universally never heard of Alan Turing. (Those few who have might know of him as a fictional character from Neal Stephenson's novel *Cryptonomicon*.) Even at the conclusion of a typical computational theory course, students only know him as the person after whom the Turing machine is named. However, we have learned that when a non-trivial amount of class time is allocated to a study of Turing's life and work, the course becomes more interesting both to teach and to take due to the remarkable variety of issues to consider. Furthermore, the traditional course material takes on an added energy and appeal because much of it can be a portal to the larger questions and topics raised by the Turing study.

After one course offering using Andrew Hodges' thorough but dense biography *Alan Turing: The Enigma* [4], we settled on Hodges' abridged volume *Turing* from a series entitled "The Great Philosophers" [5]. We have supplemented this book with a number of web resources, most notably Hodges' own official web site. [1, 9, 10]

The logistics of our approach to incorporating this material is not particularly unique. We focus on reading and writing assignments and in-class discussion, all of which is required of the students and factors into the grade. It is the context (i.e., a computational theory class) and the extent of the incorporation that we do feel offers something new.

The book *Turing* breaks down naturally into at least four segments, each with a discernable theme. In our course, students are assigned these sections (or subsections, as appropriate) on a weekly basis. They are responsible for doing the reading and completing a writing assignment. This takes the form of either a position paper or a brief online commentary on one or more of the issues raised by the reading. We have experimented with having the position papers posted online, with other students invited to post comments. Recently, however, we have settled upon having students create and add to online discussion threads pertaining to specific weekly topics. In both cases, the writing component is followed by an in-class discussion session in which all students are expected to participate. A sampling of topics and discussion questions is given below.

2.1 Turing Machines, the Limits of Computing and the Nature of Mind

The first segment of Hodges' book – encountered very early in the course – seems at first to put the cart before the horse. That is, it introduces and defines Turing machines long before the topic is broached in a typical textbook. We have found that far from being confusing, however, this sets a context for our entire study of abstract machines leading up to Turing machines, so students always are aware of the destination and the larger issues.

The origin of the Turing machine (Turing's solution to the *Entscheidungsproblem*) as well as how it works is discussed in this section, against the backdrop of Turing's life, most notably the loss of his friend Christopher Morcom and his relationship with his mother. These elements are woven into analysis of Turing's own theories about the Turing machine – as more than just a definition of the limitations of computing, but as a model of the working of the human mind. Sample questions raised for discussion of these topics include:

- Are you convinced based on what you've read so far that Turing came up with a general model of computation? That anything a Turing Machine can't compute *can't be computed at all*?
- What do you think of the basic analogy that led Turing to the development of his machine – the idea that the human "computer" also has a finite number of states of mind when working out a problem? Is there a weakness to this argument?
- How does the issue of free will versus determinism factor into this? That was obviously much of what intrigued and motivated Turing.
- Do you think it is possible to build a formal model of creative, original thought?

Issues raised during the discussions of this material animate, excite and sometimes anger students. It is clear to us that the students are eager to explore the connections between the textbook material and the larger issue of the very nature of the human mind that Turing himself explored. In the end, these discussions have significantly livened our study of theoretical models of computing.

2.2 Code-breaking, Building a Brain, and World History

Having established the relevance of Turing and his work to the course content, the following sections allow us to engage the students in a study and discussion of Turing's contributions to world history which are, the students agree, remarkably significant for a relatively unknown person. In studying Turing's work at Bletchley Park during World War II, we take up the topic of cryptography as well as ethical issues relating to Turing's own moral dilemma as a pacifist working for the war effort. Turing's unsuccessful post-war efforts to "build a brain" open our discussion of the history of computing, and the universal Turing machine idea leads to a discussion of the origin of the general-purpose, programmable computer.

Hodges' speculation that Turing's own opinion with regard to the computability of intuition changed during the war years provides ample opportunity for additional discussion of issues of mind. Sample discussion questions for this material include:

- Turing came up with a "perfect application at the heart of the world crisis." What was this application? What impact did Turing have on the course of history in World War II?
- In this reading we really get a sense of the historical significance of the ideas of the Turing machine. What basic concept was Turing thinking of when he spoke of a "practical universal machine" that would need "no fresh engineering, just fresh codes"? And how was the universal Turing machine connected to this idea?
- Turing recognized that it would be necessary for his machine to be able to *learn* – to demonstrate behavior not programmed into it. Do you think this is possible?
- Throughout this reading we see how the "political and economic needs of the day determined new ideas" in terms of computing. What are the examples in this case? Other examples you can think of from history?
- Based on what you've read, do you think Turing deserved fame for his contributions?

The direct way in which the Turing machine and its creator relate to world and computer history provides a rich and fascinating context for our study of computational theory principles. We observe students (many of whom may have been skeptical at first) embracing Turing as a significant historical figure, a perspective which almost certainly serves to increase overall interest in the course.

2.3 Machine Intelligence and the Turing Test

A second central computer science concept that bears Turing's name is the Turing test for artificial intelligence. In previous reading and discussion, we consider the nature of the human mind, and whether or not it can actually be described as a machine. This material is effectively the inverse – whether or not a machine can be intelligent, whether or not it can have a mind. In discussing this, we analyze the question of what constitutes intelligence, focusing on Turing's own belief that "the imitation of intelligence *is* intelligence" (that machines can produce "apparently non-mechanical behavior"). Hodges relates this material to Turing's code breaking work through his very accessible analysis of Searle's Chinese room paradox.

One of the most fascinating and thought-provoking aspects of this material is our consideration of Turing's own answers to the various objections raised against his assertion that machines can be intelligent. His musings on theology and consciousness in particular provide fuel for lively discussion and debate. Sample questions include:

- What roles do "discipline" and "initiative" play in intelligence? Can either be copied by a machine?
- Is learning computable, in that there is an effective procedure that describes it?
- What are some of the arguments against the validity of the Turing test?
- Turing's idea of a machine being able to model the brain "runs counter to intuition but is not at all easy to refute". Do you agree?
- What would convince you that a machine was behaving intelligently? Is intelligence a uniquely human trait?

During discussions of this material, we have found that students often find their humanity to be challenged on a fundamental level – that they exhibit the "commonsense repugnance to the idea of machines being credited with thought" that Turing spoke of. Clearly, this is not the type of connection that is typically made in a computer science class, yet it fits in naturally with computational theory when Turing's *philosophy* is studied along with the Turing machine and Turing test.

2.4 "Seeing the Truth" and Common Humanity

The concluding sections the book return to previous topics, including whether or not the human mind is capable of processing that which is "fundamentally non-computable", and the implications of the Turing test. These issues he places into a larger context, against the backdrop of the events of Turing's life.

Roger Penrose posits that "seeing the truth" is the essence of intelligence, and asserts that this is a human ability which is not computable. Yet Turing ultimately credits the mechanical with the capacity for "moments of world-shattering inspiration". What it means to "see the truth" in a burst of insight that only later is rigorously proven is a very fruitful area for discussion in a class that involves mathematical reason and theorem proving. Can such moments feasibly be expressed as an effective procedure? And could a machine thus simulate it?

Of the Turing test, Hodges writes that it "requires not so much a judge as a jury: not an expert, but common humanity". As our class discussions turn to the end of Turing's life, we couple this notion with the idea of "seeing the truth" as a way to assess the whole of Turing's life and accomplishments. Specifically, we urge the students to view Turing and his ideas from the standpoint of a jury – that is, without assuming there is a definitive answer. We he in fact a great philosopher? A significant historical figure? In what ways did he "see the truth"? Selected discussion questions include:

- Turing's hypothesis seems to be that the actual physical workings of the brain are not relevant, as long as we can model its behavior accurately. Do you agree?
- How do you evaluate Turing's 50-year prediction regarding the capabilities of computers and general attitudes towards machine intelligence, now that we're 50 years past the time when he made it?
- The circumstances of Turing's death were tragic. Do you think he would have been likely to commit suicide if he were turning 42 this year instead of in 1954? Would his credibility have been affected, as it was in his time?

As the last question suggests, there is a significant sociological connection that can be made as we delve into the circumstances surrounding Turing's later years and untimely death. Additionally, his prediction of computing capabilities 50 years in the future allow us to further bind him to contemporary computing issues in our discussion.

3. EVALUATION

The students in the most recent computational theory class were quite positive in their reception of the inclusion of the Turing related subject matter. In a survey, the students were asked whether they disagreed or agreed with 6 statements concerning this material. They were asked to respond with an integer from 1 (strongly disagree), 2 (disagree), 3 (neutral), 4 (agree), and 5 (strongly agree). The results of the survey (11 respondents) are shown in Table 1. There were 11 students in the class and all responded to the survey. The average ratings reported are the numerical means of the scores in each case ($\Sigma x_i/N$).

Statement	Average Rating

My basic attitude toward the Turing supplementary readings is one of enthusiasm.	4.0
My basic attitude toward the class discussions about the Turing supplementary readings is one of enthusiasm.	4.3
The Turing supplementary material (readings and discussions) had a significant impact on my understanding of computational theory.	3.4
The Turing supplementary material (readings and discussions) had a significant impact on my understanding of computer science beyond the scope of this particular course.	3.9
My class experience was enhanced by the Turing supplementary material (readings and discussions).	4.1
I believe that Alan Turing was an important philosopher.	4.4
I believe that Alan Turing was an important historical figure.	4.2
I would endorse the idea of similar readings in other CS courses.	3.5

Table 1: Student responses to Turing supplementary material.

As the table demonstrates, the students were enthusiastic about the readings (average score of 4.0) and discussions (average score of 4.3). This confirmed our observation that they participated in the discussions with interest and a high-level of preparation and thoughtfulness. While the students were somewhat positive (3.4) that the readings and discussions added to their understanding of the computational theory material, they were more positive (3.9) that the readings and discussions had a significant impact on their understanding of computer science beyond the scope of this particular course. They were also very positive about the material enhancing their class experience (4.1). Of course, it is these latter two factors that we were hoping would be an outcome of the use of the materials. The students were very positive about Turing's importance as a philosopher (4.4) and a major historical figure (4.2). These responses were particularly encouraging because an informal survey on the first day of class revealed that none of the students knew very much at all about Turing -- even in the more limited context of his contributions to computer science. They were not quite as positive in their responses (3.5) when they were asked whether they would endorse the idea of similar readings in other CS courses. Several made comments that their endorsement of the idea would depend on the particular course chosen, indicating that they thought it would be a good fit for some and not as good a fit for others. It is worth noting that no one *disagreed* with the idea, with roughly half the class neutral and half in agreement (2 actually *strongly agreed*).

4. FUTURE WORK

We believe that our experiment with using the life and work of Alan Turing as a way of connecting computational theory students to issues of the broader world has been very successful. We are therefore encouraged to continue to try pedagogical approaches to computer science that emphasize ideas over machines in ways that demonstrate the connectedness of the field to the larger world. We have been and will continue to use a variety of non-academic texts as supplemental readings in other courses [2, 3], utilizing a read/write/discuss paradigm similar to the one described in this paper.

At our institution, we have an opportunity in the form of a restructured university curriculum that will include mandatory freshman seminar courses. We are at this time preparing a freshman seminar that will be devoted to Alan Turing and will include all of the issues described above, only to a greater depth. There are few individuals whose life and work can so effectively raise as many interesting issues for discussion and debate in as wide a spectrum of areas – math, computer science, philosophy, biology, sociology, etc. We anticipate that this course will be effective in demonstrating the connectedness of computer science to students at their most impressionable, with the likelihood of enticing them to continue studies in the field.

5. REFERENCES

[1] The Alan Turing Home Page (http://www.turing.org.uk/turing/)

[2] Gabbert, P. and K. Treu, "*Being Digital* in the Non-Major Course," *The Journal of Computing in Small Colleges*, Vol. 13, No. 3, pp. 66-72, January 1998

[3] Gabbert, P. and K. Treu, "Experiments with the Use of Popular Press in the Computer Science Curriculum," SIGCSE Bulletin, Vol. 32, No. 1, pp. 124-128, March 2000

[4] Hodges, A., *Alan Turing: The Enigma*, Walker & Company, New York, NY, 2000.

[5] Hodges, A., *Turing*, Routledge, New York, 1999.

[6] Margolis, J. and A. Fisher, *Unlocking the Clubhouse*, The MIT Press, Cambridge, MA.

[7] Martin, J., *Introduction to Languages and the Theory of Computation*, Third Edition, McGraw Hill, Boston, MA, 2003.

[8] Morris, J. and P. Lee, "The Incredibly Shrinking Pipeline is Not Just for Women Anymore", *Computing Research News*, Vol. 16, No. 3, May 2004. (http://www.cra.org/CRN/articles/may04/morris.lee.html)

[9] *Stanford Encyclopedia of Philosophy* (http://plato.stanford.edu/entries/turing/)

[10] The Turing Archive for the History of Computing (http://www.alanturing.net/)

[11] Vesgo, J., "Drop in Batchelor's Degree Production," *Computing Research News*, Vol. 18, No. 2, March 2006. (http://www.cra.org/CRN/articles/march06/vegso.html)

[12] Vesgo, J., "Interest in CS as a Major Drops Among Incoming Freshmen," *Computing Research News*, Vol. 17, No. 3, May 2005. (http://www.cra.org/CRN/articles/may05/vegso.html)