MATHEMATICS AND COMPUTER SCIENCE:

EXPLORING A SYMBIOTIC RELATIONSHIP

Authors:

Ralph Bravaco	Shai Simonson
Department of Computer Science	Department of Computer Science
Stonehill College	Stonehill College
Easton, MA 02357	Easton, MA 02357
ralph@stonehill.edu	shai@stonehill.edu

Classification:

Computers, Fractals, Chaos, Number Theory and Cryptography, Problem Solving, Other

MATHEMATICS AND COMPUTER SCIENCE: EXPLORING A SYMBIOTIC RELATIONSHIP

ABSTRACT

This paper describes a “learning community” designed for sophomore computer science majors who are simultaneously studying discrete mathematics. The learning community consists of three courses: Discrete Mathematics, Data Structures and an integrative seminar/lab. The seminar functions as a link that integrates the two disciplines. Participation in the seminar enables students to study and appreciate the mutually beneficial relationship that exists between mathematics and computer science—a relationship that students often fail to see. Through carefully planned experimentation, students investigate how mathematics is used to formalize computer science, and how computer science can be used as a tool to explore mathematics.

A LEARNING COMMUNITY. WHAT IS IT?

Like hundreds of colleges and universities, Stonehill College has established the "learning community" as a major component of the college's General Education curriculum. While no universal definition of a learning community exists, the following description is a fairly accurate one of the Stonehill model:

> [A learning community] is any one of a variety of curricular structures that link together several existing courses – or actually restructure the material entirely – so that students have opportunities for deeper understanding and integration of the material they are learning. [1]

At Stonehill, all sophomores enroll in a learning community consisting of three courses: two courses from different disciplines and a third integrative seminar. The seminar functions as a link that integrates the two disciplines. Stonehill learning communities come in various flavors: one learning community, "Eat, Drink and be Merry", links an art course ("Food and Body Image in Art") with a philosophy of science course ("Feeding the Body") in an exploration of the multiple meanings of food in our culture. Another, "Laying Down the Law: Adolescents, Courts and Legislative Control", integrates political science with criminal justice, in an investigation of the problems of juvenile delinquency. Other learning communities intertwine Film Studies and History, Chemistry and Microbiology, and even Religious Studies and Ecology. The learning community described in this paper connects mathematics and computer science. What better bedfellows?

A LEARNING COMMUNITY INTEGRATING DISCRETE MATHEMATICS AND DATA STRUCTURES.

Typically, a first-semester sophomore computer science major at Stonehill College enrolls in both Discrete Mathematics and Data Structures. These requirements presented us with the perfect opportunity to create a new learning community linking mathematics and computer science: a three course sequence consisting of our traditional Discrete Mathematics and Data Structures courses together with an integrative seminar where students might study and appreciate the mutually beneficial relationship which exists between mathematics and computer science. Through carefully planned experimentation, students investigate how mathematics is used to formalize computer science, and how computer science can be used as a tool to explore mathematics.

THE SKELETAL STRUCTURE OF THE INTEGRATIVE SEMINAR

The integrative seminar was scheduled once a week for 2.5 hours. The format of the seminar was a closed lab where each student had access to a computer. A fifteen-week semester was divided into five three-week blocks and five self-contained topics were chosen. Students worked in groups of two, devoting three weeks to each topic. The five topics chosen for the seminar were:

Fractals, Chaos, and Theatre - An exploration into complex numbers, chaos, and computer graphics. Tom Stoppard's award winning play *Arcadia*[2] served as the catalyst for this first module. Fractal geometry, chaos, and iterative algorithms are all major players in Stoppard's hilarious comedy. And, coincidently, each of these mathematical topics is virtually impossible to explore without computers and computer science. Stoppard gave us the perfect springboard for our learning community - a good starting point. From the Mandlebrot set to chaos to the butterfly effect, this lab demonstrated how computer science functions as an aid to the mathematician. During the final session of this lab, students watched a taped interview with Tom Stoppard ("Mathematics in *Arcadia*"[3]*)* as well as several scenes from the play. Was *Arcadia* absolutely necessary for our purposes? Not really. But, the play certainly made the experience more fun, tied the topics into one neat bundle, and added a touch of liberal arts.

Cryptography - A study of the world of abstract number theory and its application to the modern world of e-commerce. In the previous lab, students saw how computers aid the mathematician. Conversely, in this lab, students discovered how some very old mathematics (number theory) is used for some very modern computer applications. During the first meeting, the students wrote programs to cement their understanding of some basic concepts in number theory and implement various cryptographic methods. They simultaneously developed solutions to problem sets in number theory, to ensure that their understanding of the foundations was clear.

During the second meeting, students used their programs to encrypt and decrypt various texts. We even held a contest to see who could crack whose codes. This lab presented the flip side of the previous lab: here, students saw how mathematics aids the computer scientist. During the third week's lab (which was designated "enrichment"), we screened a video version of Hugh Whitmore's *Breaking the Code*, a play based on the life of mathematician/computer scientist/code-breaker, Alan Turing. The character of Alan Turing is brilliantly brought to life by the animated performance of Sir Derek Jacobi. [4]

Man vs. Machine - A psychological exploration of the results when one tries to incorporate knowledge of a strategic game into a program. In this third lab, students learn the important lesson that you can program a machine to perform a task better than you yourself could do it. Technically, the lab reinforced concepts of recursion, depth first search, and time complexity. A secondary theme of this lab was the tension between theoretical and experimental computer science. When a problem is intractable, the theorists go on to another problem, and the engineer sharpens his pencil. Students need to tweak and experiment with techniques to make sure their programs can cope with a theoretically intractable problem. During the final session, we screened a video from the PBS Nova series, *The Chip vs. the Chessmaster*. [5] This video gives a brief history of game playing programs, Chess in particular, and tells the story of the Chess playing program that eventually beat world champion Gary Kasparov.

Text Compression - An exploration of the techniques used to send information in compressed formats. This lab is based on a very traditional data structures topic, the Huffman tree. In addition to programming work, students develop mathematical proofs of optimality. Here again, students see how mathematical theory serves as an aid in some very practical computer science problems and how mathematical theory is used to formalize computer science. Other aspects of Information Theory are discussed including error-correction/detection, image compression, and applications to the online music-sharing craze. Students compared their own compressed versions of *Hamlet* and the *Declaration of Independence* to the files produced by WinZip.

A Mathematical Card Trick - A computational exploration of a card trick. A rather slick card trick gives rise to some interesting mathematical questions. Some of these questions can be answered by pure discrete mathematics, and others require experiments by way of programs. The programs lead to the discovery of theorems otherwise not obvious. The programs themselves are an enigma, as they sidestep massive exponential explosion for reasons that elude mathematical analysis. The lab emphasizes the recurring theme of our learning community: that mathematics aids the computer scientist and that programs can help the mathematician. [6]

CONCLUSION

Written, anonymous evaluations of the learning community were universally positive:

- It is cool to see how Discrete and Data Structures come together when doing each lab... I think that this LC was an interesting but [sic] valuable part of our major.
- The expectation of combining the two classes in the LC has been met. Sections of both classes help to solve the assignments in the LC. The assignments are challenging but still interesting.
- I was really not sure what to expect. The only thing I was thinking about was how the professors were going to combine these two topics... I found how math and computer science came together very interesting.
- [What I found most interesting was] how everything relates to math and science and how God may be a mathematician!

Finally, aided by student input and our own imaginations, we plan to introduce one or two new labs in successive iterations of the course. Possible topics for next year include:

- Mathematical proof - can the computer help? The Four Color Theorem vs. Fermat's Last Theorem
- Computer Assisted Exploration of the Combinatorial Card Game "Set"
- Experiments in Algorithms – When Do Constant Factors Matter?
- Dynamic Programming Versus Recursion – Measuring Space/Time Tradeoffs.
- A Computer Tool to Help Solve Recurrence Equations.

Appendix

SOME SAMPLE LABS

The following are excerpts adapted from the materials for the first and last labs of the integrative seminar. All lab material can be accessed at:

http://www.stonehill.edu/compsci/LC/homepage.htm

Lab 1 -- Chaos, fractals, and computer graphics

Students were expected to read the play "Arcadia" prior to the first meeting.

Introduction

Recently, mathematics has been the backdrop to some very successful plays and films including *A Beautiful Mind*, *Good Will Hunting,* and the Pulitzer Prize winning play *Proof.* In Tom Stoppard's hilarious and award winning play, *Arcadia*, mathematics is not only a backdrop but also a major player. In a British manor house, during both the nineteenth century and the present day, and sharing the stage with sex and satire, fractal geometry, chaos, thermodynamics, and even Fermat's last theorem play starring roles in Stoppard's captivating, yet sometimes inscrutable, comedy.

Since its premiere in 1993, mathematicians have been fascinated with *Arcadia*. Indeed, *Arcadia* was the only play ever to be reviewed in *Scientific American.* [7] Allyn Jackson, in a review published in *Notices of The American Mathematical Society*, noted that "Stoppard has understood something of the poetic heart of this area of mathematics" and praised the playwright for "the thoroughness with which [he] has integrated these mathematical ideas into the action of the play."[8]

For this lab, we'll use "Arcadian mathematics" as our starting point. We will touch upon fractal geometry and chaos and demonstrate how computers and technology have helped these fields grow and mature. There are many websites that explore the mathematics and Arcadia. One of the best is Professor Robert Devaney's site at http://math.bu.edu/DYSYS/arcadia/introduction.html.

The play begins in a large country estate in Derbyshire, England. The time is April 1809. Recall that in the opening scene, thirteen-year-old Thomasina comments to her tutor:

> ...Each week I plot your equations, dot for dot, x's against y's in all matter of algebraical relation, and every week they draw themselves as commonplace geometry, as if the world of forms were nothing but arcs and angles. God's truth, Septimus, if there is an equation for a curve like a bell, there must be an equation for one like a bluebell, and if a bluebell, why not a rose? ..... Why do your equations only describe the shapes of manufacture? Armed thus, God could only make a cabinet.

Thomasina's remarks anticipated a new field of mathematics called fractal geometry. Fractal geometry has been able to model complex objects like leaves, clouds, ferns, mountains, or even the coastline of England. – objects that Euclidean geometry's lines, circles and spheres could not explain. Fractal geometry has ushered in a new way of looking at nature. Fractals have found their way into the realms of abstract art. Fractal images have even been used in popular films: *Star Trek II* used fractal images to create computer-generated images of outer space.

Fractals are detailed, intricate objects. One noticeable property of a fractal image is "self-similarity," a characteristic described in *Arcadia* by Valentine:

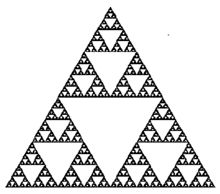
> The left hand pages are graphics of what the numbers are doing on the right hand pages. But all on different scales. Each graph is a small section of the previous one, blown up, like you'd blow up a detail of a photograph, and then the detail of a detail, and so on, forever. Or in her case, till she ran out of pages.

On the same subject, Ivars Peterson, in *The Mathematical Tourist*, writes:

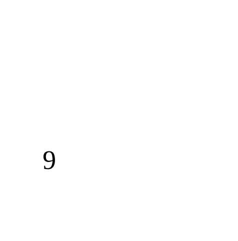
> Fractal objects contain structures nested within one another. Each small structure is a miniature, though not necessarily identical, version of the larger form. The mathematics of fractals mirrors this relation between patterns seen in the whole and patterns seen in parts of the whole. [9]

If the idea seems confusing, a single picture should clear things up.

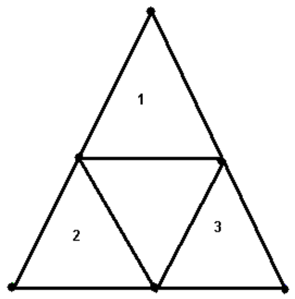
Below is a picture of what is called Sierpinski's gasket.



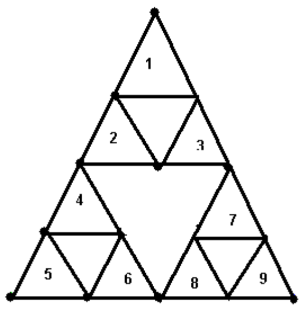
To generate Sierpinski's gasket, begin with an equilateral triangle.



Now, find the midpoint of each side and form three more triangles as in the diagram, discarding the triangle in the center.



Repeat the process.



And continue or iterate... forever, producing 1,3,9,27, 81,243,729....triangles. The Sierpinski gasket is the set of points that remain if the process is carried out indefinitely. Self-similarity is an apparent property of the Sierpinski gasket. (However, what is truly amazing is that *Sierpinski's gasket has zero area.)*

Exercise 1:

Design a program that will draw the Sierpinski gasket. Write it both iteratively and recursively.

Exercise 2:

In the play Valentine remarks:

> You never know where to expect the next dot. But gradually you start to see this shape... The unpredictable and the predetermined unfold together to make everything the way it is.

This exercise gives you an idea of how the unpredictable and the predetermined unfold together - at least mathematically.

The following algorithm is commonly referred to as the Chaos Game.[10]

```
Choose three points to form an equilateral triangle $(x_1,y_1)$, $(x_2,y_2)$ $(x_3,y_3)$.
Pick one of the three vertices at random; call it (x,y)
Repeat
    Choose a vertex, $(x_1, y_1)$, $(x_2,y_2)$ or $(x_3,y_3)$ at random. Label it q.
    Place a point p halfway between (x,y) and q
    Now, let (x, y) equal p
Forever (or until enough points have been drawn)
```

Carry out the algorithm by hand. Describe the figure that the algorithm produced. Explain how the algorithm produced such a figure.

Implement the algorithm as a program.

(*With enough iterations, figure produced is the Sierpinski gasket)*

Exercise 3.

Design and implement an algorithm, similar to the algorithms of either 1 or 2. Your output should be some self-similar figure. Be clever, use your imagination.

In line with Thomasina's discoveries, the students continue with an exploration of the Mandlebrot set and the Julia set. The exercises involve building a C++ class for complex numbers and using that class to generate some very beautiful computer graphics. This lab presented students with their very first exposure to a graphics package (CMU graphics).

Finally, students develop programs that explore chaos, iterative algorithms, and the logistic equation, as well as the butterfly effect, the tendency of a system to be very sensitive to initial conditions. ("If a butterfly is flapping its wings in London, could it cause a hurricane in Miami?")

Lab 5 -- A Mathematical Card Trick

Introduction

In this lab, you will write several programs to discover a few secrets behind a rather clever card trick – a trick that illustrates many of the fundamental concepts we've studied in discrete mathematics.

Here's the trick:

> A magician asks an unsuspecting observer to randomly choose five cards from a standard deck. The participant does not show these cards to the magician, but does show them to the magician's accomplice. The accomplice looks at the five cards, chooses four of them, and shows these four to the magician. The magician immediately identifies the fifth, hidden card.

How does the trick work? The secret is that the accomplice orders the four cards which he/she chooses to show the magician. Nonetheless, the trick seems impossible, because there are only 4! = 24 ways to order four cards, and 24 pieces of information is certainly not enough to identify a unique value among the 48 remaining cards. However, a more careful analysis reveals that the accomplice has, in fact, more than 24 choices. The accomplice may choose *which* four cards to show, as well as the arrangement of the four cards.

Since there are C(5,4) = 5 ways of choosing a group of four cards, and 24 ways to permute each group, the accomplice has 120 different pieces of information which he/she can pass to the magician. The hard part is that the magician must instantly reveal the hidden card from a single bit of information chosen from a pool of 120 possibilities. Certainly, only a well-rehearsed, seasoned performer could manage this feat in just a few seconds.

Let's examine the strategy a little more closely. The magician can easily decode 24 pieces of information by just examining the arrangement of the four visible cards. The team could agree in advance on a numbering scheme for the 24 permutations. For example, the permutations might be ordered 1234, 1243, 1324, 1342, etc in numerical order. Also, each card might be assigned a number from 1 to 52. The ordering of the four cards say 3, 7, 51, 43 would correspond to the permutation 1243 and hence the number 2.

Using this simple encoding/decoding scheme, the trick could be done with a 28-card deck. The accomplice simply communicates a number from 1 to 24 using a permutation of the four cards. This number identifies which of the remaining 24 cards is hidden. Twenty-four is certainly a lot more manageable than 120. Still, the trick takes some fast thinking on the part of the magician.

Program 1. Write a program that helps the magician and accomplice encode and decode n! permutations. For decoding, the input is a permutation of {1,2,3...n} and the output is an integer k. For encoding, the input is an integer and the output is a permutation.

For example, if n = 3, the six permutations of {1,2,3} in order are

1. 123
2. 132
3. 213
4. 231
5. 312
6. 321

For decoding: Input: 231; Output 4

For encoding: Input: 3; Output: 213

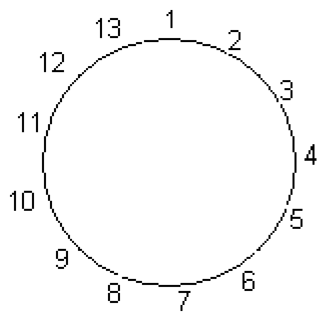
Program 2. Write a program that simulates the trick for 28 cards. Given a set of 5 cards from {1, 2, … 28}, the program should construct an ordered set of four cards. (Note: This ordered set is not unique, and your program will need to choose one) On the other hand, given an ordered set of four cards in {1, 2, … , 28} the program should calculate the unique set of five cards, and thereby the missing card.

Of course, the magician and the accomplice don’t have your programs to help them. Is there a better tactic? In the following discussion, we develop a strategy that will not only take a little less thinking on the magician’s part but also utilize a standard 52-card deck.

Our Working Strategy

First of all, notice that in any hand of five cards there must be two cards of the same suit (pigeonhole principle). The *first* card that the accomplice shows to the magician is one of these two cards. The other card of the same suit is never shown – it is the mystery card, the card that the magician must discover. Thus, the accomplice can easily communicate the suit of the hidden card: the hidden card has same suit as the first card shown to the magician. Specifying the rank of the mystery card is a little trickier but can be accomplished with a little "circular counting."

Number the cards in a suit circularly from 1(ace) to 13 (king) so that 1 follows 13 i.e. the list is ordered in a clockwise direction.



Now, given any two cards A and B, define *distance (A,B)* as the clockwise distance from A to B. It is easy to see that for any two cards A and B either distance(A,B) or distance(B,A) must always be less than or equal to 6.

Example

Cards: 3 and Jack (11)

distance(Jack, 3) = 5; distance (3, Jack) = 8

Cards: Ace(1) and 7

distance (Ace, 7) = 6; distance (7,Ace) = 7

Our working strategy proceeds as follows. From those two cards of the *same* suit, A and B, the accomplice shows the magician card A such that distance(A,B) is 6 or less. For example, given the choice between the three of clubs and the Jack of clubs, the accomplice reveals the Jack since distance (Jack ,3) = 5 and distance(3, Jack)= 8. The three of clubs remains hidden. If the two same-suit cards are the five of hearts and the six of hearts the accomplice chooses the five (since distance(5,6) = 1 but distance(6,5) = 12) leaving the six of hearts as the mystery card.

Finally, the accomplice arranges the last three cards to encode a number from 1 to 6 – the distance from the rank of first card to that of the hidden card. A quick calculation allows the magician to discover the rank of the mystery card. Notice that the magician must decode only one of 6 possibilities – not 120, not even 24. Six possibilities should not present a problem, even to the slowest of magicians.

Example:

Suppose the five cards chosen are 3H 5S 6C 7H 2D

The accomplice notices that the 3 and the 7 have the same suit-- hearts. Since the distance(3 ,7) = 4 and distance(7, 3) = 9, the accomplice chooses the 3 as the first card to show the magician, leaving the 7 of hearts as the hidden card. The magician now knows that the suit of the mystery card is hearts.

The accomplice now arranges the other three cards in the order 5 6 2 – representing the permutation 231. But, 231 is the *fourth* permutation in numerical order. (123 132 213 **231** 312 321). Thus, our fast-thinking magician immediately knows that the hidden card is the fourth card following the 3 of hearts, i.e., the 7 of hearts. Note that the cards are can be ordered from A (low) to K (high), where two cards of the same value are ordered alphabetically by suit: clubs, diamonds, hearts, spades. So that the 4D, for example, is larger than the 4C.

Other examples can be found at http://www.stonehill.edu/compsci/Japan.htm.

The Challenge

The problem we will next attack is whether or not we can perform this trick with a larger deck of cards. The lab continues to explore lower bounds and upper bounds on the number of cards for which it is possible to the trick. It concludes with

- a combinatorial proof that 124 cards is the upper bound, and
- a fascinating application of Hall's Matching Theorem on graphs that implies the lower bound is also 124 cards.

Unfortunately, the mathematics of the latter result is an existence proof, and too abstract to provide a constructive method of actually doing the trick for 124 cards. We conclude with a clever program to implement a constructive method of the proof.

Interested readers are referred to [11] for more details

REFERENCES

1. Gabelenick, E., MacGregor, J., Mathewes, R.S., and Smith, B.L. (eds.), *Learning Communities: Creating Connections among Students, Faculty and Disciplines.* New Directions for Teaching and Learning, no. 41, Jossey-Bass, San Francisco, 1990.

2. Stoppard, Tom. *Arcadia*. Faber and Faber Inc., London and New York, 1993.

3. *Mathematics in Arcadia: Tom Stoppard in Conversation with Robert Osserman.* Mathematical Sciences Research Institute, Berkeley, CA., 2000, videocassette

4. Whitemore, Hugh. *Breaking the Code,* dir. Herbert Wise, WGBH, Boston, 2001, videocassette

5. *The Chip vs. the Chessmaster.* WGBH/Nova, Boston. 1991, videocassette

6. Simonson, Shai and Holm, Tara S. "A Combinatorial Card Trick." *Proceedings of the Eighth International Conference on Human Technology,* (March 14-20, 2002). Fukusima, Japan.

7. Beardsley, Tim. "Sex and Complexity." *Scientific American*, 277, July 1997, 98.

8. Jackson, Allyn. "Love and the Second Law of Thermodynamics: Tom Stoppard's *Arcadia." Notices of the American Mathematical Society* 42, November 1995,1284-1287.

9. Petersen, Ivars. *The Mathematical Tourist: Snapshots of Modern Mathematics*. W.H. Freeman and Co. New York. 1989. |

10. Barnsley, M. F. and Rising, H. *Fractals Everywhere, 2nd ed.* Boston, MA: Academic Press, 1993.

11. Simonson, Shai and Holm, Tara. "Using a Card Trick to Teach Discrete Mathematics." *Primus.* 13:3, September 2003, 248-269.