Juggling and Card Shuffling Meet Mathematical Fonts

Erik D. Demaine* Martin L. Demaine*

In honor of Ron Graham's 80th birthday.

Abstract

We explore two of Ron Graham's passions—juggling patterns and perfect card shuffling—through one of our passions, mathematical fonts. First, for each letter of the English alphabet, we design a one-person three-ball juggling pattern where the balls trace out the letter (possibly rotated 90°). Second, using a deck of 26 cards labeled A through Z, we show how to perform a sequence of in/out perfect riffle shuffles to display any desired sequence of letters (forming a word, phrase, etc.), using algorithms already developed by Diaconis and Graham. Along the way, we pose some new open problems about perfect shuffling.

1 Introduction

A *mathematical typeface* is based on a mathematical theorem or open problem. In this way, the way that text is written, and not just the text itself, can engage the reader in mathematical thinking. A *puzzle typeface* hides the text from plain sight, requiring solving a puzzle to decipher the underlying text. By combining these two ideas, we can write secret messages encoded with various mathematical ideas, and readers who are enthusiastic about puzzles can learn about beautiful mathematics by solving mathematical puzzles to decipher text.

We have so far developed half a dozen of these mathematical/puzzle typefaces, based on mathematics ranging from hinged dissections to geometric tours to computational origami to the geometry of blown glass. See [DD15] for a survey, and visit our website to play with the fonts yourself.[1]

To celebrate Ron Graham's 80th birthday, we created two new typefaces based on two of Ron's (and our) favorite contexts where mathematics meet the unusual: juggling and card shuffling. Both typefaces are designed to be performed. The juggling typeface consists of a three-ball two-hand juggling pattern for each letter, enabling the expression of an n-letter word or phrase through n simultaneous jugglers or one juggler performing a sequence of patterns. The card shuffling typeface expresses an n-letter word or phrase through a sequence of $O(n)$ operations—in perfect shuffles, out perfect shuffles, and revealing the top card in a 26-card alphabet deck—enabling an expert card magician to pull out an audience-specified message.

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, {edemaine,mdemaine}@mit.edu

[1] http://erikdemaine.org/fonts/

2 Juggling Fonts

Juggling patterns are well studied mathematically, from the initial work of Shannon in 1981 [Sha93] to the foundational paper by Buhler, Eisenbud, Graham, and Wright [BEGW94]; see [Pol03]. One major revolution, embodied by *siteswap notation* used by many jugglers, is the abstraction of juggling patterns into a timing sequence for when objects land after being thrown. In addition to raising many clean mathematical questions about juggling patterns, this perspective enables easy communication of juggling patterns as well as computer simulation of these patterns. Some computer simulators, such as Juggling Lab [B^{+}14], add modeling of performance aspects of juggling patterns beyond the mathematical abstraction, such as hand motion between object throws.

Our idea for representing letters as juggling patterns is visual: we design cyclic juggling patterns, with particular hand motions, such that the trajectories of the balls trace the letter, in some cases rotated 90°. Figure 1 shows a few such juggling patterns and their trajectories. One way to experience the trajectories is with persistence of vision: lit balls in a dark room will paint a longer time range onto the retina. But we prefer the puzzle of watching the juggling pattern and visualizing its projection down the time axis.

Figure 2 shows the full typeface in the (easy-to-read) trajectory font. To see the more puzzling animated font, and to really see the underlying juggling patterns, visit the webapp.[2] For a classic juggling feel, we kept all patterns to three balls, two hands, and toss juggling, though we plan to make other typefaces with other styles of juggling. Some letters come from classic juggling patterns: A is Half Shower, E is Columns, F is an Exchange out of 2 balls, I is Yo-Yo, O is Shower, U is Box, W is Double Box / Extended Box, and X is Reverse Cascade. Letters D, M, and N are all variations of Columns; N is perhaps new, while the others are known but perhaps unnamed. Letter B is essentially three balls out of a four-ball Reverse Fountain (in siteswap notation, 4440). Letter C is a known (but hard) variation of Cascade (sometimes called Arches), while S and its mirror image Z are new and extra-hard variations of Cascade involving very fast hand motions between throws. Several letters (H, K, P, Q, R, T, V, Y) are essentially two-ball juggling with a third ball held and moved to finish the letter, similar to classic tricks like Fake Columns and Yo-Yo (in siteswap notation, 42). Letter G and its mirror image J are challenging variations of Columns with a back-and-forth cross throw in between (in siteswap notation, 6411). Finally, L is a synchronous pair of cross throws mixed with two-ball Columns (in extended siteswap notation, "(2,6)(2x,2x)"). Letters B, E, and R are rotated 90°, as it seems difficult to construct them upright.

This typeface is meant to be performed. With many jugglers arranged in a line, we can write a message by having each juggler perform one letter. Figure 3 shows a photograph from such a performance of Figure 1. With only one juggler, we can transition from one letter to another. For the mathematical abstraction of juggling patterns, optimal transitions can be found algorithmically [CKL06]; for the constant-size patterns of this typeface, performers can find suitable transitions through practice.

3 Card Shuffling Fonts

Perfect riffle shuffles (often called Fano shuffles) fascinate mathematicians and magicians alike. The trick is difficult to perform, requiring the deck to be perfectly divided in half, and cards to alternate perfectly from the two half-decks. In fact, there are two types of perfect shuffles, depending on which half-deck drops a card first; see Figure 4. *Outside* perfect shuffles preserve the topmost and

[2] http://erikdemaine.org/fonts/juggling/

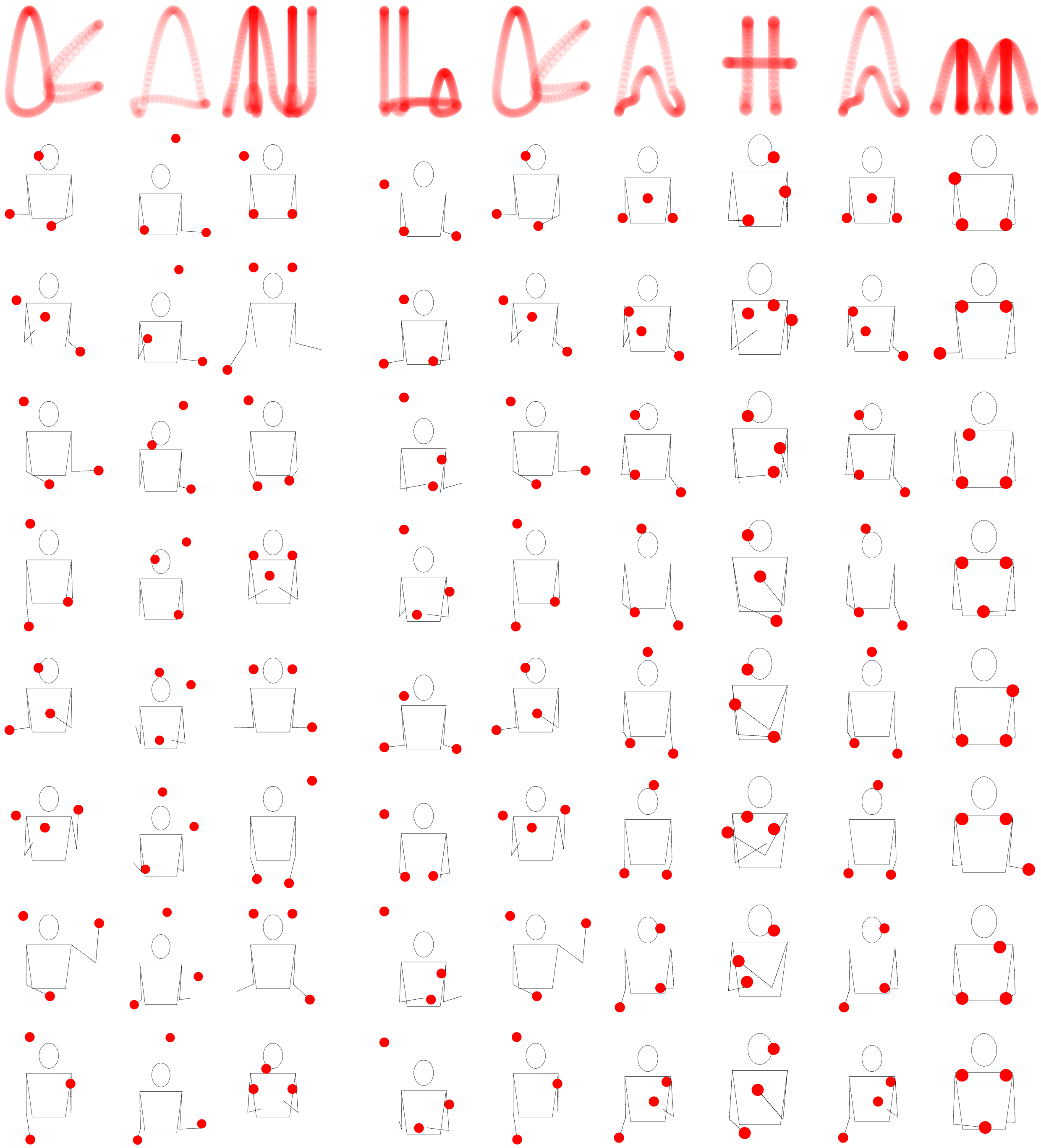


Figure 1: Expressing "RON GRAHAM" as juggling trajectories and animations. Time proceeds vertically. Letter R is rotated 90°. Animations produced with Juggling Lab [B$^+$14].

bottommost cards (on the "outside" of the deck), while *inside* perfect shuffles bring the two middle cards to the outside.

When successful, perfect shuffles imbue a seemingly random operation (riffle shuffling) with powerful deterministic properties, which can be used to great effect. Perhaps most famously, eight outside perfect shuffles of a 52-card deck return it to its original order. On the other hand, if we

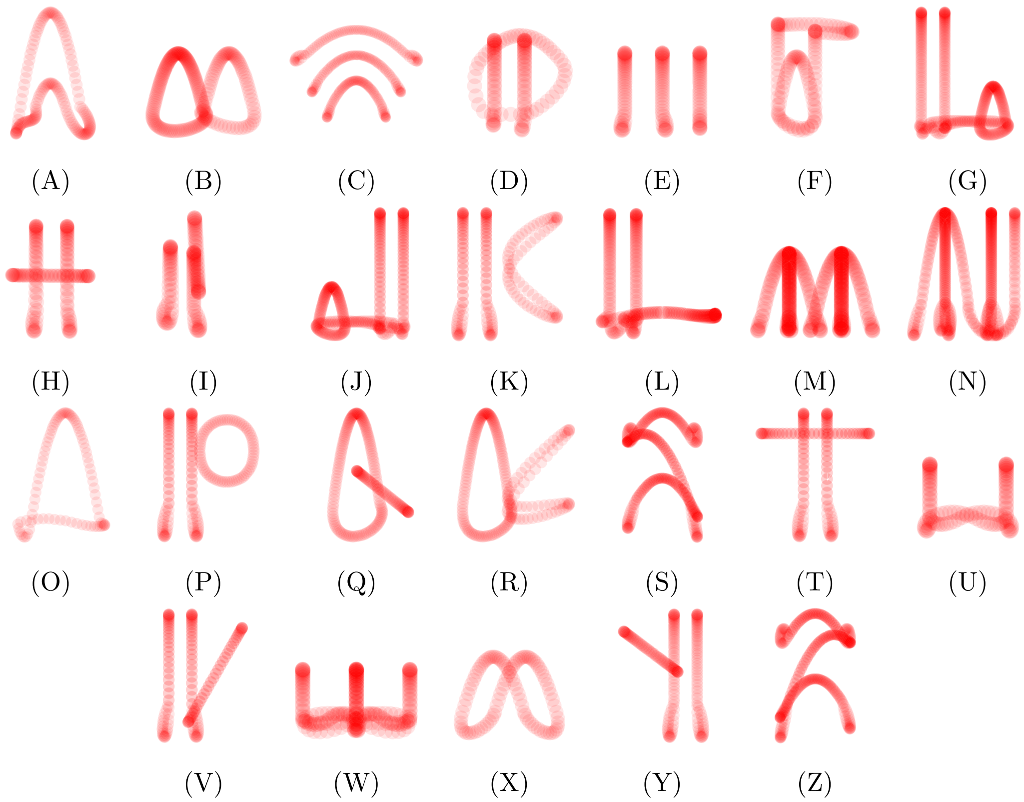


Figure 2: Juggling typeface, trajectory font. Underlying animations produced with Juggling Lab [B+14].

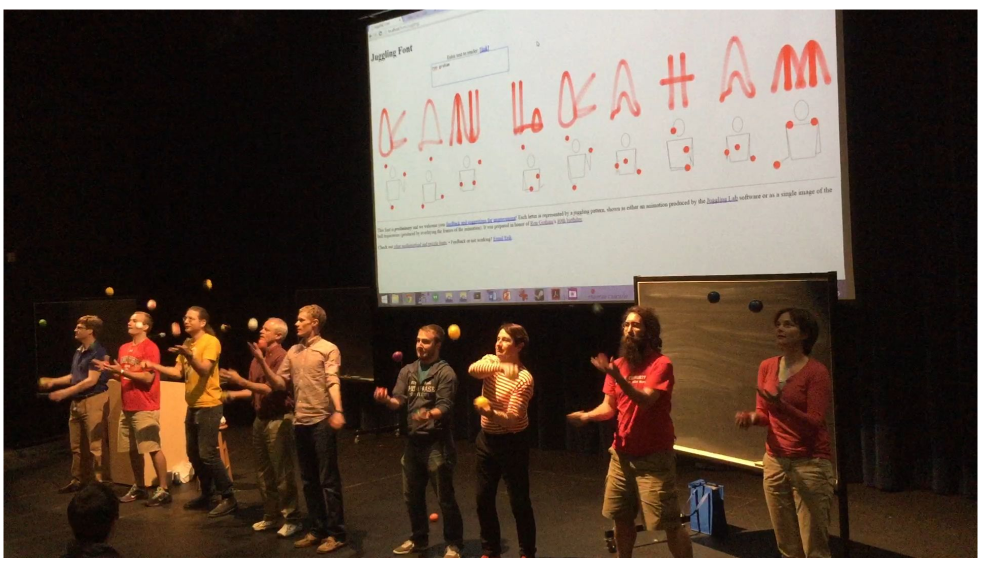
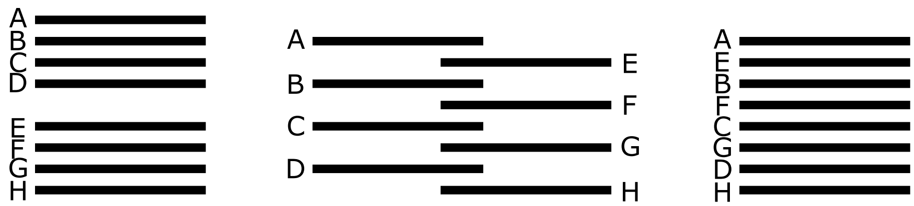
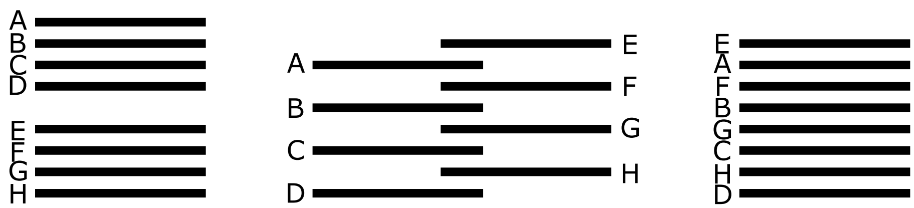


Figure 3: RON GRAHAM (Figure 1) performed by Colin Wright, Jeffrey Davis, Erik Demaine, Glenn Hurlbert, Jay Cummings, Jacob Landgraf, Peter Frankl, Pat Dragon, and M. Puck Rombach (from left to right) at the Connections in Discrete Mathematics conference on the occasion of Ron Graham's 80th birthday.



(a) Outside perfect shuffle: top and bottom cards (A and H) remain on the outside.



(b) Inside perfect shuffle: top and bottom cards (A and H) move inside the deck.

Figure 4: The two types of Fano/perfect shuffles.

repeatedly inside perfect shuffle a 52-deck card, it would take 52 iterations before the deck returns to its original order [DG07, DGK83]. These iteration counts, $k = 8$ and $k = 52$, come from finding the minimum values of k for which $2^k \equiv 1 \pmod{52-1}$ and $2^k \equiv 1 \pmod{52+1}$, respectively.

When Persi Diaconis was 13, he learned about the mathematical magic of perfect shuffles from Alex Elmsley, a British computer scientist [Hof11]. In addition to the property of eight outside perfect shuffles, Elmsley showed Diaconis his technique for moving a card from the top (0th) position to the kth position in the deck using the binary representation of k: for each 0, perform an outside perfect shuffle (O), and for each 1, perform an inside perfect shuffle (I) [Elm57, DG07].

Half a century later, Diaconis and Graham [DG07] solved the reverse problem, posed by Elmsley [Elm57]: how to move a card from the kth position up to the top of the deck. Amazingly, this feat can be achieved using at most $\lceil \lg n \rceil$ perfect shuffles in an n-card deck. This bound is information theoretically optimal: the trick provides a binary encoding (mapping outside/O to 0 and inside/I to 1) for each of the n cards. Thus, a magician who knows a card's location within a 52-card deck can extract it from the top of the deck after at most six perfect shuffles.

This trick suggests a performance font with a 26-card deck, one card for each letter of the alphabet.[3] The deck starts sorted A through Z. To spell a message (for example, provided by the audience), the magician performs at most five perfect shuffles to bring each letter to the top, reveals the top card, and continues to the next letter.[4]

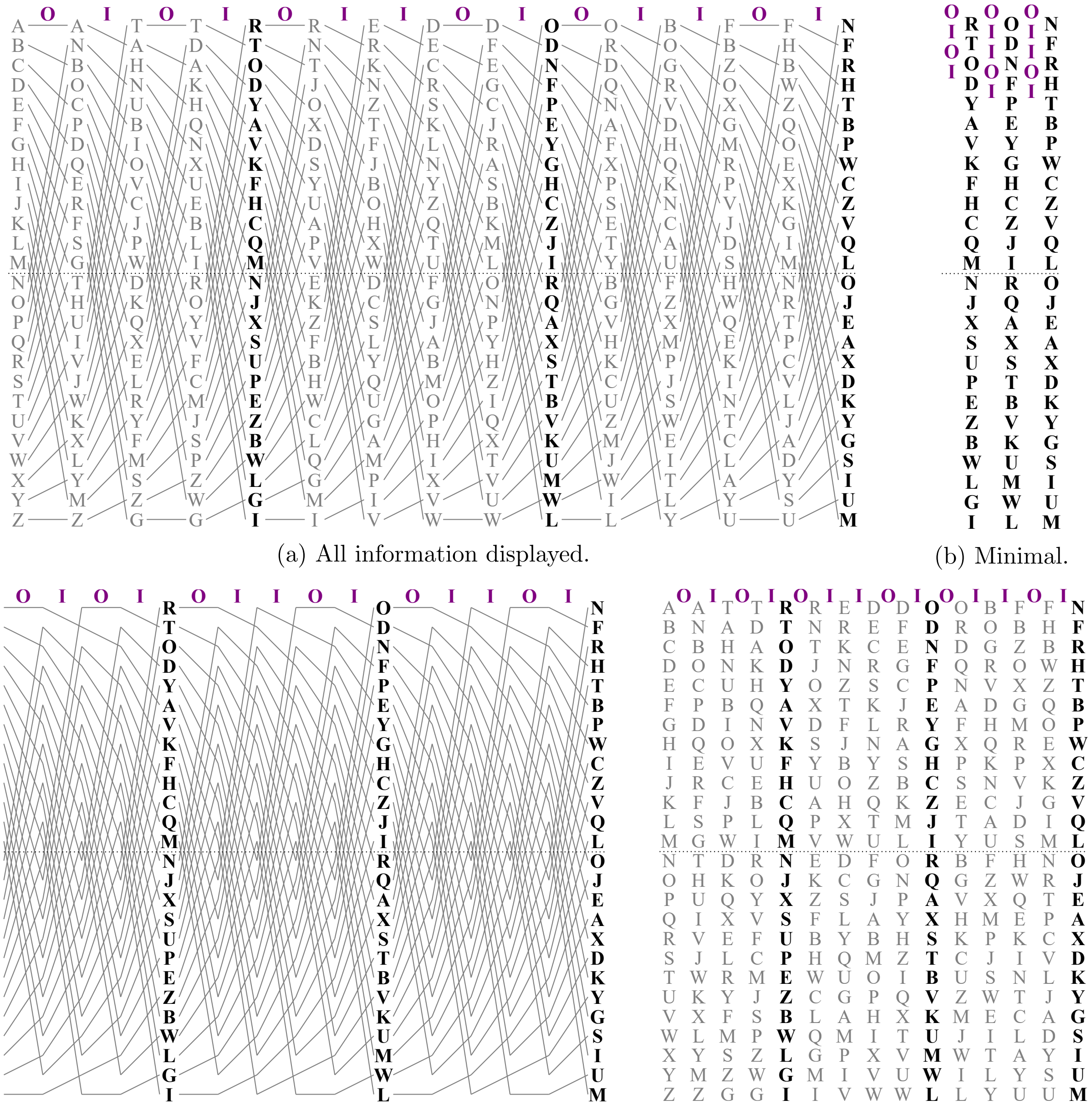
This effect can also be illustrated by a variety of visual fonts. Figure 5 shows examples produced by the webapp.[5] Here we optionally represent the shuffling permutations graphically, and optionally show each intermediate permutation of the deck. When the top card is a desired letter, the permutations are drawn bold. At the top, we indicate the inside/outside (I/O) perfect shuffle sequence.

An interesting property of this typeface is that the representation of a letter depends on the current state of the deck, which in turn depends on all previous letters. For example, in Figure 5,

[3] Several such decks are commercially available. We recommend http://www.magictricks.com/alphabet-deck.html.

[4] In fact, we thought up this effect, and experimented with it computationally, before we knew about Elmsley's Problem and its solution. Thanks to Persi and Ron for pointing us to their work during the Connections in Discrete Mathematics conference!

[5] http://erikdemaine.org/fonts/shuffle/



(a) All information displayed.

(b) Minimal.

(c) Graphics without intermediate permutations.

(d) Intermediate permutations without graphics.

Figure 5: Representing "RON" by a sequence of Fano/perfect shuffles: OIOI | OIIOI | OIIOI |.

the same sequence OIIOI produces two different letters (which happen to be in the same position in the deck at the relevant times).

We can use the perfect shuffle sequence alone to encode messages in a puzzle font. Can you decode the following puzzles, where each "|" symbol indicates the reveal of the top card? (Solutions are below.)

Puzzle 1. OOOII | IOOOI |

Puzzle 2. IOII | OIOOI | IIOOI | IIIOI | OIOOI | OOOII | OOII | OIOI |

Puzzle 3. OOOOI | IIOOI | OOII | IIOOI | | III | OOOI |

Other perfect shuffling typefaces could use different decks, for example, a 52-card deck of uppercase and lowercase letters.

A natural open question is to find the shortest execution of a given letter sequence. Diaconis and Graham [DG07] point out that there can be two solutions of at most $\lceil \lg n \rceil$ perfect shuffles, and there are presumably more of longer lengths. Our approach has been to greedily repeatedly apply the shortest solution to each resulting Elmsley's Problem, but plausibly it could help to do more shuffles earlier in order to permute the cards so that later letters become easier to reach.

A more fundamental open question is to characterize how many perfect shuffles are needed to achieve each (reachable) permutation of a deck. Diaconis, Graham, and Kantor [DGK83] characterized which permutations are reachable for each deck size, viewed as a group. For example, a 26-card deck can reach half the permutations, while a 52-card deck can reach them all. But we know little about how many perfect shuffles we need to represent each permutation. In particular, what is the diameter of the directed graph of reachable card permutations, which has an edge for each perfect shuffle (so every vertex has out-degree 2)? Given two permutations, can we compute the shortest path connecting them, or is this problem NP-hard?

Puzzle Solutions:

1. HI • 2. BIRTHDAY • 3. SHUFFLE

References

[B$^+$14] Jack Boyce et al. Juggling lab. http://jugglinglab.sourceforge.net/, 2014.

[BEGW94] Joe Buhler, David Eisenbud, Ron Graham, and Colin Wright. Juggling drops and descents. *American Mathematical Monthly*, 101:507–519, 1994.

[CKL06] Jean Cardinal, Steve Kremer, and Stefan Langerman. Juggling with pattern matching. *Theory of Computing Systems*, 39(3):425–437, 2006.

[DD15] Erik D. Demaine and Martin L. Demaine. Fun with fonts: Algorithmic typography. *Theoretical Computer Science*, 586:111–119, June 2015.

[DG07] Persi Diaconis and Ron Graham. The solutions to Elmsley's problem. *Math Horizons*, 14:22–27, February 2007.

[DGK83] Persi Diaconis, R. L. Graham, and William M. Kantor. The mathematics of perfect shuffles. *Advances in Applied Mathematics*, 4(2):175–196, 1983.

[Elm57] Alex Elmsley. The mathematics of the weave shuffle. *The Pentagram*, 11(9–11), June–August 1957.

[Hof11] Jascha Hoffman. Q&A: The mathemagician. *Nature*, 478:457, October 2011.

[Pol03] Burkard Polster. *The Mathematics of Juggling*. Springer, 2003.

[Sha93] Claude E. Shannon. Scientific aspects of juggling. In N. J. A. Sloane and A. D. Wyner, editors, *Claude Elwood Shannon: Collected Papers*, pages 850–864. IEEE Press, 1993.