Teaching Formal Methods Using Magic Tricks

Paul Curzon & Peter McOwan


PP release date: 25 May 2013

file: WP122.pdf
Teaching Formal Methods using Magic Tricks

Paul Curzon and Peter W. McOwan

School of Electronic Engineering and Computer Science
Queen Mary University of London
Mile End Road, London E1 4NS

Abstract

The cs4fn project (www.cs4fn.org) aims to both enthuse school students about computer science and teach advanced computing ideas. One of the ways we have done this is using magic tricks. We give a highly popular magic show to school students around the UK. We have also written two magic books teaching computing concepts. We describe here how magic tricks can be used to introduce formal method concepts to school children. We illustrate how card tricks can, for example be used to show what an algorithm is, why testing is insufficient, how rigorous argument can reduce the amount of testing, making models of systems, how algorithms can be verified using proof, how algebraic proofs can be used to do this, what invariants are, and how inductive proofs can be used to verify an algorithm.

1 Introduction

Introducing computer science concepts to students in an interesting and exciting way at an early stage is vitally important to generate the interest needed to ensure a pipeline of students taking the subject to university level. This also applies to particular areas of computer science and especially the theoretical side, where it is important that students take mathematics seriously.

Since 2005 we have run a public engagement project called cs4fn (Computer Science for Fun) [5, 7]. Based at Queen Mary, University of London and funded by EPSRC with support from Google, it is an international campaign that presents leading edge research in offbeat ways to excite kids about interdisciplinary computing topics. It combines three physical magazines sent free to schools (cs4fn [12] on computer science, ee4fn [11] on physical computing and electronic engineering and Audio! [9] on audio engineering), special booklets (e.g., on women in computing [8], biology and computing [3] and on Artificial Intelligence [6]), and website. Subscribers to the magazine come from over 80 countries.

We also give a variety of computer science shows in high schools around the UK, presenting to thousands of school students each year. A key aspect of our live shows has been the use of kinaesthetic, ‘unplugged’ computing activities [13]. Magic is one form
of ‘unplugged’ activity and one of our most successful kind of event has been to give
gain shows that illustrate computer science concepts using magic tricks [10]. We have
used magic tricks throughout the magazine, have a whole area dedicated to magic as
part of the webzine (www.cs4fn.org/magic) and have written two magic books [19, 20]
that teach computer science concepts and that are given free to school children. These
have been translated into German, Italian and Welsh. We have also used magic in
teacher professional development workshops both to show how computing concepts can
be taught away from computers and to demonstrate tricks that can be used to teach
specific concepts in schools.

Whilst many school outreach projects focus on naturally appealing subjects such
as robots and animation programming, with cs4fn we have made a point of covering
advanced topics including those from theoretical computer science. For example, we
have written articles on logic and proof using puzzles. We introduce Turing machines
with the idea of building one that people act out. The tape is made of chocolate that
the ‘machine’ eats and replaces as it processes them. We have also used magic tricks
as a way to introduce a range of topics related to formal program verification. In this
paper we describe this approach. In section 2 we overview how magic can be used to
teach computing concepts more generally. We then describe a series of card tricks in
section 3 that gradually introduce the idea of formal verification. In section 4 we draw
conclusions.

2 Teaching computing concepts using magic tricks

There are a wide range of links between magic tricks and computing, that go beyond
the mere use of technology to make a trick work. In essence any magic trick consists of a
secret method and a presentation. Unless both work, the magic trick will be ineffective.
Similarly, a computer program consists of an algorithm and the interface/interaction
design. This leads to a natural metaphor that can be used to introduce ideas about
computing from programming to HCI using magic.

The link is in fact deeper than this. Many magic tricks are self-working. They do
not involve sleight-of-hand or hidden mechanisms. Provided you follow the steps the
(surprising) magical effect occurs. Self-working tricks are thus precisely algorithms for
doing magic. In fact the algorithms of some tricks are based on the same algorithm as
is used in computing applications. A notable example is that of perfect shuffles [21] in
which the cards are perfectly interleaved one after another from a cut pack. They can
be used to move cards to any position within the deck based on a manipulation of a
binary representation. This trick formed the basis of a patent for a method of moving
data around computer memory.

Furthermore, if tricks are algorithms then you may want to be sure they are correct.
They need to be verified, and so formal verification is a possibility. This also opens the
way for other discussion around the limitations of formal methods.

In live performances, the approach we use is to perform the trick, challenge the
audience to work out how it works (if it isn’t really ‘magic’), show how the trick is
done, giving a proof in some form, and then explain the linked computer science. The
proofs can be developed constructively with the audience, challenging them to work out
the steps. We have used this format for our magic shows, when doing close-up magic on stalls at science fairs, as the format of web and magazine articles and in a series of magic books.

Magic tricks have been used as a way to introduce mathematical ideas for many years. Martin Gardner has done perhaps most to popularize this idea in his books and columns on recreational mathematics (see e.g., [17]). The Unplugged project at the University of Canterbury, New Zealand were pioneers in using magic to explicitly teach computing, and specifically algorithms. Their parity trick allows the magician to know which of a large series of cards was turned over even though his back was turned at the time [1, 2]. This is of course based on adding parity bits and the person doing a parity check. It thus directly introduces error-correcting and error-detecting codes.

Kruse [18] describes a trick for demonstrating binary numbers and Simonson and Holm [24] describes one usable more generally to teach topics in discrete mathematics.

Garcia and Ginat [15] [16] extended this idea to a variety of tricks and to a range of computing topics including algorithms, functions, binary and to introduce the idea of model checking.

We also argued for the use of magic tricks as part of a broad, serious fun approach to engaging school children in computer science [5], illustrating the approach with a card trick linking algorithms with self-working magic. We have since extended the basic idea, using magic tricks to illustrate a wide range of computing concepts both in shows and in two magic booklets [19, 20]. We have focussed on both basic concepts but also specific areas including the use of magic to teach HCI [22].

Magic has been used to give deeper understanding of computer science issues to a wide variety of audiences. Spool [25] for example used a magic trick to illustrate, in a powerful way, the concept of delight and its importance in the design of e-commerce experiences, as part of a keynote at an academic conference.

The specific contribution of this paper is to show that a range of magic tricks can be used to both motivate the use of formal methods as well as illustrate a range of formal methods ideas. We also give a series of specific tricks that we have used in magic shows to do this in a connected way. All are very easy for novice magicians to do with only a little practice.

3 Magic tricks for introducing formal methods

In this section we describe a series of magic tricks we have used specifically to introduce formal methods concepts. In the current show the computer science points are illustrated in the context of our research project, CHI+MED [4] on the design and verification of safer medical devices, using medical device examples. Following the format we use in shows and close-up magic at science fairs, in the subsequent subsections we describe the magical effect of the trick, its mechanics (i.e., the algorithm) and then discuss the formal methods link. Over a series of tricks we can illustrate a series of ideas about formal modelling and verification in a connected way.

The first step in introducing formal methods using magic is to make the link between tricks and algorithms. From there we show why verification in general is important, how full testing is impractical but how logical argument can reduce the need to test, and how a rigorous diagrammatic proof can convince us that a trick always works.
We can then show how creating a mathematical model of a trick can allow us to use algebraic techniques to mathematically prove desirable properties - formal verification. We then show how the idea of invariants can be introduced as a basis for proving a trick work using induction. Finally we look at a trick that we use to discuss the limitations of formal methods and to motivate the modelling of human factors. For each trick we draw the parallel that as programs and tricks are both algorithms the same ideas apply to both.

3.1 The Australian Magician’s Dream: programs and algorithms

We first show the link between magic tricks and algorithms (and so programs).

3.1.1 The effect, mechanics and presentation

In this trick the magician is able to predict a revealed card that no one could possibly have known in advance.

Before the show starts, take an ordinary shuffled pack of cards and place the 8 of hearts in the 16th position. Place a distinctive card (e.g., the Ace of Hearts) in the 32nd position. Then take the 8 of hearts from a second pack (ideally from an over-sized pack for extra effect) and place it in a sealed envelope under the magicians table where it remains in full sight throughout the trick.

Get a volunteer from the audience to the front. Spread the cards across the table so they can see it is a normal shuffled pack. Announce that to make the trick a little quicker we need to roughly halve the pack. Spread your hands so they are over the 16th and 32nd card to indicate what you mean, asking them to point to a card of their choice roughly in the middle. This casually but usefully limits the spectator’s choice to dividing the pack between the 16th and 32nd card. Discard all the cards to the right of the one they point to, confirming with them that it was their free choice. Pick up the remaining cards and explain that before magic shows you have weird dreams where magicians teach you new tricks. You had such a dream last night when an Australian magician came to you and taught you the “Down-Under Deal”: a way to select a card that no one could possibly know you would end up with.

Now deal out the cards into two piles. You do this saying “Down” for the first card to start the first pile, placing it face down and “Under” for the second placing it face up to make the second pile. Repeat this saying “Down...Under” as you go until all the cards have been dealt out. You now discard the “Down” pile, noting you always throw away the down pile. Pick up the ‘under’ pile and repeat the process. Continue to do this until you are left with one card. (It will be the 8 of Hearts) in the face up ‘under’ pile. Say this is the selected card. Get the volunteer to confirm they had a free choice in where they split the pack. Turn the top few cards of the discard pile over to show “had you split the pack one card differently it would have been a different card”. Have them confirm they had no idea of what the resulting card would be and show it to the audience.

Now point out that the weird thing is that in your dream the Australian magician told you to place one particular card in an envelope. Ask the volunteer to look “Down Under the table” and reveal the card that was predicted. It is also the 8 of Hearts.
1. Place the chosen card in position 16
2. Discard roughly the bottom half
3. Repeat 4 times:
   • Discard the first and then every second card thereafter
4. Reveal the card is the one predicted.

Figure 1: The algorithm behind the Australian Magician’s dream

3.1.2 The formal methods link
This trick can be used to show that a self-working trick is an algorithm and that is exactly what computer programs are. To explain the mechanics put up a slide showing a pseudocode version (see Figure 1). Note how it is a series of steps, including loops, that if followed blindly and precisely leads to a specific magic effect. Explain how a program is a similar set of instructions for a computer to follow.

It works because, as long as there are no more than 31 cards, by repeatedly discarding every second card you are guaranteed to end with the 16th.

This trick actually is good for making this link as it has a deeper link that can be explained. A parallel (but otherwise identical) version of the underlying algorithm is the basis of the way early computers searched through data stored on punch cards. The binary representation of the number on a punch card can be used to filter out individual cards. Starting with a pin in the units column and working through the binary digits shaking out cards you can be left with any card required. You use the code: if there is a 0 in the binary at that position of the target number then discard the ‘down’ pile - the cards that shake out. If there is a 1 in that position in the target then keep the down pile. To find card 16 you therefore repeatedly discard the down pile until on the 5th round you keep the down card spelling out 10000. This is also what was done in the trick.

Self-working tricks and computer programs are the same thing. People who invent new tricks are doing the same thing as those writing new programs: programmers really are wizards!

3.2 The 21 card trick: testing versus logical argument
In this next trick we illustrate why testing is not a good way to be sure a trick (or program) works, and how with logical reasoning we can reduce the need for testing. This trick can also be used to make the link between magic and algorithms. This is also an appropriate way to introduce ideas about verification to younger school students and less mathematically gifted students, as it introduces the underlying motivation and ideas without overt use of maths.
3.2.1 The effect, mechanics and presentation

You lay out a series of cards on the table. A volunteer selects one without telling anyone what it is. You are able to read their mind and reveal the card.

Deal out 21 cards into three piles of 7 so all are visible. Ask a member of the audience to choose one of the cards without revealing it to anyone. Explain you are going to read their mind and work out which one it is. They must think hard about just this one card throughout the trick, clearing their mind of everything else. Stand in front of them and stare at their forehead intently. Tell the audience they must be quiet and the volunteer that they must not giggle. There is likely to be some noise and the volunteer is likely to giggle. Explain giggles are like bubbles that cloud everything so the giggles/noise are making it impossible. You will need to start over. Ask the volunteer to indicate which pile the card was in without saying which. You will deal them out again. They must think of the same card.

Pick up the three piles but place the pile they indicate in the middle. Deal the cards out again across the rows. Repeat the ‘mind reading’ procedure, this time standing closely behind them staring at the back of the head (“trying a different way in as the forehead is quite tough bone”). Again any disturbance is a problem. Get them to point to a pile again. Pick the cards up and deal them out in the same way. Next try from the side “there is a clearer route through the ears if they pull their hair out of the way (and cleaned their ears this morning)”. This time say you’ve almost got the card - “got the colour and almost the number”, but you want to try once more “to triangulate”. Get them to point to a pile, pick the cards up and deal them out the same way one last time. This time stare at the other ear. Claim you’ve got the card now. Go back to the cards and do the reveal: “It’s not in this pile....and it’s not in this pile” turning the two end piles over. “It’s ... this one” - lifting out the middle card of the middle pile. Have the volunteer confirm it is the card they were thinking of.

3.2.2 The formal methods link

This is again a self-working trick. The process of placing the pile with the card in in the middle as the piles are gathered, and dealing out across the rows, once done 3 times, guarantees the chosen card ends up in the middle card of the middle pile.

This can be used as a discussion of the need for verification. “I’m not going to do a trick in front of a live audience unless I’m sure it always works”. How can you be sure? - you can do tests: try the trick and see if it does work. But how many such tests do you need to do to be sure? Is checking a couple of times enough? There are 52 cards that each could be in one of 21 positions. Is it always going to work? That is too many possibilities to check them all. By introducing some logical reasoning, we can cut down the possibilities that need to be tested. The first thing to note is that the values of the cards do not matter, what matters is just the possible starting positions. Some simple reasoning reduces our test plan to just 21 tests. However we can do better. We can give a diagrammatic proof that all of those 21 positions end up in the middle if they are the position being thought of (see Figure 2). It tracks what happens to each card as the selected pile is placed in the middle. For example on the first step after the pile selected is in the middle this means that it is only the middle seven cards that we need to track as a chosen card starting elsewhere is moved to one of those positions. As we
deal out along the rows, these seven cards end up across the middle row, and so on.
This can also be used to introduce the idea of modelling a system you wish to reason about, abstracting to only the details that matter.

3.3 Mind-melding: mathematical models and algebraic proofs

Having illustrated how models and proofs can be used in an informal way, we can now show how algebraic techniques can actually be used to model a system and so show that tricks (and so programs) work.

3.3.1 The effect, mechanics and presentation

Two members of the audience are “mind-melded” so that as they deal out cards, without seeing the cards, they manage to match red and black cards.

Get two volunteers from the audience. Explain you are going to meld their minds together. Take an ordinary pack of 52 cards and ask the volunteers to shuffle them. Now deal out the cards so each gets half. Make them stand opposite one another across a small table. One person gets to look at their cards, the other doesn’t - they have to rely on the mind meld. Get them to stare into each other’s eyes to make the link.

Now the face-up person chooses some red cards from their hand, announcing how many cards have been taken, and places them in a ‘red’ pile face up in front of them. The face-down person takes the same number at random from their hand without looking at them and puts them on the table face down. They should be told not to think too hard about which cards - they are using the ‘psychic mind-meld link’ to get the right ones. The face up person does the same with some black cards to make a ‘black pile’, again copied by the face down person without looking at their cards. This is repeated with the first person choosing groups of red or black cards at will and being ‘copied’ by the other person until all the pack is gone. Each person should now have a ‘red’ pile and a ‘black pile’ in front of them. Emphasise though that no one knows what is actually in the face down piles. Those cards were chosen randomly, with no one seeing them at any point.

Move the cards round so one person has the two red piles and the other the two black piles. At this point say you think something amazing has happened (though you spotted one mistake). Pick a card out of each face down pile at random and swap them, saying you noticed that they ended up in the wrong place. Explain that what you believe what has happened through the mind meld, and despite any one seeing the cards, is that we have ended up with the red and blacks in a psychic link. There are the same number of red cards in the face down ‘red’ pile as black cards in the face down ‘black’ pile. Get one person to count out the red cards in the face down red pile onto the table. Announce the number. Then get the other person to count the black cards one at a time onto the table asking for a round of applause for them making the mind meld work if the number is the same.

Your prediction is right – the number of black cards and of red cards in those two piles are the same.
Figure 2: A diagrammatic proof of the 21 card trick
3.3.2 The formal methods link

To see what is happening, we first need to build a mathematical model. We do not know how many blacks or reds ended up in any pile - it will be different every time. Instead we give them names: B0 and R0 for the number of blacks and reds in the face up black and red piles, respectively; B1 and R1 for the number of blacks and reds in the face down ‘black’ pile, respectively; and B2 and R2 for the number of blacks and reds in the face down ‘red’ pile, respectively.

We now need to ask what if any facts we know. There are 26 reds in the pack, spread through the three piles so:

\[ R_0 + R_1 + R_2 = 26 \]  

Similarly for the 26 black cards we know:

\[ B_0 + B_1 + B_2 = 26 \]  

We also know that we always matched the numbers of cards in the face up red pile and face down red pile, though the latter is an unknown mixture of reds and blacks. This can be stated as:

\[ R_0 = R_1 + B_1 \]  

Similarly the numbers of cards in the black piles were made to match.

\[ B_0 = R_2 + B_2 \]  

We thus have a mathematical model in these equations of the system (i.e., the trick). We can now use algebra to try and prove the required property of the trick.

Taking (1) and (2) we get;

\[ R_0 + R_1 + R_2 = B_0 + B_1 + B_2 \]  

Using (3) and (4) to eliminate R0 and B0 gives

\[ R_1 + B_1 + R_1 + R_2 = R_2 + B_2 + B_1 + B_2 \]  

Cancelling gives the property:

\[ R_1 = B_2 \]  

Moving back in to the real world of the magic trick, what does this mean? Well, R1 was the number of red cards in the face down red pile. B2 was the number of black cards in the face down black pile. This says they are guaranteed to be the same: the predicted property. We have verified the main requirement of the trick.

Algebra has thus show that the trick always works. As an extra twist we can also look to the algebra to see that swapping the two cards “that were in the wrong places” makes no difference at all to the property’s truth.

We have thus shown how mathematical modelling and proof can be used to show that a trick (or program) meets a desired property. It also illustrates the idea of a formal specification of an algorithm. It can also lead to an introduction of model checking.
3.4 The Psychic Test: invariants and inductive proof

A further detailed example suitable for longer shows or if magic is being used at the
start of a class before more traditional exercises, we give a trick that can be used to
introduce the concept of an invariant and inductive proof in a rigorous correctness
argument or mathematical proof.

3.4.1 The effect, mechanics and presentation

A volunteer claiming to have no such powers is proved to be psychic. They are able to
match a series of cards into pairs without seeing any of them!

Take 10 cards consisting of a series of 5 cards of a suit followed by the same 5 cards
of a different suit placed in the same order. Ideally they should not be consecutive
numbers. Zener cards or other cards with matching sets can also be used.

Choose a volunteer from the audience who claims to have no psychic powers. Fan
the cards to show that you have a mix of cards and then turn the pack over. Explain
that you will first mix the cards up. Fan the cards, face down and ask them to touch
the back of any card. Cut the pack at this point, putting the top half to the bottom
and fan the cards again. Repeat this several times until they are happy the cards are
sufficiently mixed.

Tell them that they have to try to use whatever psychic or magical powers they
have. The aim is to leave two cards that match, but without seeing any cards at all.
Count out 5 cards into a pile on the table, reversing their order as you do so. Place
the remaining 5 cards straight down to make a second pile (unreversed). Give them
4 tokens (e.g., coins) and explain they can use them to channel their powers. Like
crystals they magnify any powers a person has! They should put each down on one of
the two piles without thinking too hard so their natural powers come through. They
could put all on one pile or spread them between the two piles. You will just do the
actions dictated by the tokens.

Once they have placed the tokens you now do the same number of ‘moves’ on a pile
as they have placed tokens. A move consists just of moving a card from the top to the
bottom of the pile. Explain this and then take the resulting top card of each pile and
place them together at the side. Place one of the tokens on top and point out it doesn’t
matter what these two cards are. What matters is which cards are left at the end. Now
repeat the process with the remaining three tokens, again putting aside the top cards
into a new pile with a token. Do this whole process again with the two tokens, and then
with the final token. Before they place the last token point out this is critical. They
can put it on either pile. Point out that if they get it wrong then you the magician will
look stupid! Once they have placed the token, look doubtful and ask them if they are
sure! Reluctantly go ahead and do the last move on that pile, and discard the top two
cards and token as before.

Remind them that the cards were shuffled at the start, and that they had a free
choice throughout. Ask them if, given they don’t think they have psychic powers,
they think it likely that the remaining cards match. Turn over one and suggest to
the audience that if the other card is the same they should cheer as you have found
a psychic amongst them. If not they should groan. Reveal the card, to cheers as it
matches.
Once the applause has died down, point out that once you have found someone with psychic powers it’s worth checking how strong those powers actually are. Reveal each of the other pairs you had put aside in turn with increasing excitement as every pair is seen to match!

Tell the volunteer to stop denying their powers and instead use them for the good of humanity!

3.4.2 The formal methods link

This trick can be used to illustrate invariants and inductive proof. This is a more complicated proof than the previous ones, so the details of inductive proof may not be suitable for a young, general audience, though even they can certainly learn the trick and understand how and why it works.

We explain how it works by giving an informal proof and introducing the ideas of invariants and inductive proofs as we go. There are actually two distinct parts to the trick: the shuffling step and the removal of pairs. Both involve an invariant property and are proved with an inductive proof.

First you demonstrate that the ‘shuffling’ has no effect on the important property of the pack – that it is two sets of the same cards, end to end in the same order. Note that this property is secretly set up at the start - it is the base case of the inductive proof. It is the invariant of the proof. Now point out that all you do by cutting a pack is rotate the cards retaining their order in the rotation. The underlying order stays the same. Not everything is the same however - the sequence starts in a different place. Thus there being an invariant does not mean nothing changes, just that some specific important property does. Each time we do the cut, the invariant is therefore preserved (the step case of the proof). Therefore, however many times the pack is rotated, the original property is preserved. The cards consist of two sets one after the other and in the same order.

Now to the main part of the trick. Its verification also involves an inductive proof, though it is a little more complicated. We now introduce a new invariant property. It is in two parts. The first part is a property of the two piles: that they are in the same order but in reverse. The second property is about the discarded pairs - that they match.

For the base case show the audience how the first pile was reversed but not the second. The first part of the invariant did hold at the start. The second part is also (trivially) true as there are no discarded piles yet.

Next we must show the step case. This new invariant property always holds again after a round has been completed and the resulting top two cards have been discarded.

First we look at whether the two cards that are discarded match after a round. We must show that the ones that end at the top match. This can be demonstrated for a specific number of cards left in the piles (e.g., 3 with 2 tokens) via a case split on where the tokens are placed. More generally it can be seen that whatever number of cards there are, if there is one less token than cards, then this will bring the same two cards to the top. This could be proved rigorously itself by induction, but to keep things simple we just demonstrate it holds physically with the cards.

Now because we have again just been rotating the cards the remaining cards remain in the same order. This can again be seen by demonstrating it holds for any given
number of cards in the piles.

We have therefore shown the step case. After every round of eliminating a pair, both parts of the invariant are preserved: all the pairs discarded so far match. The two piles are reversed copies of each other.

Therefore by induction the property will be true at the end of the trick. All the pairs will match.

This trick can also lead on to a discussion of Floyd-Hoare style program verification and in particular it also demonstrates how an invariant, despite the name, may not be true at all times. As we rotate individual cards in the main part of the trick, the invariant ceases to hold. The two piles are temporarily not rotated copies of each other. The point is that it is always restored by the start of the next round.

3.5 The Four Aces: limitations and human computer interaction

Our formal methods research is concerned with verification of interfaces and interaction design to reduce operator error [14, 23]. This next trick can be used to discuss more indirectly how this kind of issue can be modelled.

3.5.1 The effect, mechanics and presentation

You deal out 4 hands of cards each with an Ace as the last card. The audience keep track of where the Aces are. They see that one person gets them all and has a perfect hand. However, when the cards are revealed, they have no Aces at all. It is you the magician who has the perfect hand.

Get a volunteer out to the front. Deal 4 Aces face up to make 4 hands. Explain to the audience this is a trick about why you should never gamble, and certainly not with magicians or computer scientists. Add three normal cards face down on top of each Ace showing the audience the cards as you do so. Explain you have made 4 hands that they should assume are the way the cards fell at the end of the last round. Now in this game whoever has the most aces wins, so if you can keep track of where the Aces go then you can use that information to decide whether to gamble and so beat the odds. Tell the audience that is what they must do: watch the Aces. Now turn each Ace back over leaving it at the bottom of its pile. Collect the 4 piles up and note that if the dealer does not shuffle then you know exactly where the Aces will go. As they are every fourth card, if you deal out four hands then they will all end up together on the fourth pile. Start to do this to illustrate, counting as you deal 1, 2, 3 and an Ace. At this point pause and pointing to the Ace with the next card ask the volunteer to turn it over and show the audience it really is an Ace. As they do so, draw the card you pointed with back and slip it to the bottom of the pack, taking the top card in its place. Unknown to the audience the next Ace is now three cards from the top not four. Tell the volunteer to leave the Ace face up so we can see that is where the Aces are and then continue to deal the other cards out, counting 1, 2, 3, Ace as you do.

Now point out that one person (let’s assume it’s the volunteer) has all the Aces. They cannot lose. They will therefore gamble high. Now ask the volunteer to point (note “point” not “choose”) to two of the other piles. If one of the piles they point to is the third (which unknown to the audience has the three Aces) then remove the pile
they didn’t point to, saying that they drop out. Talk more about how the person can’t lose so will keep raising the stakes while anyone else stays in. Now ask the person to point to one of the two remaining piles. Whichever they point to remove the one that isn’t the one with the Aces, saying they drop out. If originally they point to the first two then remove both of them and say that they drop out.

Note that with one person still in the one with the Aces will keep raising the stakes. At this point you should drop out but you are not going to, you are going to cheat and try and steal the Aces while no one is watching. In this game you are allowed to swap one card with your opponent at this point. The volunteer won’t want to as that would make their hand worse. You however will. Slide out the face-up Ace from the bottom of the fourth pile. Also slide out the bottom card of the third pile, turning it over to show it is just an ordinary card. Swap them leaving them face up. Note you now have one Ace but the other person has three so will still win. Now tell the audience they must watch the fourth pile. You are going to try and make them all blink at the same time, giving you time to steal the Aces. On the count of three clap you hands in front of the volunteer. Ask them if they blinked. Note that they use to have three Aces and a winning hand but now...turn the fourth pile over and show everyone you’ve stolen them. All that money gambled has been lost! So who has the perfect hand? Turn the third pile over to reveal you have all the Aces. Announce that that is why you should not gamble with magicians or computer scientists.

3.5.2 The formal methods link

This trick has an underlying algorithm, but includes a step that relies on sleight of hand. The audience do not see everything. Despite that the trick could still be verified. In sleight of hand tricks, however, proving that the mechanics alone work is not enough to guarantee the success of the trick. This in itself is a way to talk about the limitations of formal methods and how they are proofs about the model and not reality. A trick (or program) that has been verified is not guaranteed to be successful. In particular HCI issues such as usability are also important.

In this case the trick relies on simple misdirection too. However, there is a lesson in that. Magic tricks like this show that it is possible to engineer a system so that everyone makes the same mistake at the same time. Some human errors are not about negligence or stupidity but are about cognitive limitations. In this case we can only focus our attention on one small area at a time. If our attention is drawn to one point then we will miss other things. Magicians design systems so we make mistakes, computer scientists must design computer systems to do the opposite. Instead of drawing a person’s attention away from things that matter, their attention should be drawn to them. For example, if a nurse mistypes a drug dose into an infusion pump delivering a drug, then before starting the infusion, you want their eye drawn back to the screen to check the dose, not away from it.

This shows that some human behaviour is systematic. That means it can be modelled and usefully reasoned about. We can verify systems with respect to whether they encourage certain kinds of human error. This is a focus of our CHI+MED research project’s application of formal methods: formal modelling of user behaviour [14, 23]. More generally it shows that we can reason about interfaces and interaction design, by modelling the aspects that matter depending on the usability properties of interest.
4 Conclusions

We have argued that card tricks are a fun way to introduce ideas of modelling, formal specification and formal program verification. The need to verify tricks is motivated by not wanting a trick to go wrong in front of a live audience. We have shown that with simple, easy to learn tricks a range of formal methods concepts can be introduced, gradually building in formality. Because self-working tricks are algorithms, the link to program verification can be directly made.

We have done these tricks both individually and together as a way of explaining formal methods concepts at science fairs as close-up magic, and as a part of magic shows to audiences up to several hundreds in size. We have also given them to high school audiences of a variety of ages. Feedback, both informally and via formal feedback forms, from both teachers and students has been highly positive. Our aim in these shows has been to inspire students rather than directly teach a program verification class. However, we believe, when combined with more traditional teaching they provide a powerful and fun way to introduce formal methods concepts. Our target audience has been school students but the approach, with more formality added, could also be used to illustrate theory to university students too. We leave this as further work.

In the descriptions above we have given direct explanations, however in practice the audience should be encouraged to work out the steps of the proofs themselves constructively. The beauty of using card tricks is that, unlike with a program, you have a physical model in the cards themselves to make the properties you are proving tangible.

A wide variety of tricks can be used to illustrate these concepts and others not included here. As a magic trick is essentially a computational process, just one embodied in the real world, magic is a powerful, fun and flexible way to introduce theoretical computing concepts.

Acknowledgements

We are grateful for support from EPSRC under research agreements EP/F032641/1 and EP/G059063/1 as part of the cs4fn public engagement project and the CHI+MED research project on safer medical devices. Google have also supported our live shows. We thank the reviewers for their useful comments.

References


