Teaching Puzzle-based Learning: Development of Transferable Skills

Nickolas Falkner, Raja Sooriamurthi and
Zbigniew Michalewicz

Abstract. While computer science and engineering students are trained to recognise familiar problems with known solutions, they may not be sufficiently prepared to address novel real-world problems. A successful computer science graduate does far more than just program and we must train our students to reach the required levels of analytical and computational thinking, rather than hoping that it will just 'develop'. As a step in this direction, we have created and experimented with a new first-year level course, Puzzle-based Learning (PBL), that is aimed at getting students to think about how to frame and solve unstructured problems. The pedagogical goal is increase students' mathematical awareness and general problem solving skills by employing puzzles, which are educational, engaging, and thought provoking. In this paper we continue sharing our experiences in teaching such a course. Whereas a brief discussion on our pedagogical objectives were covered in the first paper together with the material of the first of two lectures on pattern recognition, this follow-up paper presents the material of the second of two lectures, in which additional exercises are discussed to reinforce the lesson. Along the way we provide a glimpse of some foundational ideas of computer science such as incomputability and general system development strategies such as incremental and iterative reasoning. This paper discusses the outcomes of PBL courses, which include expected improvement in the overall results achieved by students who have undertaken PBL courses, compared to those students who have not.

Key words and phrases: puzzle-based, computational thinking, problem-based.

ZDM Subject Classification: A20, B50, B70, D40, D50.

Introduction

In (Falkner et al. 2010) and (Falkner et al. 2011) we have identified goals for developing learning in a puzzle-based, domain-free, context, and presented

a plan for a course that developed puzzle-solving skills in a variety of contexts. To review the core points, puzzles were used in order to teach problem-solving, because:

(1) *Independence:* The puzzles are not specifically tied to a single problem-solving domain.

(2) *Generality:* Educational puzzles should explain some universal mathematical problem-solving principles. This is of key importance. Most people agree that problem solving, like any other skill, can only be learned by deliberate practice, i.e., by solving problems. However, this activity must be supported by strategies provided by an instructor. These general strategies would allow for solving new yet unknown problems in the future.

(3) *Simplicity:* Educational puzzles should be easy to state and easy to remember. This is also very important, as easy-to-remember puzzles increase the chance that the solution method, including the universal mathematical problem-solving principles, is also remembered.

(4) *Eureka factor:* Educational puzzles should initially frustrate the problem-solver, but with the promise of resolution. A puzzle should be interesting because its result is counter-intuitive: problem-solvers often use intuition to start their quest for the solution and this approach can lead them astray. Eventually a *Eureka!* moment is reached (Martin Gardner's *Aha!*), when the correct path to solving the puzzle is recognized.

(5) *Entertainment factor:* Educational puzzles should be entertaining and engaging. Entertainment is often a side-effect of simplicity, frustration, the Eureka factor, and an interesting setting.

We refer to Figure 1, to reinforce the goals of building a strong foundation for critical thinking and logical reasoning, in order to develop mature, capable professionals who can manage the rigorous demands on project-based thinking and large-scale team-based problem solving.

For this paper we have selected one of the topics covered in the whole course (topic #11: pattern recognition), and presented the outline of a first lecture on this topic that engaged student interest in the domain of discourse and provided us with new puzzles. It also introduced some early techniques for solving puzzles that rely upon pattern recognition.

Becoming a good puzzle-solver is, plainly, not enough to have recognised value in terms of academic achievement or professional recognition - the skills that are taught in Puzzle-Based Learning must be able to be transferred into more applied
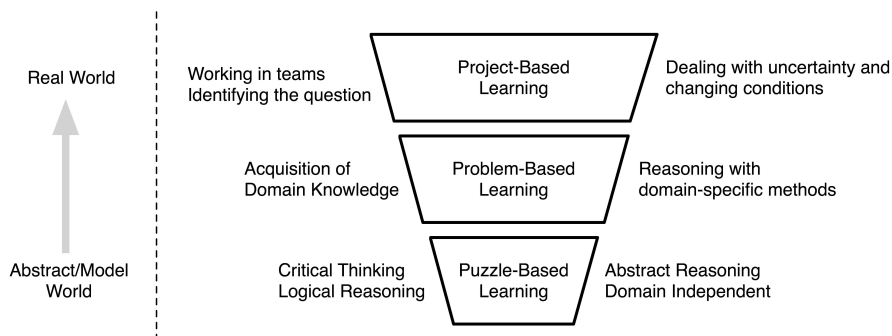
Real World

Abstract/Model
World

Working in teams
Identifying the question

Project-Based
Learning

Dealing with uncertainty and
changing conditions

Acquisition of
Domain Knowledge

Problem-Based
Learning

Reasoning with
domain-specific methods

Critical Thinking
Logical Reasoning

Puzzle-Based
Learning

Abstract Reasoning
Domain Independent

Figure 1. A continuum of learning and skills needed for problem solving in the real world. In this continuum, each layer of skills builds upon the layers below it. The focus of puzzle-based learning is on domain independent, transferable skills. In addition, we aim to foster introspection and reflection on the personal problem solving process. What was I thinking? What is the solution? Why did I not see it?

domains. To facilitate the skills transfer, we provide a second lecture on the same material to allow students to develop heuristics or thinking frameworks that can be separated from individual puzzles and applied more generally.

Why do we focus on the transferability of the skills? We are concerned with producing mature problem-solvers who can move into the workforce and apply their skills to which domain they find themselves in. Industry values good problem solvers (Poundstone, 2000), and we consider the development of a sound thinking and solving methodology to be a critical part in the education of a computer science graduate. However, by focussing on the puzzle, rather than limiting ourselves to the computer science, or engineering, domains, we frame the solving techniques and approaches in a way that they can be applied anywhere.

We do not know what kind of world our graduates will be working within, in 10 years time. If we train their thinking in a way that does not depend on any particular context then, as the world changes, they can adapt their approach and still be successful, reasoning, problem solvers.

The first lecture introduces the topic, generates interest among the students, and begins the solution development process. The second lecture goes beyond this and, once students have dealt with the concepts in this lecture, they are more able to identify the puzzle inherent in a problem, and the associated problems in a project. This ability to identify and then apply the solution technique has

brought the PBL approach out of the abstract and academic basis and provided a transferable skill.

Methodology

Ideally, we would be able to isolate groups of students, expose one to Puzzle-Based Learning and deprive another and then quantify the differences in their performance across an otherwise identical program, where age, gender, previous education, nationality and all other variables were sufficiently well represented that we could then clearly identify that it was only our program that enhanced their skills. This is, obviously, not possible.

In our first paper on this topic, Lecture part 1 (Falkner et al. 2012), we presented our approach and a set of simple puzzles that encouraged participation and mathematical awareness. However, it would be a grand stretch to say that exposure to the Tennis Twins puzzle was enough to inculcate a new approach to problem solving that could be applied elsewhere.

In this second half of the lecture, we demonstrate those puzzles that have deceptively straight-forward solutions, apparently have no solution or where a brute-force solution will quickly lead to frustration. It is in these puzzles that we start to exercise the mind in ways that can then bring direct benefit to other areas, as we will identify in annotations to each puzzle. In order to determine if this spectrum of increasing difficulty has had any effect, we then carry out a simple analysis of the results of those students who did and did not undertake PBL as part of their first year preparation. However, we do not make a strong claim on the outcome, because of our difficulty in isolating the impact.

A Sample Lecture - Part 2 of 2

We start the second lecture on pattern recognition by discussing an interesting puzzle found in (Stewart, 1999) - this puzzle enjoys enormous popularity in the class environment.

PUZZLE 5. *Ten pirates have plundered a ship and discovered 100 gold pieces. They need to divide the loot among themselves. They want to be fair and abide by the law of the sea: to the strongest go the spoils. So they have an arm-wrestling match to determine how strong each pirate is and then sort themselves*

from weakest to strongest. No two pirates are equally strong so there is no doubt about the order. We can label the pirates from weakest to strongest as P1, P2, and so forth, up to P10. The pirates also believe in democracy, and so they allow the strongest pirate to make a proposal about the division, and everyone votes on it, including the proposer. If 50 percent or more of the pirates vote in favour, then the proposal is accepted and implemented. Otherwise, the proposer is thrown overboard, into the shark-infested waters, and the procedure is repeated with the next strongest pirate.

All pirates like gold, but they hate sharks even more than they like gold. So any one of them would rather stay onboard the ship and get no gold than be thrown overboard to the sharks. All the pirates are rational, and they know that if they damage any of the gold pieces (e.g. by trying to divide them into smaller pieces, then the bullion will lose almost all of its value). Finally, the pirates cannot agree to share pieces as they do not trust each other.

What proposal should the strongest pirate propose to get the most gold?

Iterative and Incremental Reasoning

Sometimes, the best way to analyse different strategies is to start from a smaller instance of the problem as at that stage it may be obvious which strategy is good and which is bad. Then we can transfer this knowledge to the next-to-last decision, and then continue in succession because each of these strategic decisions is centred on the same question: What will the next pirate do if I do this? For any n pirates, the question is based on the decision for $n-1$ pirates. In this way we can gradually grow the solution.

Aside: As we mentioned earlier puzzle-based learning is a means to the end of being good general solvers in the context of problems and puzzles. This notion of starting small and growing a solution is a prevalent methodology of software development termed iterative and incremental development. We use this puzzle to emphasize the bridge upwards in our puzzle-problem-project continuum given in Figure 1.

So, let us start at the end, when there are just two pirates, P1 and P2. In that case the strategy of the strongest pirate, P2, is obvious: propose 100 gold pieces for himself, and none for P1. His vote would carry 50 percent of the vote necessary for the acceptance of the proposal and he would be one rich pirate!

Now we can consider the case with three pirates. Note that pirate P1 knows (and P3 knows that P1 knows!) that if P3's proposal is turned down, the procedure would proceed to the two-pirate stage where P1 gets nothing. So P1 would

vote for absolutely any proposal from P3 that gets him *something*. Knowing then that the optimal strategy for P3 is to use a minimal amount of gold to bribe P1 to secure his vote, P3 should propose 99 gold pieces for himself, 0 for P2, and 1 gold piece for P1.

The strategy of P4 in the scenario with four pirates is similar. As he needs 50 percent of the vote, he needs a vote of one additional pirate. Again, he should use a minimum amount of gold to secure this vote, so his proposal is 99 gold pieces for himself, 0 for P3, 1 gold piece for P2, and 0 for P1. Of course, P2 would be happy to vote for this proposal; otherwise P4 is thrown overboard, the procedure reduces to three pirates, and P2 gets nothing.

Now, the strategy of P5 in the scenario of five pirates is just slightly different. He needs two additional votes from his fellows. Thus he proposes 98 gold pieces for himself, 0 for P4, 1 gold piece for P3, 0 for P2, and 1 gold piece for P1. Clearly, the votes of P3 and P1 are secure, because in the four-pirate scenario they would get nothing.

It is straightforward now to design a proposal for P6 in a six-pirate scenario, for P7 in a seven-pirate scenario, etc. In particular, the proposal for P10 is: 96 gold pieces for himself, 1 gold piece for each of the pirates P8, P6, P4, and P2, and none for the rest. This solves the small version of the puzzle. It is good to be the strongest pirate, at least when there is a small number of pirates and a lot of gold.

Scaling the Solution Process

Now, let us move to the larger version of this puzzle, leaving all the assumptions as they were but increasing the number of pirates to 500. The same pattern emerges, but there is a catch, because it only works only up to the 200th pirate. P200 will offer 1 gold piece for himself, 1 gold piece for each even-numbered pirate, and none for the rest. And that is when the fun starts in this larger version of the problem.

P201 still can follow the previous strategy except that he runs out of gold and he proposes nothing for himself. So he proposes 1 gold piece for each odd-numbered pirate from P199 to P1. In that way he gets nothing but at least he stays on board and avoids being eaten by sharks.

P202 also gets nothing. He has to give all 100 gold pieces to 100 pirates and stay dry. The selection of these pirates is not unique, as there are 101 pirates who are willing to accept the gold (pirates who do not get anything in the 201-pirate scenario), so there are 101 ways to distribute these bribes.

What about the 203-pirate scenario? This pirate must get 102 votes for his proposal including his own vote and he does not have enough gold pieces to give to 101 of his fellow pirates. So P203 will go overboard regardless of what he proposes! Too bad for him.

This is important for P204 though, as he knows that P203 would vote for anything to save his life! So P204 can count on P203 no matter what he proposes. That makes his task easy, as he can count on P203, himself, and 100 fellows that get a gold piece each, so he can secure 102 votes. Again, the recipients of the gold should be among the 101 pirates who would receive nothing under P202's proposal.

The pirate P205 in the 205-pirate scenario faces an impossible task. He cannot count on P203 or P204 for support: each will vote against him to save themselves. So P205 will be thrown overboard no matter what he proposes. The moral is: do not be the strongest in a group of 205 democratic pirates. The same fate awaits P206: he can be sure of P205's vote, but that is all he can count on, so overboard he goes. Similarly, P207 faces a soggy end to his existence, as he needs 104 votes: his own, 100 from the gold, and 3 additional followers. He can get votes from P205 and P206, but these are not enough. . . so overboard he goes.

The fate of pirate P208 is different, as he also needs 104 votes, but P205, P206, and P207 will vote for him to save their lives! With his own vote and 100 votes, his proposal will be accepted and he will survive. Of course, the recipients of his gold must be among those who would get nothing under P204's proposal: the even-numbered pirates P2 through P200, and then P201, P203, and P204.

Now, we can see the pattern, which continues indefinitely. Pirates who are capable of making successful proposals (even though they get no gold from their proposals, but at least they get to stay on the ship) are separated from one another by ever longer sequences of pirates who would be thrown overboard no matter what they propose! So the pirates who can make a successful proposal are P201, P202, P204, P208, P216, P232, P264, P328, P456, and so on (i.e. pirates whose number equals 200 plus a power of 2).

It is also easy to see which pirates receive the gold. As we saw before, the solution is not unique, but one way to do this is for P201 to offer gold to the odd-numbered pirates P1 through P199, for P202 to offer gold to the even-numbered pirates P2 through P200, for P204 to the odd numbers, for P208 to the even numbers, and so on, alternating between even and odd.

So, as the puzzle clearly illustrates, being the strongest and having a chance to put forward the first proposal is not always the best (unless, of course, the

number of pirates is quite small; sometimes it is good to be a big fish in a small pond!).

Transference

In a class room setting we have used this example to illustrate the power of iterative and incremental reasoning in general and the nature of such reasoning in software development. The transferable knowledge from this particular example includes both the possibility of obtaining a solution at all, the strongly counter-intuitive nature of the result (which has shades of Total Utilisation games and Nash Equilibria) and the requirement of a divide-and-conquer strategy that is an excellent learning point for any mathematician or computer science.

In-Computability - a Puzzle Based Glimpse

The final puzzle on pattern recognition presented in the standard version of the puzzle-based learning course is the following.

PUZZLE 6. *A successful businessman opened a retirement account by depositing a certain amount of money (some dollars, no cents) just before the end of a year. He made another deposit (again, some dollars, no cents) one year later. From that time, towards the end of each year he deposited some amount of money. His deposits followed a pattern: From the third year onwards, the amount of each deposit was equal to the total of two previous deposits (from two previous years). His 20th deposit was $1,000,000 exactly.*
What was his initial deposit?

Clearly, the first few steps in approaching the problems are easy. If we introduce two variables, x and y, for deposits made in the first and second year, respectively, then we can generate the whole sequence of 20 deposits made over 20 years:

- 1st year: x
- 2nd year: y
- 3rd year: $x + y$
- 4th year: $x + 2y$
- 5th year: $2x + 3y$

- 6th year: $3x + 5y$
- 7th year: $5x + 8y$
- 8th year: $8x + 13y$
- 9th year: $13x + 21y$
- 10th year: $21x + 34y$
- 11th year: $34x + 55y$
- 12th year: $55x + 89y$
- 13th year: $89x + 144y$
- 14th year: $144x + 233y$
- 15th year: $233x + 377y$
- 16th year: $377x + 610y$
- 17th year: $610x + 987y$
- 18th year: $987x + 1597y$
- 19th year: $1597x + 2584y$
- 20th year: $2584x + 4181y$

This would lead immediately to one equation with two unknowns:

$$2584x + 4181y = 1000000$$

The question is: How to solve it?

This is a good moment to involve (again) students in some discussion. It might be also a good moment to tell the students that the above equation constitutes an example of a diophantine equation (named after Diophantus of Alexandria, a famous Greek mathematician of antiquity), in which we have fewer equations than unknowns and the values of the unknowns are restricted to integer values. Some diophantine equations have no solution, some have a finite number, and some have infinitely many.

If there are computer science students attending the lecture, it might be worthwhile to note that diophantine equations played a hand in the birth of computer science. On 8th August 1900, at the International Congress of Mathematicians in Paris, the German mathematician David Hilbert gave a list of twenty-three mathematical problems.[1] The problems were all unsolved at the time, and several of them turned out to be very influential for 20th century mathematics.

---

[1]Actually, Hilbert presented ten of the problems (1, 2, 6, 7, 8, 13, 16, 19, 21 and 22), and the full list of twenty-three was published later.

One of these twenty-three problems (the tenth) was to find a "finitary way" to solve any diophantine equation. Hilbert's "finitary way" was what we call today "an algorithm." The term *algorithm* was formalized years later by Alan Turing (and the formalized notion of what an algorithm is has come to be called a *Turing Machine*). Finally, in 1970, Jurii Matijasevic, a twenty-two year old student at the University of Leningrad, proved that Hilbert's tenth problem is computationally unsolvable - no such algorithm can exist. But Hilbert's tenth problem was very significant at that time as he wanted a "finitary way" to solve any mathematical problem... he wanted something he did not have a name for, he wanted an algorithm. So on the morning of 8th August 1900, marked the start of computer science! It is a shame that David Hilbert never knew it.

A Simpler Similar Problem

Before solving the Diophantine equation

$$2584x + 4181y = 1000000$$

posed by Puzzle 6, we turn the students' attention to a different problem which seems unrelated to the diophantine equation at hand, and which was published in the early 13th century in Italy (actually, it was published in 1202 in a book titled *Liber abaci* by Leonardo of Pisa, who was also known as Fibonacci):

PUZZLE 7. *Beginning with a pair of rabbits (one male and one female) born on January 1*ˢᵗ*, assume that:*

1. *The rabbits begin to produce offspring two months after their own birth.*
2. *After reaching the age of two months, each pair produces a mixed pair (i.e. one male and one female), and then another mixed pair each month thereafter.*
3. *No rabbit dies.*

How many pairs of rabbits will there be after 12 months?

The solution is the famous Fibonacci sequence:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 \ldots$$

So at the end of the year (after 12 months) there will be 144 pairs of rabbits, all resulting from the original pair born on January 1 of that year. Of course, we should assume that the reproductive capabilities of the rabbits are not influenced

by shorter or longer months of 28 or 31 days. In this puzzle, each month is a "unit of time."

Each number in the Fibonacci sequence is called a *Fibonacci number*. It is clear from the description of the problem that each Fibonacci number is obtained by adding the two previous Fibonacci numbers together. For example, the next Fibonacci number can be obtained by adding 144 and 89. Thus, the next Fibonacci number is 233 (this would be the number of pairs of rabbits after 13 months).

The Fibonacci numbers form a sequence defined as follows (it is customary these days to start the sequence from zero):

$$F(0) = 0, \quad F(1) = 1, \text{ and then } F(n) = F(n-1) + F(n-2).$$

In other words, after the two starting values, each number is the sum of the two preceding numbers. As we start with 0 and 1, we can easily determine the numbers in this sequence:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811. . .

The Fibonacci numbers were made *really* popular by Dan Brown in his novel *The Da Vinci Code* and are still considered by some to be "mysterious" even today. One of the most fascinating things about the Fibonacci numbers is their connection to nature: the growth of buds on trees, the pinecone's rows, the starfish, the petals on various flowers, iris, buttercup, daisy, and the sunflower, the appendages and chambers on many fruits and vegetables such as the lemon, apple, chile, and the artichoke. The Fibonacci numbers are also present in art and architecture; they are useful in studying rhythmic patterns that are formed from one-beat and two-beat notes or syllables. They are present in many different areas of mathematics. For example, the Fibonacci number $F(n+1)$ gives the number of ways for 2×1 dominoes to cover a $n \times 2$ board. Other amazing properties of Fibonacci numbers include the following. One can prove that any natural number greater than one can be expressed as a sum of Fibonacci numbers, e.g. $10 = 2 + 8$, $19 = 1 + 5 + 13$, or $26 = 5 + 21$. Further,

- $F_1 + F_2 + F_3 + \ldots + F_n = F_{n+2} - 1$

- $F_1 + F_3 + F_5 + \ldots + F_{2n-1} = F_{2n}$

- $F_1^2 + F_2^2 + F_3^2 + \ldots + F_n^2 = F_n \times F_{n+1}$

to list a few. However, one of the most amazing properties of Fibonacci numbers is their connection with the *golden ratio*.[2] Let us explain.

At least since the Renaissance, many artists and architects have proportioned their works to approximate the golden ratio (especially in the form of the golden rectangle, in which the ratio of the longer side to the shorter is the golden ratio) believing this proportion to be aesthetically pleasing. The golden ratio φ is defined as the ratio that results when a line is divided in such a way that the whole line has the same ratio to the larger segment as the larger segment has to the smaller segment:
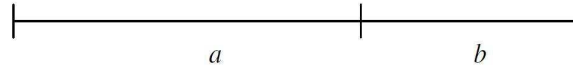
In other words, $(a + b)$ is to a as a is to b:

$$\frac{a + b}{a} = \frac{a}{b}$$

Assuming that the larger part of the segment is one (i.e. $a = 1$), we can build an algebraic expression for the golden ratio:

$$\frac{x}{1} = \frac{1}{x - 1}$$

which is equivalent to $x^2 - x - 1 = 0$. This quadratic equation has only one positive solution:

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

which is $1.6180339887498948482045868344366\ldots$ It is not straightforward to find the closed form (i.e. a formula that only depends on n and not on some previous numbers) of the n-th Fibonacci number. But like every sequence defined by linear recurrence, the Fibonacci numbers have a closed form solution. It has become known as Binet's formula:

$$F(n) = \frac{\varphi^n - (1 - \varphi)^n}{\sqrt{5}}$$

---

[2]The golden ratio, also known as the divine proportion, golden mean, or golden section, is a number often encountered when taking the ratios of distances in simple geometric figures such as the pentagon, pentagram, decagon and dodecahedron. In mathematics and the arts, two quantities are in the golden ratio if the ratio between the sum of those quantities and the larger one is the same as the ratio between the larger one and the smaller. The golden ratio is approximately 1.6180339887 and is denoted φ, or sometimes τ.

where φ is the golden ratio defined above. For computer science and mathematics students, we draw their attention to the similarity between the recurrence relationship for the Fibonacci sequence:

$$F(n) - F(n-1) - F(n-2) = 0$$

and the equation we solved:

$$x^2 - x - 1 = 0.$$

This similarity is a key point for solving problems with general recurrence relationships (such as analysis of algorithms) during university studies - again, this might be of interest to computer science students.

But let us return to our discussion on the connection between Fibonacci numbers and the golden ratio. Apart from Binet's formula discussed earlier, the ratio between the next and the previous Fibonacci numbers approaches φ:

$$F_{n+1}/F_n \to \varphi.$$

We can take a few first non-zero Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21, 34, etc. and calculate the appropriate ratios:

$$
\begin{aligned}
2/1 &= 2 \\
3/2 &= 1.5 \\
5/3 &= 1.666\ldots \\
8/5 &= 1.6 \\
13/8 &= 1.625 \\
21/13 &= 1.615\ldots \\
34/21 &= 1.619\ldots
\end{aligned}
$$

Indeed, these ratios approach $\varphi = 1.618034$.

One can prove that Fibonacci number / (previous Fibonacci number) $\approx \varphi$; this is also true for so-called composite Fibonacci numbers, which, instead of starting from 1, 1 (as it is the case in the classic Fibonacci sequence), start from

arbitrary x and y:

$$x$$
$$y$$
$$x + y$$
$$x + 2y$$
$$2x + 3y$$
$$3x + 5y$$
$$5x + 8y$$
$$8x + 13y$$
$$13x + 21y$$
$$21x + 34y$$

Does it look familiar? With this "discovery" we can return to solving Puzzle 6. As for these composite Fibonacci numbers the ratio between the next and the previous number also approaches φ, our task suddenly is easy. If we assume that the deposit made at the 19th year was Q:

- 19th year: $1597x + 2584y = Q$
- 20th year: $2584x + 4181y = 1000000$

then we also know that

$$1000000/Q \approx 1.618034.$$

So $Q = 618034$ (after slight rounding) and we get two equations with two unknowns (for deposits made at the 19th and 20th years, respectively):

$$1597x + 2584y = 618034$$
$$2584x + 4181y = 1000000.$$

These two equations lead immediately to a solution

$$x = 154$$
$$y = 144.$$

This is a very long discussion of a number of key mathematical issues - what is it that we can call 'transferable' from this extended discussion? We introduce the exceedingly important concept that problems exist for which we know no solution can be found and that the apparently recently delivered discipline of Computer

Science has been hand-in-hand with Mathematics for far longer than most would think. However, we also develop a notion that is found in areas as diverse as the Monadic programming models of Haskell and the promises/futures models of Distributed Computing - the notion of an unresolved place holder upon which we can safely operate until a solution is found that allows everything to be neatly unrolled.

We also reinforce the fundamentals of linear algebra in terms of the Diophantine equations and the requirement to have sufficient specificity to provide a single answer that we know to be right, rather than a family of answers within which the answer may be seen to lurk.

It would be unfair to suggest that it is mere exposure to these aspects that leads to the development of transferrable knowledge - we are explicit in our statement of problems and our expectations of what students are expected to do. There are many informal opportunities, as part of discussions in lectures or tutorials, where we actively reinforce the "freely associated" ideas that rise out of this work and that can be used to solve other problems. We also take a more formal approach to clearly identify how students are using their skills, through the use of regular homework activities.

The final minutes of any of our lecture presentations are spent on a summary of presented material and on discussion of a weekly homework. In the summary part of the lecture we point out that often the more knowledge and experience we have, the easier it is to solve problems. So the professional mathematician is likely to solve these puzzles faster. . . that should not be surprising. The more we know about mathematical principles, various problem-solving techniques, properties of numbers, etc. the easier it will be for us to find the solution. This was also the case in many of Sherlock Holmes' deductions; it is worthwhile to have a closer look at one of his cases and discuss it with the students:[3]

On the return to his apartment at 221B Baker Street, Sherlock Holmes found a pipe left there by a gentleman who came earlier, but who could not stay. While waiting for the return of this unknown gentleman, Sherlock Holmes examined the pipe and made many observations:

> "*The owner is obviously a muscular man, left handed, with an excellent set of teeth, careless in his habits, and with no need to practice economy.*"

---

[3]From Arthur Conan Doyle's story *The Yellow Face.*

Sherlock needed his extensive knowledge (library of facts) to run these types of deductions. On many occasions (e.g. in *The Adventure of the Blue Carbuncle*) he claimed:

> "*It is my business to know what other people don't know.*"

In the case of the pipe owner, he recognized the tobacco in the pipe:

> "*This is Grosvenor mixture at eightpence an ounce. As he might get an excellent smoke for half the price, he has no need to practice economy.*"

Similarly, in many problem-solving activities, we should have at our disposal a library of facts, principles, techniques, and methods to make the problem-solving process easy and enjoyable. Again, the larger our library, the easier it will be for us to find the solution.

Phantom Patterns

Further, we remind students that in the puzzles discussed in this lecture, we have tried to discover patterns to predict (or determine in an analytical way) the next number or symbol. When we search for patterns, we should constantly keep the following questions in mind: How will this help us predict the next symbol or event? How will this help us solve a larger instance of the problem?

It is worthwhile to emphasize this point further by considering the following example: Say, we flip a coin many times and we get some sequence of heads (H) or tails (T):

```
HHTHTTTTHTHHHTTTHTHHTHTHHHTHTTTTTHTHTHTHTHTH
HHHHHHTHTHTHTHTHTHHTHTTTTTTTHHTHTHTHTHHTTTHHHH
HTHTHHTHTHTHTHHTHTHHTHTHTTTTTHTHTHTHTHTHTHTHHT
HTHTHHTHHTHTHHHHHTHTTTTTHTTHHTHTHTHTHHTTTHTH
THHHHTHTHHTHTHHTHTHTHHTHTTTHHTHTHTHTHHTTTHTH
TTTHTHTHHHTHTTTTTTTHHTHTHTHTT
```

We can analyse this sequence and discover some interesting patterns that would be false, in the sense that they will not hold in the future. For example, the above sample may suggest that "every sequence of TTTTTTT is followed by HH" (there are two such occurrences). Such conclusions are meaningless and they follow a rule: *If we torture the data long enough, the data will confess!* However, this happens on a daily basis (e.g. commentators "explaining" the gyrations of the stock market).

Again, this is an excellent point to bring out in lectures, as it provides a brake to runaway, or frantic, attempts to model patterns onto randomness. Not only do students require guidelines as to how to start, but they also need a great deal of help in learning when a given approach has been exhausted. This is an opportunity to encourage discussion of a student's solution process. If a student has no real idea of why they succeed or fail then they cannot improve upon their approaches for their own reward. Our work with the more advanced parts of this course is always associated with explaining, carefully and patiently, to students why a particular answer is a good approach to take or not. In particular, we have to clearly identify when a student has made an 'assumption' that is effectively 'I now solve the problem' by including more than is actually required for the solution.

This highlights one of the most useful transferable skills from this course, a trimmed and contemporary version of Ockham's Razor that one need not use anything to solve a puzzle that is beyond what is required. In one form or another, this regularly surfaces in later vocabulary in the successful and high achieving students of this course.

To test our students, we provided a puzzle with a simple appearance as part of one of the assignments. The students were asked to solve the problem, which had an apparently mathematical solution path but no simple mathematical solution. As it transpired, some thought and pattern recognition allowed the problem to be solved in minutes. Students who refused to give up on the mathematical approach wasted up to hours because they refused to believe that their methods had stalled, even when they started cycling back to previous unsuccessful solutions.

The instructor in puzzle-based learning must be aware that students face constant discouragement when they first attempt puzzle solving. In the face of this, students may attempt to fit a problem-solving technique that they are comfortable with onto every problem. To paraphrase, "when all you have is a hammer, every problem looks like a nail." Part of the benefit of the "torturing data" example is that it shows that a range of approaches to interpreting data can be useful. Ideally, if a student who has graduated from a puzzle-based learning curriculum were presented with the data and asked to spot the pattern, they would use a variety of techniques, count the Hs and the Ts, do some simple analysis and work out the likelihood of this being a straight-forward flipping sequence. With supporting working, an answer revolving around "Heads or Tails, with a 50/50 probability" is both a valid and in-depth answer. We have analysed the data

correctly and solved the puzzle - that there is, in fact, no real puzzle here, more of a mystery.

We can now discuss the solution to the fraudulent accounting problem, raised in the earlier section on pattern recognition. To repeat the question:

"You believe that an accountant has been changing company data to hide fraud. How would you detect this if you cannot compare the original and the altered records?"

This is a case where interrogating the data will result in a successful outcome. By using this question, we can start discussing Benford's Law and Benford's Distribution. Benford's law states that in many lists of data from real sources the leading digit is not distributed uniformly as one would expect but has a specific and non-uniform distribution. The first digit is '1' approximately 30% of the time, with '9' occurring less than 5% of the time. Large-scale accounting data is one area where Benford's Law applies and can be used to assess whether the pattern of numbers in the data match what is expected.

Benford's Law works as a detection tool because most fraudulent data is fabricated randomly to conceal or replace other data. By choosing random and uniformly distributed digits, the fraudulent account is not concealing their activities but highlighting them.

Talking about random sequences and patterns (as the one above), it is hard to resist another observation: Most people tend to think that the number of heads (H) and tails (T) in the long run should be the same. In other words, the deviation of the number of heads (H) and tails (T) from the expected values (occurrence of 50%) in the long run is smaller and smaller. They interpret this as a prediction: the larger the deviation at any point, the greater the restoring force toward the expected value. We can see this clearly during a roulette game: after seven "reds" many players bet on "black"...

This however, is not the case. The point is that the ratio between the number of heads and tails does get closer to 1 in the long run, but not in absolute numbers. For example, after 100 flips we may get 53 heads to 47 tails, and after 1000 flips: 523 heads to 477 tails. Then:

$$53/47 \approx 1.1277 \text{ and } 523/477 \approx 1.0964$$

so the ratio does indeed get smaller, whereas the difference in absolute numbers:

$$53 - 47 = 6 \text{ and } 523 - 477 = 46$$

is getting larger! It is important not to confuse these two phenomena.

There is another surprising phenomenon connected with pattern recognition and the number of consecutive heads or tails that appear in coin flips: There are well-known cases of mathematicians who entertain their students by asking someone to generate an artificial (i.e. made up) sequence of heads and tails of some length (say, 100). A different student is asked to flip a real coin 100 times and record the results. The teacher then looks at both results and correctly identifies the real coin flips. Mathematicians manage to identify the real coin flips by picking the sequence with the longest streak of either heads or tails! The fake results contain much shorter sequences... If there is still a time during the lecture, one can arrange easily for such experiment...

The last few minutes of a lecture are for presenting (and discussing briefly) a homework (students have usually full week for submitting their solutions). One of our favourite puzzles on this topic is the following:

PUZZLE 8. *Skye, Robin and Royal are three members of a 100-member, one-eyed alien culture. These aliens are very intelligent, but their culture has a curious tradition. If they are ever able to determine the colour of their own eye, which they know is either brown or blue, they must commit suicide in front of the whole group at their daily meeting, at which attendance is mandatory. The aliens have had no contact with the outside world and they have no mirrors or reflecting surfaces (for obvious reasons). Further they never discuss eye colour in any way because they all know what great logical brains they have and so are very hesitant to give away any clues about eye colour. Skye, Robin and Royal are the only blue-eyed aliens. One day, a well-meaning space traveler visits the aliens, and at their morning meeting gets up and makes an effort at good relations by saying "Your people and mine are not so different. For example, I see that you have two different eye colors, just like I do in my own family."*

Do you think that this statement provided some new information to the aliens? Do you think that this announcement "helped" some aliens in determining the color of their eye? What happened after the departure of the visitor? Can you provide with a description of day-to-day development in their community?

Usually, the first reaction of the students is that "*the space traveler has not provided with any new information, as all aliens knew that they have either blue or brown eyes, and all of them could actually see samples of these colors at other aliens...*" Again, we have an excellent opportunity to provide a lesson that can be transferred to many future contexts - "Am I right in assuming that this problem is impossible?" All of our work with large group and specialist PBL teaching has

shown us that people will quite happily stop at the point that they believe the problem is impossible - until someone demonstrates that it is not, at which point they resume. As educators, encouraging our students to persist until they can prove that no such solution exists is essential!

Returning to the Alien puzzle, as in many examples presented during the lecture, the intuition that the puzzle cannot be solved is wrong. As in some other cases, we need to discover a pattern - as to see it clearly, we may consider first a scenario where there is only one blue-eyed alien in the group; this would be followed by analyzing a scenario where there are two blue-eyed aliens in the group - and at this stage we should be ready for the three blue-eyed aliens in the group. In a sense, this is a very similar approach to the one that lead us to a solution for the pirates' puzzle.

Student reports of solving this puzzle clearly show that this is how the successful students solve this problem. They start from a small number and build to a larger. Once they have received the reward of solving the problem, this technique is added to their personal set of puzzle-solving approaches and is often useful in determining the correct solution.

Observations

In general, students enjoy PBL class sessions - very often we get encouraging comments at our evaluation forms, e.g. *"The course topics are interesting and useful"*, *"I think the topic is very interesting. I enjoy coming to class because it is very hands on and allows me to use critical thinking"*, *"This course seems to be expanding my mind by giving me new ways to interpret and solve problems that I would be completely lost on"*, *"I have learned a lot about how to approach problems and the thought processes behind it. My ability to prove mathematical and logic proofs has improved"*, *"I like that the course opens my eyes to independent critical thinking and learning new ways to approach problems"*, *"The way you are taught a new way about attempting problems, helps in all courses, not just this one"*, *"Stimulating abstract way to look at problems. Applicable to all my courses"*, *"The lectures are entertaining and I feel like I'm continuously being challenged to prove my increasing problem solving skills"*, or (one of our favourites) *"It is fun and makes me think"*. Clearly, it seems that the skills that we encourage and develop in PBL are truly useful and transferable.

If this is really the case, then we would expect to see an improvement in the overall results achieved by students who have undertaken our PBL courses, compared to those students who have not.

Within one of our institutions' school of Computer Science, we have taken the average second year results from the two cohorts in those second year courses that were undertaken in the semesters following that course, in order to test this hypothesis. A 'course' at our institution is a 3-unit component of a 24-unit full program. Students would normally undertake 12 units per semester, as four courses.

Table 1. Student results in second year with, and without PBL

	No PBL 09	PBL 09	PBL09 and Pass
Count	275	111	106
Mean	56.5	59.0	60.2
Std Dev	22.9	22.4	21.5
Coef of Var	40.5%	37.9%	35.7%

The students who had not undertaken the PBL course enrolled in a total of 275 second year courses, where those who had undertaken PBL enrolled in 111. (The PBL09 and Pass column indicates the subset of students who enrolled in PBL, and passed it, and then went on to enrol in further courses.)

Although the average second year marks for students who undertook PBL is higher, and higher again for those who passed, the coefficient of variance is quite high. Carrying out a two-tailed t test over the data, at the 0.05 confidence level, does establish that the two populations do have a statistically significant difference in their means.

This, by itself, does not show a degree of transferability between courses, as we must consider the overall enrolment structure. Are a different group of students enrolling in PBL, compared to other courses? We have analysed the enrolment data for the courses and have not identified any significant differences in enrolment profile that would explain a 5% improvement in overall student mark due to the overrepresentation of 'good' students.

However, it is important to remember that the 3-unit Puzzle Based Learning is a voluntary enrolment course and is not currently part of any compulsory programs. Students self-select for this course and the difference in marks may be due to these students being more engaged, or more willing to participate.

Therefore, at this stage, while early indications are that there is a correlation between improved marks and attending PBL, we have no causal relationships that we can derive. There is a compulsory PBL component in Engineering, a 1-unit course, that will eventually provide a longitudinal study over the 4 years of student degrees, but this data will not be available until 2013.

Conclusions

Puzzle-based learning is an experiment in progress and, like any experiment, we are still establishing how to modify our experimental protocol to truly measure the outputs. The goal is to foster general domain independent reasoning and critical thinking skills that can lay a foundation for problem-solving in future course work. As fun as puzzles inherently are, they are just a means to this pedagogical end. Our preliminary experience in different contexts has been encouraging and well received as we continue to explore this approach.

The link between the reasoning involved in solving puzzles and various foundational concepts in computer science helps to reinforce these ideas in entertaining and informative ways. In this pair of papers we have focussed primarily on pattern recognition. In addition to this one topic, over the course of the semester we visit several Computer Science (CS)/ Computational Thinking ideas. As an example consider the following puzzles:

A jail warden meets 23 new prisoners when they arrive. He tells them, "You may meet today and plan a strategy. But after today, you will be in isolated cells and will have no communication with one another. In the prison is a switch room which contains two switches labeled A and B, each of which can be in either the 'on' or 'off' position. I am not telling you their present positions. The switches are not connected to anything. After today, from time to time whenever I feel so inclined, I will select one prisoner at random and escort him to the switch room. The prisoner will select one of the two switches and reverse its position. He can't move both but he can't move none either. Then he'll be led back to his cell. No one else will enter the switch room until I lead the next prisoner there, and he'll be instructed to do the same thing. I'm going to choose prisoners at random. I may choose the same guy three times in a row, or I may jump around and come back. But given enough time, everyone will eventually visit the switch room as many times as everyone else. At any time anyone of you may declare to me, 'We have all visited the switch room.' If it is true, then you will all be set free. If it is false then you will all be killed." What is the strategy they come up with so that they can be free? [Hint: the idea of using state to represent information.]

There are 100 barrels of wine one of which is poisoned. A king would like to determine the barrel with the poisoned wine by serving the wine

to some mice to see which are affected. Even a few sips of the poisoned concoction will cause a mouse to faint. How many mice are needed to determine the tainted barrel? [Hint: the binary number system]

The benefit to the student goes beyond the short course that they take part in, or any course-specific skills, as the lessons learnt may be applied to every other course in which they participate. One of the most crucial steps for any CS student is moving from struggling with syntax and compiler errors, to analysing problems and applying their thinking skills to identify exactly what they have to do. The skills that they learn in puzzle-based learning are far more than games, as we can see when students apply these skills in their algorithmic development, their identification of test cases and their improved understanding of insufficient problem specification.

Our future work is planned to include student surveys in second and third year to gain a qualitative and quantitative measure of how the students perceive the value of PBL. We will also continue to investigate correlations between student performance and PBL, to determine if we are able to make stronger statements about the benefits of attending PBL.

We have a significant body of anecdotal evidence but this must be refined and quantified if we are to establish a causal relationship that PBL does provide a transferable benefit to students.

For more information on the nature of puzzles and the approaches used in puzzle-based learning, readers are directed to the website associated with the text, `www.PuzzleBasedLearning.edu.au`.

References

[1] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial and A. Palincsar, Motivating project-based learning: Sustaining the doing, supporting the learning, *Educational Psychologist* **26**, no. 3&4, 1991, 369–398.

[2] J. D. Bransford, R. S. Sherwood, N. J. Vye and J. Rieser, Teaching thinking and problem solving: Research foundations, *American Psychologist* **41**, 1986, 1078–1089.

[3] M. Danesi, *The Puzzle Instinct: The meaning of puzzles in human life*, Indiana University Press, 2002.

[4] T. C. DeFranco and F. Curcio, A Division Problem with a remainder embedded across two contexts: Children's Solutions in Restrictive vs. Real-World Settings, *Focus on learning problems in mathematics* **19**, no. 2, 1997, 58–72.

[5] N. Falkner, R. Sooriamurthi and Z. Michalewicz, Puzzle-Based Learning for Engineering and Computer Science, *IEEE Computer* **43**, no. 4 (April 2010), 20–28.

[6] N. Falkner, R. Sooriamurthi and Z. Michalewicz, Teaching Puzzle-based Learning: Development of Basic Concepts, *Teaching Mathematics and Computer Science* **10**, no. 1 (2012), 183–204.

[7] A. Fisher, *Critical Thinking: An Introduction*, Cambridge University Press, Cambridge, UK, 2001.

[8] M. Gardner, *Entertaining Mathematical Puzzles*, Dover Publications, New York, 1961.

[9] J. A. Paulos, *Innumeracy: Mathematical Illiteracy and Its Consequences*, Hill and Wang, New York, 2001.

[10] G. Polya, How to Solve It: A New Aspect of Mathematical Method, Princeton University Press, Princeton, 1945.

[11] W. Poundstone, *How would you move Mount Fuji?: Microsoft's Cult of the Puzzle – How the World's Smartest Companies Select the Most Creative Thinkers*, Little Brown and Company, 2000.

[12] I. Stewart, A Puzzle for Pirates, *Scientific American* (May 1999), 98–99.

[13] University of Adelaide: Mathematics Working Party Final Report, internal document, June 2008.

[14] J. M. Wing, Computational Thinking, *Communications of the ACM* **49**, no. 3 (March 2006), 33–35.

NICKOLAS FALKNER
SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF ADELAIDE
ADELAIDE, SA 5005
AUSTRALIA

E-mail: `jnick@cs.adelaide.edu.au`

RAJA SOORIAMURTHI
INFORMATION SYSTEMS PROGRAM
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA 15213
USA

E-mail: `raja@cmu.edu`

ZBIGNIEW MICHALEWICZ
SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF ADELAIDE
ADELAIDE, SA 5005
AUSTRALIA
AND
INSTITUTE OF COMPUTER SCIENCE
POLISH ACADEMY OF SCIENCES AND THE
POLISH-JAPANESE INSTITUTE OF INFORMATION TECHNOLOGY
WARSAW
POLAND

E-mail: `zbigniew.michalewicz@adelaide.edu.au`