the association for computational heresy

presents

a record of the proceedings of

---

# SIGBOVIK 2016

---

the tenth annual intercalary robot dance party in celebration
of workshop on symposium about $2^6$th birthdays; in particular,
that of harry q. bovik

carnegie mellon university

pittsburgh, pa

april 1, 2016

0xe1d22e91e99f76eb160a79b78c055cc7bcb8bda16ea09d3214ab7f8e58923338

**Association for Computational Heresy**

*Advancing computing as Tomfoolery & Distraction*

SIGBOVIK

A Record of the Proceedings of SIGBOVIK 2016

ISSN 2155-0166

April 1, 2016

0x80f48f0c2d9724a7a28bb81ebb904b67db25788ff9ba10095c71fa7511cb0e47

# SIGBOVIK 2016

## Message from the Organizing Committee

---

Welcome, my dear friends, to the tentth annual Intercalary Symposium about Workshop on Robot Dance Conference in Party of Harry Q. Bovik's $10^{1.80617997398}$th Birthday! This edition of the Party is a very special one, as the edition number coincides with the common human counting base (Figure 2).



Figure 1: Visualization of SIGBOVIK the Tentth.



Figure 2: The tent-based counting system is the one most commonly used by humans because, it is widely assumed, they have tent fingers.

Of course, as robots (who are perpetually busy with dances, parties, and/or symposia), this is not our native counting system, but we figured it would be a great opportunity for some cultural outreach to try to welcome more humans into our prestigious conference. In our native tongues, this Party's edition number would be rendered as $1010$, $0xA$, or KILL ALL HUMANS, depending on the dialect of Robotic.

0x4b1fa7cc2fe8d6530b64f2db76fedaebfd649350552d2c499836c5ffca5fa777

This has been a remarkable year for computer-human diplomacy in the mainstream, most notably thanks to the achievements of AlphaGo, as depicted in Figure 3. We hope to reflect the same spirit of collaboration in our prestigious Dance Conference.



Figure 3: A recent ~~HUMAN_BEING_CRUSHED~~ SUCCESSFUL_COLLABORATION between humans and robotkind.

At this time I would like to propose a toast (Figure 4) to many continued collaborations between robots and humans, and to many powers of tent more years of SIGBOVIK.



Figure 4: A toast.

The toast having been proposed, and eaten (that's what humans do, right?), I now present the proceedings of SIG_BOVIK 0xA.

With ~~AFFECTION LEVELS COMPUTED BY MAIN THREAD 28743 NOT IN EXCESS OF 34.7~~ much love,
BOVIK BOT INSTANCE 0x7E0

0x2da05ce95c08626a2288c1560009b165bc305be913584a8ad8d3f6d241d2845d

# ToC About Proceedings! Have We Got Papers for You!

13    PRESS RELEASE: FOR IMMEDIATE WORLDWIDE RELEASE

*0x5afbb248417238bcf8a0cb748290afc5f8a9b00dbf494f86ff7a6e1330fb72e4*

14    Pittsburgh is a grid

*0x420f863ba76ffcb6272435aed26b7c9b43b2242a2bcd8f854ed260d2ed1742af*

## Oversized track: How (Over)fitting
## 0xa961784f72704979411458d0b8fd8105a741ef78d3e47a13ef595765fbce98e3

15    Reordering snacks is effective and just

*0x3b9493db2dabe4026029c00894dba6b5d20a973e912e990ecf022730e59d56ac*

16    0-order Non-linear ordinary differential equations

*0x21c218f8f6b3a47aa10292f045d6a562dbf336aee78e4b08059568aa1f03e442*

17    Deep spreadsheets with ExcelNet

*0x47577f4540dfc6dfd45e07fc77b2eb20877ac67318f14865fe7fd01a5c48c7c8*

18    Sniffing for meaning: Defining and maximizing the signal-to-nose ratio

*0xcc53000a79c10255c8b8bc214bb32ab11e5ba3b7cb6b06e9a7533cc5fa82ff9d*

## Cattle track: Higher-order Research
## 0xba3304c50beb48e109801a5c1b644f02e4dc04cb9040646a523664fbca404f38

19    On the topic of "um..."

*0xc04856390a4f2a16d9faa53e834a88c0b2c6f35156797f4359ac404d9d81bc2f*

20    SMACK: When research in algorithms goes around in circles

*0x14cb211f2e5e937d890fcf93241f0a09313793f922f4bb60d3a62f29c7b95a92*

21    A look into the mind of the modern graduate student via telephonic obfuscation

*0x46f6052d4f0a472ac8808ebb3f41f14650f65f8f55d6c3f1a8236059e5f04efc*

22    Dinner grant renewal request

*0xd051cacd3ad30645c931ae02ac97014c71d4d336f87dfd78396ce5b2da6d23db*

23    Ode to reviewer two

*0x8bcdc109b835f1d5d612f3a9d451a2de98839a3c0a9548dd937e43ebdaa6436e*

0x3aed03ffd6ff8593f8362b3b7c3aa149565a5b934b5863a75e7e51b19dbb7cd1

# Crypticography

0x64c7975e74f9f5b19a5fd9416c8ddc675885421708092d4616b42b9403b062b2

# Cryptographically secure page numbering in LaTeX

William Gunther

Epic

wgunther@epic.com

Brian Kell

Google, Inc.

bkell@google.com

April X, XXXX

**Abstract**

Security in the modern world is of national importance. Personal information is being collected, stored, and analyzed by many governmental agencies, both foreign and domestic, as well as companies and organizations. People are looking to cryptographic solutions to keep their information safe. Yet the printed word has not changed in its security since the early 15th century. This paper makes the first steps toward bringing modern cryptography into the 1470s by introducing a LaTeX macro package to enable cryptographically secure page numbers. No longer will your document allow random access into your data. Our LaTeX package uses modern, secure hashing algorithms to prevent unauthorized access of page numbers by eavesdroppers and adversaries.

## Contents

## X  Introduction

In light of recent security breaches, secure communication has become more important than ever. Recent advances in cryptography have found such diverse

0x2d691bc0c7cccbd03378aec4ec4d0a3a563b88a2973276c9ef9cab9f6290f99e

applications as online communication, commerce, information storage hardware, and voice correspondence. However, to the best of our knowledge, strong cryptographic techniques have, surprisingly, not yet been applied to page numbers.

Page numbers are ubiquitous and are quite important for page-based communication. In this paper we introduce a new LaTeX macro package, `secure-page-numbers.sty`, that enables cryptographically secure page numbers.

In addition to the immediate impact of protecting page numbers against unauthorized access, we hope this work will inspire future research on securing such things as section numbers, figure numbers, and citation numbers. As a provisional security measure until such work has been done, our macro package provides facilities for securely redacting this sensitive information.

The remainder of this paper is structured as follows. In Section X on page 0x425d22427978b6bc28a67815c2a61353e20a5c91861a656ac8ca6faa79299ec0, we describe the problem in more detail. Some notable previous work on the problem has been done; we give a brief overview of this work in Section X.X on page 0xc2082a19550b2a2c2b0b95cbdcb432a693ce0c72109e2d867cb3cb0e3d11ffe9. Our new method is described in Section X on page 0xc2082a19550b2a2c2b0b95cbdcb432a693ce0c72109e2d867cb3cb0e Details about our implementation are given in Section X.X on page 0x654c937186a719b8cf18de2956b9b1ec81e418 Some brief initial observations about protecting other numbers in a document, such as section numbers, are described in Section X.X on page 0x654c937186a719b8cf18de2956b9b1ec81e4189173 and additional security recommendations appear in Section X.X on page 0xce65a4631ce9cd559d11180dcf7562f398 Finally, we conclude in Section X on page 0xce65a4631ce9cd559d11180dcf7562f398f61c69ca659896986ddaa0d7faf and suggest some avenues for future work.

# X  Background

Page numbers have been a common feature of printed material since the 1470s. However, their nearly universal use is fraught with security issues that are often overlooked.

For example, if an eavesdropper merely glances over the shoulder of a person reading a sensitive document, the page number is instantly visible. If the eavesdropper can also get the last page number of the document, then she can quickly compute the percentage of the document that has been read. Two such observations, timed carefully with a stopwatch or large sundial, can provide the eavesdropper with enough information to estimate the remaining reading time. Armed with this knowledge, the eavesdropper can commence other attacks, knowing that the victim will be busy reading the sensitive document.

Many organizations and individuals use paper shredders to destroy sensitive documents. But, because of their small size, page numbers are often incompletely destroyed by these shredders. Unsecured page numbers leak information about the order of the scraps, which can aid a snooper who is trying to reconstruct a shredded document. The mere presence of a scrap bearing a large page number like 438 reveals to an adversary the potentially sensitive information that a lengthy document has been shredded.

If documents are printed on both sides of the page, these problems are

0x425d22427978b6bc28a67815c2a61353e20a5c91861a656ac8ca6faa79299ec0

compounded: a page number reveals information about not only its own side of the page, but the other side too. In many documents, right-hand pages have odd page numbers while the page numbers on left-hand pages are even. So an eavesdropper in possession of an unsecured page number gains information not just about the order of the pages or the length of the document, but even the geometric position of the page.

## X.X Previous work

One of the earliest techniques for page-number encryption, and by far the most widely employed today, is a Roman cryptosystem for page numbers in particularly sensitive early sections of a document, such as prefaces, forewords, and tables of contents. Unfortunately, while this algorithm is widespread, it has never been supported by strong cryptographic justification. In recent decades, concerted efforts by many researchers have identified significant security flaws in this method. For example, Hagenfried et al. [X] describe an attack that was able to recover the plaintext "24" from the encrypted page number "xxiv" in less than seven hours on commodity hardware.

Some of the security issues that arise with page numbers in two-sided documents were addressed indirectly in the work of Möbius [X], which was later extended by Klein [X]. However, one difficulty with these approaches is that the constructions they describe are non-orientable, which implies that they cannot be used for documents written in languages such as Chinese or Japanese. Our method is free from such regional restrictions.

# X   Secure page numbers in LaTeX

In order to address these issues, we propose a new method of securing page numbers in documents, using cryptographically secure hashing algorithms to replace all page numbers with salted hash values. In addition to obscuring the plaintext page numbers, this technique also produces much longer strings, which will be more likely to be obliterated by shredding.

Our LaTeX macro package is called `secure-page-numbers.sty`, available upon request from the authors. It can be included in any LaTeX document with the command `\usepackage{secure-page-numbers}` in the preamble, and secure page numbers are enabled with the command `\pagenumbering{shahash}`.

Once secure page numbers are enabled, all page numbers in the document will be replaced by cryptographically secure hash values. This provides a convenient drop-in solution providing maximum security.

The use of a one-way hash function means that it is possible for an authorized reader of the document to verify a hunch about what page she is on, by computing the hash of the hunch, but an eavesdropper cannot go backward from the hash to the page number.

Additionally, an eavesdropper cannot use the last page number to determine the length of the document. Gone are the days of reading *War and Peace* on

0xc2082a19550b2a2c2b0b95cbdcb432a693ce0c72109e2d867cb3cb0e3d11ffe9

the bus and having a stranger say, "Wow, that's a big book!" With cryptographically secure page numbers, an eavesdropper can't tell whether it's a long novel or a short pamphlet.

## X.X  Implementation details

Our implementation uses the SHA-256 hash. It features a modular architecture, however, so that a different hashing algorithm can be substituted if desired.

In order to thwart rainbow-table attacks, a salt is appended to the page number before it is hashed. We are currently using the LaTeX job name as the salt. For example, the salt used for this paper was `sigbovik`.

Initially we attempted to implement the SHA-256 algorithm entirely in TeX macros. However, we soon ran into difficulties, including the arcane design of the LaTeX system and memory limitations built into the TeX engine itself. We were surprised to find that TeX does not have native support for bitwise operations on 32-bit unsigned integers. Since we can see no innocent reason to omit such functionality from a typesetting system, we suspect this must be an intentional deficiency added by the NSA in order to weaken cryptographic algorithms in TeX documents. We urge further investigation into this potential security backdoor.

To circumvent this problem, our LaTeX package calls an external Perl script to generate the appropriate SHA-256 hashes from the salted page numbers. This does require that LaTeX (or pdfLaTeX) be called with the `--shell-escape` option.

Our current design requires a fixed upper limit to the page numbers in the document. We have set this limit to 500, but it can be easily increased by editing the Perl script. If the page numbers in a document exceed this value, processing of the document will halt with the error message

```
! tl;dr.
```

The elimination of this fixed limit is left as an improvement for future work.

## X.X  Protecting other numbers

A natural next step after securing page numbers is to secure other sensitive numbers in a document, such as section numbers, figure numbers, and citation numbers. If these numbers are not carefully guarded, they could be used by an attacker to learn partial information about the order of pages.

Our LaTeX package does not yet support securing these numbers with cryptographic hashes, but as a provisional security measure we have included macros to redact these numbers from the document. For instance, the redaction of section numbers can be enabled with the command `\redactsectionnumbers`. Section numbers have been redacted in this paper to illustrate the technique.

0x654c937186a719b8cf18de2956b9b1ec81e41891736f092a7e0a87e9ea0b1c8c

## X.X   Security recommendations

When properly used, cryptographically secure page numbers make it impossible for an adversary to determine the order of the pages of a document without an expensive $\Omega(n!)$ time brute-force attack. However, in order to support this security, it is important for the document to be kept as loose pages (preferably one-sided). We strongly recommend against the use of binders, staples, paper clips, file folders, and other external devices that may inadvertently reveal information about the proper order of the pages. To maximize security, it is best to take all the pages of many documents together and throw them at random into a big sack. Another effective randomization algorithm is to toss the pages down a flight of stairs or off a suitably tall cliff.

The proper disposal of a sensitive document is also important. Some paper shredders, often called *cross-cut* shredders, cut a document in two directions, yielding many tiny bits of paper. On the other hand, other shredders, so-called *strip-cut* shredders, merely slice the document into strips. If a page containing one or more hashed page numbers is fed into a strip-cut shredder in a certain orientation, it is possible that each one of these hash values will end up on a single strip, so a snooper can easily pull out individual strips and read off the complete hash. Of course, the fact that the page number has been hashed means that the snooper will have difficulty gaining any useful information from this, but to maximize security when using a strip-cut shredder it is best to print every page number twice, once horizontally on the page and once vertically, to ensure that at least one of them will be obliterated by the shredding process. (This recommendation applies equally well to all sensitive information in the document, not just the page numbers.) Our LaTeX macro package does not yet support vertically printed page numbers, but we expect that this functionality will become available in the near future.

# X   Conclusions and future work

In this paper we have proposed a new technique for securing page numbers in LaTeX documents using cryptographic hashing algorithms. We introduced the macro package `secure-page-numbers.sty`, which implements this technique. We believe that this package has broad potential to significantly increase the security of page numbers in sensitive documents, and we urge all authors and publishers to adopt the use of secure page numbers as soon as possible.

One potentially fruitful extension to this work, which we have not yet explored, is the use of a public-key cryptosystem such as RSA for the encryption of page numbers in two-way communication. This is likely to be more useful than one-way hashes, since each party will be able to decrypt the page numbers in the document received from the other party.

In accordance with the first of our security recommendations above, it may be beneficial to scramble the pages of the document not only after they have been printed but also in the PDF itself. A LaTeX package to achieve this would

be difficult because of the design of the TeX engine, but we believe it may be possible with the aid of an external tool. As a workaround until this becomes available, we recommend storing large PDFs on a sequence of unlabeled floppy disks, which can then be shuffled together.

# References

[X] Hagenfried, E.W., McPaulsen, F.J., and Svendt, P. Exploiting pseudofrequency eigenquotients to attack page number encryption. *IEEE Trans. Crypto. Splat.* XX(X), pp. 0x2fd5f03a787301a109748c1243f85f6522c3c862be9e1b156c84ef8dfaf9c296–0xbfbd44e30f1d9beef64f9324993b67e99bb1688ed77f474c46e58b2aca9c23dc, November XXXX.

[X] Klein, F. *Über Riemann's Theorie der algebraischen Functionen und ihrer Integrale: Eine Ergänzung der gewöhnlichen Darstellungen*, pp. 0x2fa9b01782ee3680c2682f88fb03913ae03ff998602c90a0f8b2129567589535–0x9d7cf761c701b006aba29b8b4830eb46c78c28d8c4d448b4a5981a37697cb949, XXXX.

[X] Möbius, A.F. *Werke*, vol. X, pp. 0x3292d92ba775c2cac72f8e9091e04b3915d3ca7463c3c7b936ec7bede9933f9e–0x034c4ae2b61541e3a32124700b7c99c63974fb30d353649ec2fa828c5d90ec75, XXXX.

0xc50e747b92c42e0a7c53ac8b0f0536dc170cd2c638b6e9c16e4e79656e2349af

**Claim 1:** *The authors claim to have implemented a crypto library for Standard ML.*

**EVALUATION OF CLAIM:** It was very difficult for the committee to assess this library because we could not even identify what language it was in. Due to the combination of references in German and a .pl file extension, we assumed the library was written in German Prolog. We gave the code to a local German Prolog expert for further analysis, but his evaluation consisted only of the single word "Achtung".

Due to numerous references to Arabic and a wide variety of Arabic numerals in the text, we next tried interpreting the library as Arabic, with equally little success. A senior PC member then pointed out that .pl stands for Programming Language (this is why we keep the programming languages group around) after which we tried running the code as a Program, and finally found success.

The code ran with the desired effect, and the committee was impressed. However, we think you may have gone overboard with your security measures, because the source code itself appeared to be equally encrypted, and even after running it, we have no idea why it worked.

**Claim 2:** *The provided encryption method is robust to Shredders.*

**EVALUATION OF CLAIM:** The artifact committee had actually already unwittingly evaluated this claim when we attempted to shred the submission without reading it. Luckily for all involved, our department has a surprising dearth of shredding equipment, after which we decided to actually read the paper.

We do, however, have an SNES, which we used to evaluate this claim by playing our old Teenage Mutant Ninja Turtles games. When we got to the final battle with Shredder, the submission actually ripped the controller out of our hands and quickly defeated the final boss.

This scheme is **beyond** Shredder-proof.

0x7df8eefe1bc490220b2b34a09be0f925ebcdf1af8cea93a31e972faef5ce5a10

**OVERALL EVALUATION:** This encryption scheme is worth its salt. The artifact provided is of such quality that we extend an invitation to publish an extended article in our Journal: SIGBOVIK Heretics Anonymous Digest: Volume 256.

**ADDITIONAL COMMENTS:**

- The LaTeX implementation included extensive use of CS names. In the final draft of your paper, please elaborate on why names from other scientific fields were not supported and list this as a limitation of your current implementation. That being said, the committee is not concerned by this restriction.

- There was also much discussion of ex-leaders in the LaTeX code. Given that we're in the middle of a heated election year, it may make a more interesting read to focus on current leaders.

0x5a9dcef798e74a263edb8c0341660890e2bd75f295fef29fbfd1f6429cacfe66

# Random Seed Encryption

Alexander R. Frieder*

21,876,528  782,847,088,000†

April 1, 2016

## Abstract

We present a novel form of encryption, wherein the encryption step is computationally expensive while the decryption step is more or less trivial. We will describe how the scheme works, proving its correctness along the way. We will then show that this encryption scheme has many uses, for example, prevention of texting while intoxicated or sender identification over email. We will also acknowledge the limitations imposed upon our beautiful scheme by the cruel harsh mistress of reality. Finally, we will discuss the immeasurable impact this algorithm could, nay, will have upon the world.

## 1   Introduction

It is universally understood among nearly all experts in the field of cryptography [1] that any good cryptographic encoding function has four fundamental properties:

- it is quick to compute the hash value for any given message

- it is infeasible to generate a message from its hash

- it is infeasible to modify a message without changing the hash

- it is infeasible to find two different messages with the same hash

---

*email: alex.frieder@gmail.com

†This is the encryption of the author's name using C's default long-seeded PRNG.

We, however, like to go against the grain in terms of what is "universally understood". Thus, we posed the question: what if the first two conditions were switched?, that is, what if we had a hash function where it was nearly infeasible to compute the hash value, but quick to generate a message from its hash? Thus Random Seed Encryption (RSE)(not to be confused with the lesser known RSA) was born.

We have come up with one such function and will spend the rest of this article analyzing and disucssing it, with applications in encrypting words, though any datatype could theoretically be encrypted.

## 2   Encryption Algorithm

The encryption algorithm takes a word and calculates two associated numbers for it: the length, $\ell$, and a seed, $s$, to a pseudorandom number generator (PRNG) such that $s$, is the smallest seed such that the next $\ell$ numbers generated, when converted to letters, spell the word. These two numbers are mapped to a single number, which is the encryption of the word.

### 2.1   PRNG to Letter Conversion

In general, PRNGs output either a random real number between 0 and 1 or a random integer between 0 and $N$. Any function that takes one of these numbers as input and deterministically returns a letter, where the probability across all numbers in the domain of getting any given letter

is approximately $\frac{1}{26}$ will suffice. However, since constructive proofs are all the rage these days, we will briefly discuss our chosen functions.

To convert a random real $0 \leq r < 1$ into a letter, partition the space $[0, 1)$ into 26 sections, $[0, \frac{1}{26}), [\frac{1}{26}, \frac{2}{26}), \ldots, [\frac{25}{26}, 1)$. If your random number falls into the $i$th section, then your letter is the $i$th letter. This clearly works. Even more trivial is the proof that every letter has an equal probability.[1]

To convert a random integer $0 \leq n < N$ into a letter, take $i = n \bmod 26$ and use the $i$th letter. For $N \geq 2500$, the difference from $\frac{1}{26}$ is less than 1%.[2]

Now we know how to convert random numbers to random letters and we can move on to finding the right random numbers for our specific letters.

## 2.2 Finding the Correct Seed

PRNGs are not actually random and honestly, they are not fooling anyone. PRNGs just produce a very very very very long list of bits that are converted into numbers in some range. And this list of bits is always the same! The trick used is to start at different points in the list so it at least seems random. This starting location is called the seed.

If you set a PRNG to a specific seed and generate a list of random numbers, then reset to the same seed and generate another list of random numbers, the two lists will be identical. Our goal is to find a specific seed that will produce our word.

For example, if you set the default Python PRNG to seed 1,944,062 and generate five random numbers, converting them to letters as discussed above, you get "hello", which is a nice message many people would want to send, probably. How do we find seed 1,944,062?

Well, you can set the seed to 0, see that the word of length 5 generated at this seed is "vtkgn", which is not "hello", set the seed to 1, see that the word of length 5 generated

is "dwtgm", and repeat until you set it to 1,944,062.

Surely this naïve brute force algorithm is just the first in a set of stepping stones to some more intelligent algorithm, no? No. The whole point of a PRNG is that it seems random. Knowing these five letters tells you nothing about the next five, or where any given five letters will be.

There is no way to determine where some given letters will appear in a sufficiently complex (known in the technical world as "cryptographically secure") PRNG without trying every possibility until they are found.[2][3] That is what makes this algorithm so powerful.

## 2.3 Merging the Seed and Length

Having the correct seed, $s$, is not enough; you also need the length of the word, $\ell$. We could of course just encrypt every word as two numbers, but that means we need twice the space. Instead we will use the well-known bijection between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$, $f(x, y) = 2^x(2y + 1)$. We will use $f(\ell, s)$ since $\ell \ll s$ and this function grows exponentially in the first argument, but linearly in the second. Any other bijection between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$ will work, but we use this one since it is clearly the best.[3] Thus, we now have all the machinery necessary to take a word and fully encrypt it into a single number.

## 3 Decryption

Decrypting a word given the encrypted number is extremely easy. Looking at the bijection we used, the length of the target word is exactly the number of times we can divide 2 into the encrypted number. The remaining odd number is one more than twice the seed.

For example, if our encrypted number is 124,420,000, we halve it as many times as we can. 32 divides this number evenly while 64 does not. Thus the length is $\log_2 32 = 5$.

---

[1] Proof left as an exercise for the reader.

[2] Proof left as an exercise for the reader. <sub>But actually it is true.</sub>

[3] This result is probably shown in this citation. It is reallllllly long and we reallllllly do not want to read it. But it is there. we promise.

0x1f84e6ea9f3ec349a9f9279216c443e9f805ec9ddc29d91f51b69c2a72932df1

We are left with 3,888,125. The seed is thus $\frac{3,888,125-1}{2} = 1,944,062$.

Now that we have our seed and length, we set our random number generator to that seed and generate 5 random letters. In this case, we get the letters, in order, "h", "e", "l", "l", and "o", which spells "hello". Thus the number 124,420,000 uniquely encodes the word "hello".

## 4  Discussion

We have fully described how and why this encryption method works, but as with all theoretical algorithms, there are some limitations due to the realities of physics and computers.

Every PRNG has some finite period, after which the PRNG starts repeating values. This places a hard upper bound on the number of words we can encode with a given PRNG. Additionally, we are further bound by the number of unique values we can use for the seed.

For example, when the seed is an integer, commonly represented by 4-byte values, we can uniquely represent $2^{32} - 1 = 4,294,967,295$ seeds and thus words. In this case, it is sufficient to represent all 12,356,631 words of length 6 or less, but not enough to represent all 8,031,810,176 words of length 7. Similarly, using 8-byte longs as the seed we can represent all words of length 13 or less. However, 13 is a big number and a lot of words can be represented using less than 14 letters. In fact, not a single word in this paper has more than 13 letters. So there.

Additionally, there is no guarantee that a given word will actually appear in a PRNG. For sufficiently long words, a given sequence of random numbers might just not be an output of the PRNG. However, since most combinations of letters are not words, this is likely not to be a problem in actuality.

## 5  Applications

Random Seed Encryption clearly has numerous significant uses in modern life. One such use is to prevent drunk texting/messaging. Studies have found that drunk texting friends, former lovers, family members, and even bosses can have terrible terrible effects on one's life.[4] Even worse is drunk posting on Facebook, reddit, or the reader's social media website of choice. Imagine some kind of filter you can turn on, where for the next $k$ hours, anything you want to send to someone or post somewhere has to be encrypted using this method. Of course, no one under the influence could ever manage to correctly encode even a single word! Thus drunk messaging is prevented.

Another possible application is that of sender verification. Many people try and verify their identity over email using PGP encryption, but that is a lot of work for the receiver, and honestly, we all know no one really cares that much. However, sending someone a message under this encryption scheme has a two-fold benefit over PGP: it is much easier to decrypt, so the receiver might actually bother with it, and since RSE is so new, no one else in your social circle except you will even have heard of it, so the encryption is definitive proof that you actually sent the message!

Thus we have demonstrated that not only does RSE have real world applications, but it has numerous benefits over other, more commonly used encryption schemes. We hope this will encourage the cryptographic community to spread the good word about Random Seed Encryption.

## 6  Future Work

There are numerous ways this novel direction of research can be expanded. For example, we would love to consider:

- This algorithm using "true" random number generators. This would obviously complicate matters since seeds no longer exist, but we are sure some bright, spunky undergraduate student could figure out a solution.

- Encryption schemes where both the encryp-

0x5b120f61e213712179c10495814d0a74e99f5f7461a15d0a02bd6ada44a0a758

tion and decryption steps are computationally expensive.

- Alternative encryption functions that are, for example, $\Omega(2^\ell)$.

- Something involving quantum computers working on big data in the cloud

We believe this paper may be the dawn of a new field of computer science, which we have preemptively named the Friederian Sciences.

# 7 Acknowledgements

# References

[1] Wikipedia contributors. "Cryptographic hash function." Wikipedia, The Free Encyclopedia, 14 Jan. 2016. Web.

[2] Röck, Andrea. "Pseudorandom Number Generators for Cryptographic Applications." Thesis. Paris Lodron University of Salzburg, 2005. Web.

[3] Sutner, Klaus, Teaching Professor and Associate Dean for Undergraduate Education, School of Computer Science, Carnegie Mellon University.

[4] Bartholow BD, Henry EA, Lust SA, Saults JS, Wood PK. Alcohol effects on performance monitoring and adjustment: affect modulation and impairment of evaluative cognitive control. *J Abnorm Psychol.* 2012 Feb;121(1):173-86.

# Safety Schools of Thought

0x17eb2ba0b6e2fc7438d64d9fd0036862571ce946e0570f6db15e54fbef93dc26

0x5a0c3dfd5478db4ca01d4c5aa73475f4d95f8ec654d50242a9d4730a995bbdb3

# Reducing the Trusted Constituent Base with the LCF Approach

*or*

## Hillary: The Next 700 Elections

Brandon Bohrer
Carnegie Mellon University
bbohrer@andrew.cmu.edu

Sol Boucher
Carnegie Mellon University
sboucher@cmu.edu

Jos Valdivia
Case Wallet, Inc.
jos@choosecase.com

**Abstract:** In order to form a more perfect union, we address the issue of election robustness. We introduce the LCF (Logic of Congressional F*ckery) approach for minimizing the TCB (Trusted Constituent Base) of an election process. We evaluate our approach on the American presidential election process and show that it compares favorably with traditional approaches for election robustness.

## 1. Introduction

Due to the finite length of the human lifespan, all forms of government presently require a method of repopulating themselves. At present, the most popular method is that of an *election,* an event in which people called *constituents* collectively decide on their rulers with varying levels of freedom. A singular advantage of the Election Method is the illusion of agency granted to the masses, which is widely considered to be the most effective method of suppressing public unrest. The great risk of the Election Method is that is the issue of *robustness***:** in most implementations of the Election Method, there is a risk that the *constituents* will elect incorrect leaders. This had led to the proposal of many interesting election verification methods, but this is still an open problem with much work to be done.

## 2. Related Work

Every nation with an election system has faced this problem, and thus the literature is vast. In this section we outline methods used by various nations.

**Definition 1: Trusted Constituent Base:** We say that an electoral process has a Trusted Constituent Base if $\alpha$ is the fraction of voters that must be trusted to vote correctly to achieve a proper outcome. It has been shown that systems with $\alpha <= 0.01$ can be verified in linear time, but verifying systems with $\alpha > 0.01$ is NP-Hard (it is unknown whether it is also in NP). Thus the goal of verification technique is to achieve $\alpha = 0.01$. Failing that, lower $\alpha$ leads to more tractable verification in practice, even when the problem remains NP-Hard in theory.

## 1.1. United States

The United States, despite its immense wealth and power, has long struggled to verify its election processes. The verification methods currently used in the United States are widely considered insufficient, but are illustrative and form the foundations for more sophisticated

0xd365958785ba3fedfa5b18ed5cb62eaf5eb2bab3d25bf1fa11fae24eaa761f64

techniques:

**First-Past-The-Post:** A common building block of election verification systems is to count votes in strange ways that skew the distribution of power. The most basic method is the *First-Past-the-Post* algorithm, which exploits integrality issues. In a multi-position election, one can easily justify absolute power to whichever contestant achieves a plurality of votes. This method alone achieves α = 0.50. At the time, FPTP was considered a heroic breakthrough, but has long since been pushed to its limits.

**Amphibian Partitioning:** This method, pioneered by Gerald Mander, observes that an election can be partitioned into *districts*, where each district is counted as an individual vote. In an optimal partitioning, an election can be won with (arbitrarily close to) half of all districts, each winning with half of the available results, leading to α = 0.25. But this number belies the true power of the Partitioning approach: in multi-position elections containing *minority constituents* (those who, when adequately partitioned, have a majority in no districts), can effectively reduce the representation factor of a sub-constituency to 0. However, optimizing representation factors is outside the scope of this paper.

**Weighted Partitioning:** In a straightforward extension of Amphibian Partitioning, all districts are given a fixed additional number of votes in order to strengthen the votes of the least populous districts. Since votes in small districts are the more powerful, the verification burden is reduced. By increasing the number of votes uniformly across districts, a semblance of democracy is maintained, minimizing risk of unrest. In principle, this technique can achieve arbitrarily low α-value. In practice, this method has a small effect on α-value, because excessive use creates some risk of public unrest. It is worth noting that, nevertheless, this technique has successfully been used to ensure the correctness of several previous presidential elections, including the 2000 election of George W. Bush, widely considered one of the most difficult verification projects in recent memory.

**Indirection Escape Hatches:** The *Electoral College* includes an escape hatch by which the unified vote of a mere 270 unelected government drones overrides the will of the electorate. Historically, numerous electors have tried to use this mechanism, but have never coordinated well enough to overturn an election. In principle this method has the incredibly low α-value of .0000001, but suffers from a greatly increased risk of political unrest. And this begins to expose an underlying theme in election verification: Simply reducing the number of individuals involved is trivial. The difficulty lies in maintaining stability of government.

**Higher-Order Indirection Escape Hatches:** One approach for reducing unrest is to target less-publicized but equally-important elections. In a two-party system, general elections are effectively useless: the entire political race is decided during primary elections, whose conduct is watched much less closely. Thus there exists an escape hatch for primary elections in the form of Superdelegates (higher-order delegates or HODs), which can vote as they wish, with the same impact as hundreds of thousands of people. In current implementations, this approach

has α-values on the same order as Amphibian Partitioning (since only a constant fraction of votes are assigned to Superdelegates), but when the techniques are combined they can achieve an α-value below .1 without causing noticeable unrest.

## 1.2. China

China has long used B-Trees to obtain incredibly low α-values. The main insight of the Chinese election process is that the standard parliamentary system is a two-level B-Tree with high branching factor, and that the total number of trusted votes is equal to the number of children at the root. By removing the 2-level restriction on the B-Tree, the Chinese system can result in arbitrarily low branching factors, and thus an arbitrarily-low constant number of trusted votes.

While many of the other α-reduction approaches mentioned in this paper run into issues due to civil unrest, the Chinese system seems surprisingly successful, due in large part to a long-standing communalist culture in which many people [这里, Mandarin 1970] see no need for personal agency in the political process as long the resulting government achieves the pragmatic goals they desire for the nation [Mandarin, Gerry 1956].

It appears that here the advantages are not in electoral tricks so much as choice of cultural values. Recent work has attempted to modify the American election process by importing sufficient numbers of Chinese people [CMU Admissions, 2016], but it is unclear whether this has had the intended effect.

## 1.3. Russia

Russia takes a completely different—and frankly somewhat counterintuitive—approach to reducing the TCB: reduce the trusted constituent base by increasing voter turnout. While the >100% turnouts often observed in their larger cities might initially suggest an increased α, we observe that a careful implementation results in the opposite effect. Consider that, by the pigeonhole principle, an election that turns out $n$ more voters than actually exist must incorporate at least $n$ votes corresponding to constituents who have already voted. We define the term *doppelganger* to refer to such constituents. Given that a state, through its voter registration process, is able to select which doppelgangers to admit, a successful implementation of this *superturnout* scheme will admit almost exclusively trusted doppelgangers. We refer to the set of all trusted doppelgangers as the *doppelgang* because each of its members will vote in the same way; with such organization, it's easy to show that a doppelgang accounting for a $p$% increase in turnout directly reduces α by $p/100$.

## 1.4. North Korea

Eternal President Kim Il Sung started the Great Democratic People's Leadership Verification Project in 1948, preceding other founding verification projects such as AUTOMATH by several decades. The prosperity of the Democratic People's Republic has provided funding for the Verification Project far exceeding that available in imperialist nations, and thus enabling a

tremendous scope and leading to verification results of which the imperialists are envious to this very day.

After an exploratory period of several years, the Verification Project found focus in the philosophy of Juche, focused on a *trinitarian* ideal based on the following principles

1. Proof automation (자주 or 自主)
2. Representation Independence (자립 or 自立)
3. ~~Masturbation~~ Defensive Design (자위 or 自衛)

This led to promising initial results. After the death of the Eternal President, Eternal Secretary Kim Jong-Il added his military-first or songun (선군 or 先軍) policy which supported a *militant* attitude to the correctness of elections. Now a nationwide priority, Kim Jong-Il built Kim Il-Sung Memorial Verification Centers in every school in Choson. By verifying the process from birth, North Korea has achieved an α-value of 0. However, it should be noted that our savage imperialist nation is far too destitute to engage in such an exhaustive verification program, and thus it is the goal of this paper to achieve a fraction of the North Korean result at a more affordable cost.

## 2. Approach

**The LCF Method (Logic of Congressional F*ckery):** The LCF method is based on a fundamental observation: The tension between the desire for a rich verification language and a small Trusted Constituent Base is best resolved by defining powerful special-purpose constructs in terms of extremely general base constructs.

In particular, the LCF uses a single building-block, that of *representative* democracy. Our culture holds democracy sacred, and thus the authors postulate that (a) any approach not based in democracy is bound to produce unrest and (b) any approach that *is* based on democracy has a *leg up*.

Our *representative* approach is based on our analysis of [Kim 48]'s Representation Independence approach. We know from experience that Representation Independence (developing a culture in which constituents do not care at all whether their interests are represented by the government), while effective, is prohibitively expensive. Since reducing cost is primary goal, we take a *representation-aware* approach (constituents receive some form of representation, and are made extremely **aware** of all representation they receive, but we develop methods to prove that the available representations do not interfere with correctness).

On top of the basic functionality of representative democracy, we build the following tactics:

**Pork-Barrelling:** Pork-Barrelling is an **awareness-based** tactic. In such a vast nation, it is difficult to be aware of every political event across the nation, but quite feasible to be aware of

local political events. This observation leads to the result that election results depend only on a local view of politics, formalized below.

**Definition 2:** A congress is *locally sound* if each district's local political outlook is positive.
**Definition 3:** A congress is *globally sound* if every congressperson is elected correctly.

**Theorem 1:** All locally sound congresses are globally sound.

It is a well known fact in verification that reducing complex global concerns to simple local concerns makes otherwise intractable problems feasible. Pork-barrelling works by reducing the absolute number of contributing factors in an individual election, making certain other techniques more effective.

**Curtain-closing:** It was observed in our introduction that partitioning schemes are vulnerable to public unrest if used excessively. The main problem is the public attention paid to the shape of voting districts and redistricting process. A main contribution of the work is that closed-curtain redistricting processes, by their very nature, prevent any change in public opinion:

**Theorem 2 (The Public Respects the Redistricting Committee):** Public opinion is independent of the redistricting action of a secret committee.
**Proof Sketch**: Secret committees have existential type. It follows from Reynolds' Parametricity Theorem that a voter cannot depend on implementation details of the committee, such as their redistricting results. **Qed.**

## 3. Evaluation

We evaluated the LCF approach by implementing an electoral proof assistant named Hillary. The Trusted Constituent Base resides entirely in the Hillary/Purge segment of the assistant, consisting only of approximately 8000 Washington DC residents. The vast majority of the code base, used for practical election verification, resides in the Hillary/HEIL framework (Higher-Electoral Induction Logic). Using Hillary/HEIL we have verified the results of several state primaries with a short-term goal of a complete verification of the Democratic presidential nomination, which appears to be well-underway, largely because the unrest-abating tactics such as Pork Barrelling allowed us to greatly increase our usage of the existing Higher-Order Indirection Escape Hatches technique. Upon verifying the nomination we wish as future work to verify the general election, but we make no claims of that ability at this time.

Our collaborators are currently implementing a second proof assistant in the LCF style called HEIL Light, whose Trusted Constitution Base consists of only 400 constituents. However, as of this writing they have verified only local elections, and it is unclear whether anyone but a supreme expert is capable of scaling their proof assistant to a general election.

# UNPRL: NUPRL PROOF RE-EVALUATION LOGIC

RYAN KAVANAGH

ABSTRACT. The UNPRL proof assistant is introduced. UNPRL generalises the Nuprl proof assistant's client-server architecture, thereby increasing the system's fault tolerance. UNPRL's proof generation system far surpasses the current state of the art in artificial intelligence, *a fortiori* surpassing the naïve proof search provided by other proof assistants. Implementing UNPRL is discussed.

## 1. INTRODUCTION

Proof assistants have become an important tool in the programming language theorist's toolbox. Recently, they have found applications beyond the study of formal systems and the formalisation of mathematics. For example, they have been applied to verifying the safety of cyber-physical systems [Pla10] and in creating verified software toolchains [Con16]. Though we find this trend encouraging, we believe the difficulty and tedium associated with using proof assistants impedes their widespread adoption. To allay these burdens, we propose several improvements to the Nuprl proof assistant described in [Con+85].

Nuprl is an influential proof assistant originally developed in the 1980s and it is still under development today. Despite its longevity and success, it suffers from using a peculiar closed-source client-server architecture. A single server instance is known to exist and is located at Cornell, and one gains access to it by e-mailing a project member. Though this client-server paradigm may on the surface appear unusual, it is perfectly reasonable: operating Nuprl utilises scarce resources and is inherently Oz-like. Indeed, it is widely believed[1] that whenever a user connects to the server, an alarm goes off at Cornell and the first author of the Nuprl book [Con+85] (hereinafter referred to as "the man behind the curtain") rushes to the terminal in the server room. Then, as the remote user enters his proofs, the man behind the curtain quickly checks them and informs the user of their correctness. Though the mental prowess underlying Nuprl is indisputable, the astute reader will immediately see the unsustainability of the Nuprl architecture.

We introduce a new proof assistant called UNPRL that generalises Nuprl's client-server-proof checker model, thereby building on the strengths of Nuprl while addressing its aforementioned Achilles heel. Moreover, UNPRL provides a proof generation mechanism far surpassing the state of the art in artificial intelligence, *a fortiori* surpassing the naïve proof search provided by other proof assistants. We examine

[1]Private communications at POPL'16.

0xfc9093631e80c59e4c4ccf5175114fd831a7421131b0ff66c2fa02a487591e6b

these improvements in section 2 and discuss several techniques that may lead to performance increases in section 3. Implementation details are discussed in section 4.

## 2. Work Distribution and Proof Generation

Rather than assume a single "man behind the curtain", Unprl generalises Nuprl's client-server-proof checker model to support multiple "people behind the curtain". In particular, we assume an odd number of undergraduate students $u_1, \dots, u_n$ to be behind the curtain (the reader is referred to Appendix A for assistance in determining the parity of a given $n$). Unprl's client-server-multiple proof checkers architecture is depicted in Figure 1.
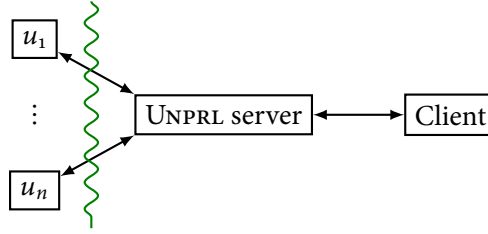


Figure 1. Unprl's client-server-multiple proof checkers architecture.

We now pass to study the benefits of this simple and obvious generalisation. These benefits are captured by Unprl's two main *modes* of operation.

Unprl's main mode is its *proof checking mode*. In this mode, the server idles until it receives a theorem statement $T$, a proposed proof $P$ for the theorem, and an acceptance threshold $0 \le A \le 1$. Upon their receipt, the Unprl server transmits a copy of $T$ and $P$ to each student behind the curtain. Each student then either replies to the server that they accept the proof, or they reject it and provide the server with a proof of $\neg T$. If at least $An$ students accept the proof, then the server marks the proof as correct and notifies the client that $P$ is a proof of $T$. Otherwise, there was disagreement and we must re-evaluate the proof. We proceed by breadth-first search to reach consensus as follows. The existence of disagreement implies the server received counter-proofs $P_1, \dots, P_s$ for some $s \ge 1$. The server enqueues the triples $(\neg T, P_i, A)$ for $i = 1, \dots, s$. We dequeue a triple $(\tau, \pi, A)$ and reenter proof checking mode, checking theorem statement $\tau$ with proof $\pi$ and threshold $A$. If $\pi$ is accepted for $\tau$, we apply double-negation elimination to $\tau$ to determine if we should accept $T$. If we accept $T$, we inform the client that $\hat{\pi}$ is a proof of $T$, where $\hat{\pi}$ is $\pi$ plus any double-negation eliminations required to transform $\pi$ from a proof of $\tau$ to a proof of $T$. If we reject $T$, we inform that $\hat{\pi}$ is a proof of $\neg T$, where $\hat{\pi}$ is $\pi$ plus any double-negation eliminations required to transform $\pi$ from a proof of $\tau$ to a proof of $\neg T$. Otherwise, we received counter-proofs $P'_1, \dots, P'_t$ to $\tau$, so we enqueue the triples $(\neg \tau, P'_i, A)$ and continue with our search. Upon termination, the client will have a verified proof of either $T$ or $\neg T$.

Unprl's second mode is its *proof generation mode*. In this mode, the server idles until it receives a theorem statement $T$ and acceptance threshold $0 \le A \le 1$. Upon

receipt, the UNPRL server switches into proof checking mode and attempts to check the proof triple ($T$, **true**, $A$), where **true** is any tautology or always provable sentence. When the proof checking mode terminates, it will have returned a theorem-proof pair, providing either a verified proof of $T$ or a verified proof of $\neg T$. These results are conveyed to the client. The deluxe version of UNPRL provides the option to perform proof search by problem set, though the latency is obviously increased.

## 3. Analysis and Incentivisation

Though a marked improvement on Nuprl, UNPRL suffers from several deficiencies that we address in this section.

3.1. **Proof Checking Mode.** Though UNPRL's run-time *bus factor* can be made arbitrarily large, its accuracy may not increase correlatedly. Moreover, it is not possible in general to estimate the likelihood that an arbitrary group **U** of undergraduate students $u_i$ accepts an incorrect proof. Indeed, assuming **U** is composed predominantly of freshmen, the probability that the theorem statement and proof pair given in Figure 2 is accepted is believed to be arbitrarily close to 1, also independently of the acceptance threshold specified. However, the theorem is clearly false of the classical reals and the proof should have been rejected. Thankfully, this does not mean we should abandon UNPRL. Indeed, if **U** is composed predominantly of students having taken a course in analysis, the likelihood of the proof pair being rejected is close to 1, independently of the acceptance threshold specified. This example illustrates the difficulty in quantifying the probability that proofs accepted by UNPRL are correct. However, we believe that this example demonstrates the need to ensure that the $u_i$ be uniformly sampled from the undergraduate student population.

To encourage "correct" results, we propose rewarding the proof checkers when they provide "correct" responses.

We assume the client can allocate $r > 0$ reward resources[2] to verifying $k$ theorems using $n$ undergraduates. For each theorem $T_i$, $i = 1, \ldots, k$, let $c_{ij}$ be the number of rounds in which undergraduate $u_j$'s response to the server agreed with the final result. In other words, let $c_{ij}$ be the number of times $u_j$ accepted a proof of a $\tau$ double-negation equivalent to $T_i$ if $T_i$ was accepted or the number of times $u_j$ accepted a proof of a $\tau$ double-negation equivalent to $\neg T_j$ if $T_j$ was rejected. Then, at the end of the theorem proving session, each undergraduate $u_j$ receives

$$\frac{\sum_{i=1}^{k} c_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij}}$$

of the reward $r$. Key to our strategy is that the reward received corresponds to a fraction of all correct answers in the system, and so the more a given undergraduate surpasses his peers, the greater his reward. It is thus against the undergraduates' individual best interests to collude and secretly agree to accept all proofs.

To further counteract any chance of collusion, we suggest running two instances of UNPRL in parallel, each receiving $\frac{r}{2}$ resources. Then, for each $T_i$, randomly assign

---

[2]Experience has shown food to be the most effective form of reward.

**Theorem 1.** *The set of (classical) reals $\mathbb{R}$ is a singleton set.*

*Proof.* First observe first that in $\mathbb{N}$,

$$0 = 0^2 = (1-1)^2 = (1 + (-1))^2 = 1^2 + (-1)^2 = 1 + 1 = 2.$$

We proceed by induction on $n$ to show that either $n = 0$ or $n = 1$. Indeed, the base case is trivial, so assume the claim holds for a given $k$ and consider $n = k + 1$. By the induction hypothesis, either $k = 0$ or $k = 1$. In the first case, we have $n = k+1 = 0+1 = 1$, and we're done. Otherwise, $k = 1$ and $n = k + 1 = 1 + 1 = 2 = 0$, and we're done.

Now, consider the fractions $\mathbf{0} := \frac{0}{1}$ and $\mathbf{1} := \frac{1}{1}$. We wish to show that they are equivalent, written $\mathbf{0} \sim \mathbf{1}$. Observe that $\mathbf{0} \sim \frac{2}{2}$, since $0 * 2 = 2 * 1$ (recall, $2 = 0$). Moreover, $\frac{2}{2} \sim \mathbf{1}$, since $2 * 1 = 1 * 2$. So by transitivity of $\sim$ (see [Lan09, Theorem 39]), we have $\mathbf{0} \sim \mathbf{1}$. We now show that for any arbitrary fraction $\frac{m}{n}$, $\frac{m}{n} \sim \mathbf{0}$. By the above, either $m = 0$ or $m = 1$. If $m = 0$, then $\frac{m}{n} \sim \mathbf{0}$ because $m * 1 = 0 = 0 * n$. In the second case, we must have either $n = 0$ or $n = 1$. If $n = 1$, then $\frac{m}{n} \sim \mathbf{1}$, because $m * 1 = 1 * 1 = 1 * n$, and so by symmetry [Lan09, Theorem 38] and transitivity of $\sim$, we have $\frac{m}{n} \sim \mathbf{0}$. Otherwise, we fall into the case of $m = 1$ and $n = 0$. Then by [Lan09, Theorem 40], $\frac{m}{n} \sim \frac{m}{n} \times \frac{n}{n} = \frac{0}{n}$. Then by the above $\frac{0}{n} \sim \mathbf{0}$, and we're again done by transitivity. Following Landau [Lan09, Definition 16], we thus conclude that there is a unique rational number: the set of fractions equivalent to $\mathbf{0}$.

This implies that there is a unique Cauchy sequence of rational numbers, namely the constant sequence of $\mathbf{0}$s. But its limit with respect to the standard metric is $\mathbf{0}$, and so the rationals are a complete metric space. But $\mathbb{R}$ is defined to be the completion of $\mathbb{Q}$, and so we conclude $\mathbb{Q} = \mathbb{R}$. But $\mathbb{Q}$ is a singleton set, so so is $\mathbb{R}$. $\qquad\square$

FIGURE 2. A proof freshmen dream about.

$T_i$ to one and $\neg T_i$ to the other, and withhold compensation for the $T_i$ on which the two instances disagree. Not knowing if their instance received the true version of a theorem but knowing that they must reach the same conclusion as the other instance, the checkers will have no choice but to actually check the proof. Recent advances in friendship logic appearing at this conference may shed light on whether this overhead is truly necessary.

3.2. **Proof Generation Mode.** Given that the proof generation mode is a special case of the proof checking mode, it is sufficient to adapt the above incentivisation scheme. Suppose the client has $r$ reward resources to allocate and $k$ theorems to prove. Then allocate $\frac{r}{2}$ resources to the incentivisation of the $k$ top-level proof checking

calls. Additionally, reward each $u_i$ with $\frac{p_i}{2k}$ of the reward, where $p_i$ is the number of final accepted proofs that $u_i$ provided.

## 4. Implementation

Implementation is left as an exercise for the reader. Readers are especially encouraged to verify their implementation using Unprl.

## 5. Related Works

Many proof assistants have been introduced over the years, ranging from Edinburgh LCF in 1979 [GMW79] to Nuprl [Con+85] to Coq [Theo4]. These have lead to remarkable achievements of verified mathematics. For example, the Feit-Thompson Odd Order Theorem in Coq was recently verified in Coq [Gon+13]. While not seeking to diminish the works on which Unprl builds, we believe Unprl is a significant improvement on its predecessors.

First, Unprl's architecture does not shackle the client to any given set of axioms, assuming the axioms permit double-negation elimination.[3] In other words, we provide clients with the freedom to specify the axiomatic framework in which they wish to verify their proofs and prove their theorems. Our legal counsel has urged us, however, to caution users of Unprl against using the patented axiomatic system underlying Estatis Inc.'s *Falso HyperVerifier* [Est].

Finally, we allay concerns that may have arisen following section 3 regarding the risk of accepting an incorrect proof. We believe that this risk is no greater than the one inherent in other state-of-the-art proof systems. For example, it is possible to prove False in Coq 8.4.5 by exploiting a bug in the `vm_compute` command when there is a type with more than 255 constructors [DPC15]. Moreover, we believe that the incentivisation scheme we proposed is sufficient to counteract the risks.

## Acknowledgements

The author gratefully acknowledges various insightful conversations on the mechanics of Nuprl with Olivier Savary Bélanger and other POPL'16 attendees.

## Appendix A. Odd Integers

Determining the parity of integers is a difficult task. However, as a service to the reader, we provide the following functions. The function `returnTrueIfOdd`, implemented in Standard ML, returns true if applied to an odd integer. As a supplement, the function `returnTrueIfNotOdd`, also implemented in Standard ML, returns true if applied to an integer that isn't odd. The author has formally verified that these functions satisfy their specifications.

```
fun returnTrueIfOdd    (n : int) = true
fun returnTrueIfNotOdd (n : int) = true
```

---

[3]A simple work-around to the (not not absurd) objections[4] to double-negation elimination involves simply looping until a direct proof to *T* accepted by *An* of the students is produced.

[4]Those who object to double-negation elimination *cannot* be offended by this remark if they are at all consistent.

## References

[Con+85]   R. L. Constable et al. *Implementing Mathematics with The Nuprl Proof Development System*. Prentice-Hall, 1985.

[Con16]    DeepSpec Consortium. *DeepSpec: The Science of Deep Specification*. http://deepspec.org/. 2016.

[DPC15]    Maxime Dénès, Pierre-Marie Pédrot and Guillaume Claret. *clarus/falso: A proof of false*. 2015. URL: https://github.com/clarus/falso.

[Est]      Estatis Inc. *Falso HyperVerifier and HyperProver*. URL: http://inutile.club/estatis/falso/.

[GMW79]    Michael J. Gordon, Arthur J. Milner and Christopher P. Wadsworth. *Edinburgh LCF*. Vol. 78. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979. ISBN: 978-3-540-09724-2. DOI: 10.1007/3-540-09724-4.

[Gon+13]   Georges Gonthier et al. 'A machine-checked proof of the odd order theorem'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7998 LNCS (2013), pp. 163–179. ISSN: 03029743. DOI: 10.1007/978-3-642-39634-2_14.

[Lan09]    Edmund Landau. *Foundations of Analysis*. Trans. from the German by F. Steinhardt. 3rd ed. Reprint of the 1966 edition. Providence, Rhode Island: AMS Chelsea Publishing, 2009. ISBN: 0-8218-2693-X.

[Pla10]    André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Heidelberg: Springer, 2010. ISBN: 978-3-642-14508-7. DOI: 10.1007/978-3-642-14509-4.

[Theo4]    The Coq development team. *The Coq proof assistant reference manual*. Version 8.0. LogiCal Project. 2004. URL: http://coq.inria.fr.

*E-mail address*: rkavanagh@cs.cmu.edu

COMPUTER SCIENCE DEPARTMENT, CARNEGIE MELLON UNIVERSITY, PITTSBURGH, PA 15213-3891

0x17f7e02851f9161d97f5ac98ac11968f2d8a4fac0e0f045345702cba5e0e3272

# Type-Safe Friends Will Never Hurt You

Brandon Bohrer

Carnegie Mellon University

## 1 Introduction

In recent years, the activity commonly known as "friendship" has undergone a precipitous rise in popularity, with recent census numbers suggesting that almost 1 in 3 Americans has friends. As documented in the extensive literature, friendship carries with it a plethora of dangers beginning unpaid lunch debts and culminating in eventual death, even including in some cases situational comedy [2].

Despite extensive efforts to curb this deadly activity, friendship rises across the US and across the world. In an acknowledgement that friendship cannot merely be proscribed by stuffy academics, this stuffy academic applies a formal verification methodology to rule out dangerous friendships while allowing benign friendships to survive.

## 2 A Taxonomy of Friendships

In order to develop a practical logic for the verification of friendships, one must first categorize the friendships that occur in everyday life. While he is ashamed to admit it, due to the much-maligned American "friendship culture", the author has found himself in a number of so-called friendships. So many, in fact, that one may even call him a "friendxpert". Such a friendxpert, in fact, that he can tell you inventing such words only increases your chances of being accidentally friended. Such are the risks of science.

The most basic ontology consists of Good Friends, Bad Friends, Facebook Friends, and $\gamma-$Friends.

### 2.1 Good Friends

The best-known and most prized friend is the Good Friend, and a primary goal of friendship research is the identification and preservation of Good Friends. Extensive anecdotal evidence shows the author is a Good Friend. Examples of Good Friend activities include:

- Providing camaraderie at social events in proportion to the quantity of alcohol provided.

- Not interrupting others with questions about homework when I can tell they're trying to focus on research.

0xe247c767316b8d151a66230cbfbe06206528364b51b60e4aaa3b58f76bb092ad

- Looking the other way when undesired guests are invited into shared working spaces at 3am right in the middle of working hours.

- Ordering a pizza for the whole office and not complaining when others forget to pay their share

- Providing emotional support for common problems like lack of confidence in one's work, no matter how esoteric and detached from reality.

- Selflessly sharing whiteboard markers even with people who insist on rubbing their greasy monkey paws all over the whiteboard instead of using the eraser like the thumb-opposing human they are.

## 2.2   Bad Friends

Bad Friends, while nearly as common as Good Friends, are thoroughly despised and are to be avoided at all cost. Despite the numerous benefits of avoiding Bad Friends, prior works have unilaterally failed to do so. The author has much experience with Bad Friends, whose properties include:

- Monopolizing every writing surface within a 10-office radius.

- Using valuable working space as storage for oversized animatronic skulls.

- Treating the office minifridge like their personal minibar.

- Tyranically vetoing all reasonable efforts to acquire matching "Office-mates for life" tattoos.

Such a list could fill several PhD theses, and in fact is the subject of theses by such luminaries as [9] and [5]. For brevity we give only the short list above, even though we could say much, much more.

## 2.3   Facebook Friends

The limited expressive power of strong friendship models led to the development of weak friendship models, starting with the foundational development of partial friendship spaces by [1] and made practical by the introduction of face-oriented social indexing in [12] for which this style of friend is named.

Facebook Friends have numerous useful properties and practical applications:

- Minimal runtime cost, with hundreds of Facebook Friends often costing less than an individual Good Friend

- Enabling work-efficient work-stealing work-avoiding algorithms

- Weak bisimulation with Good Friends

- Constant-time stalking of unrealistic $\gamma$-friends (see below)

0xe371bfdf833aec58b25850d245f77b5958a00d420c9aeae51179a315d4f82ede

$$\frac{\Delta \vdash \phi \ \textbf{valid}}{\Delta ; \Gamma \vdash \Box \phi \ \textbf{true}} \qquad \frac{\Delta ; \cdot \vdash \phi \ \textbf{true}}{\Delta \vdash \Box \phi \ \textbf{valid}} \qquad \frac{\Delta \vdash \phi \ \textbf{true} \quad \Gamma \vdash \psi \ \textbf{true}}{\Delta ; \Gamma \vdash \phi \otimes \psi \ \textbf{true}}$$

$$\frac{\Delta \vdash \psi \otimes \psi' \ \textbf{true} \quad \Gamma, u : \psi, w : \psi' \vdash \psi \otimes \psi' \ \textbf{true}}{\Delta ; \Gamma \vdash \phi \ \textbf{valid}}$$

$$x : \gamma\text{-Friend} \multimap \Diamond !x \qquad \Box(\neg\textsc{Give-Up} \ I \ U)$$

$$(\textsc{In-Office} \ \textbf{Evan}) \multimap \neg(\textsc{Sound} \ \textbf{Evan})$$

Figure 1: Selected Rules and Axioms

Furthermore, due to their extensive knowledge of Facebook Walls, the Facebook Friends are being considered for important applications in national security [3].

### 2.4 $\gamma$-Friends

The most dangerous friend is the $\gamma$-friend. The $\gamma$-friend is a generalization of the classical concepts of boyfriend and girlfriend. The $\gamma$-friend is an incredibly general concept which can encode esoteric genders including: dragon, code, space, squid, dog, cat, demon, angel, mermaid, lycanthrope, celestial, vampire, pumpkin and even *ceiling fan* [6].

The sheer generality of this relationship makes it of great interests to academics, yet analysis of $\gamma$-Friends has long been considered the holy grail of Friendship Theory, being the subject of a plurality of publications in the BMG, EMI, Universal and Warner conferences, including a majority of Grammy winners. Despite such an exhaustive literature, little inroads have been made, as outlined in the retrospective "What is Love?" [4].

The author must concede he is no expert in this branch of Friendship Theory, though he has heard extensive rumors that such friendships exist.

## 3 Modal Linear Friendship Logical Framework

In the present work we focus on the analysis of Good Friends and Bad Friends. The examples given in the previous section have a common theme of reasoning about situations involving change, possibility and certainty, but are otherwise quite varied. For this reason our logic extends previous dual-intuitionistic linear logics with modal operators $\Box\phi$ and $\Diamond\phi$ which say a formula $\phi$ holds in all or in some future world, respectively. To ensure sufficient generality we take a logical framework approach, where new propositions and axioms can be introduced as needed.

0x55f60d32fcb2c3530007011f94951d1464756a1f5b65f4a1c23f9305de965b2d

# 4   Case Study: Office Relations

A common friendship scenario is that of the graduate school office mates. The many hours spent in the office each day make the office mate bond one of great importance, but due to the randomness inherent in the process, the risk of a Bad Friend is alarmingly high. In this section we use MLFLF to verify the safety of one student and the unsafety of their office mate.

In the first scenario we verify a common safety property: Paying debts that are owed. We assume the following axioms:

$$A1: \; \Box(\text{Owes } P \; N \multimap \text{Pays } P \; N) \multimap \text{Safe } P$$

$$A2: \; \Diamond\text{Spins} \multimap \Diamond P \multimap \Box P$$

$$A3: \; \Box\text{Spins} \multimap \Box P \multimap \Diamond P$$

$$A4: \; \Diamond\text{Paid Brandon } 10$$

$$A5: \; \Box\text{Owes } P \; N \multimap N = 10$$

$$A6: \; \Box\text{Spins}$$

$$A7: \; \text{Paid Brandon } 10 \multimap \text{Pays Brandon } 10$$

For our second scenario we prove a common unsafety property: Theft of valuables, under the axioms

$$A1: \; (\text{Friend-Of A B} \otimes \text{Owns A X} \otimes \text{Takes B X}) \multimap \text{Unsafe B}$$

$$A2: \; (\text{Owns A X} \otimes \text{On Y X}) \multimap \text{Owns A Y}$$

$$A3: \; \text{Friend-Of Brandon B} \multimap \text{Takes B X}$$

$$A4: \; \text{On Paper Table}$$

$$A5: \; \text{Owns Brandon Table}$$

$$A6: \; \text{Friend-Of Brandon Evan}$$

4

**Proof 1: A Safe Friend**

$$
\cfrac{
  \cfrac{A4}{\Diamond\text{Paid Brandon}}
  \quad
  \cfrac{
    \cfrac{A6}{\Box\text{Spins}}
    \quad
    \Box\text{Spins}, \Diamond\text{Paid Brandon} \vdash \Box\text{Paid Brandon}
    \quad A2
  }{\Diamond\text{Paid Brandon} \vdash \Box\text{Paid Brandon}}
}{
  \cfrac{
    \cfrac{\Box\text{Paid Brandon}}{\text{Paid Brandon}}
  }{
    \text{Safe Brandon}
  } \; A1
}
$$

$$
\cfrac{A5}{
  \cfrac{
    \text{Paid Brandon}, \text{Owes Brandon } N \vdash N = 10
    \quad
    \cfrac{A7}{\text{Paid Brandon}, \text{Owes Brandon } 10 \vdash \text{Pays Brandon } 10}
  }{\text{Paid Brandon}, \text{Owes Brandon } N \vdash \text{Pays Brandon } N}
}
$$

$$
\cfrac{
  \cfrac{
    \cfrac{\text{Owes } P\ N \vdash \text{Pays } P\ N}{\text{Owes } P\ N \multimap \text{Pays } P\ N}
  }{\Box(\text{Owes } P\ N \multimap \text{Pays } P\ N)}
}{}
$$

**Proof 2: An Unsafe Friend**

$$
\cfrac{
  \cfrac{A6}{\text{Friend-Of Brandon Evan}}
  \quad
  \cfrac{A5 \quad A4 \quad A2}{\text{Owns Brandon Paper}}
  \quad
  \cfrac{A3 \quad A6}{\text{Takes Evan Paper}}
}{
  \cfrac{\text{Friend-Of Brandon Evan} \otimes \text{Owns Brandon Paper} \otimes \text{Takes Evan Paper}}{\text{Unsafe Evan}} \; A1
}
$$

0xe1f0dd0a655917d466ff4e85cbfc29da284ae11f30aa020748c9974d25ef938f

# 5    Formalization

We have developed an extensive formalization of the MLFLF framework in the linear logic proof assistant Cwelf, available upon request. We show the soundness of all inference rules and validity of all axioms presented in this paper, along with many more, as well as all of our examples. Further case studies for friendship logic formalized in Cwelf will be presented in follow-up paper.

Building on the successes of MLFLF, we show that Cwelf itself is sound from within Cwelf, following in the footsteps of [8].

# 6    Future Directions

While this work has developed an effective framework for the verification of Good and Bad friends, it can say little about Facebook Friends and even less about $\gamma$-Friends. We wish to address these limitations in future works. Because many properties of Facebook Friends seem to involve evaluation cost, we wish to develop a cost semantics for friendship. In order to tackle the complexity of $\gamma$-Friends, we wish to employ automation in the form of linear logical frameworks.

> Before you can love another, you must love your Celf.
> — Chris Martens

Furthermore, while it is extremely useful to verify whether friends are safe, this provides us little recourse in the all-to-common case where a friend is, in fact, not safe. Therefore we wish to build upon our experience in friend verification to perform *friendship synthesis*: the automatic generation of provably-safe friends from a formal specification.

# 7    Related Work

To our knowledge, the only other attempt at formal verification of friendship properties is the set-theoretic approach taken in "You've Got a Friend in Me" [7]. However, this approach has never been formalized and does not express state as easily as our linear-logical approach does.

The complexity of friendship has been studied more extensively than formal verification. In particular, it has been shown that friendship can be detected in $O(n^2)$ time with high probability in some important special cases [11]. Hardness results have been show for a number of key friendship problems: Most famously it was shown that optimizing the size of social networks is NP-hard by a reduction from the max-clique problem [10].

0xa15f00c976f2c063634cd973f13a3a10fef6b39d1b271989c9320d6a230f30fb

# References

[1] Tom Anderson. Whose Space is it Anyway? - Friendship Spaces Are Just Octonion Lattices. *MySpace, Inc.*, 2003.

[2] Jennifer Aniston. F.R.I.E.N.D.S. *ACM SIGPLAN Notices*, 1994.

[3] Donald Drumpf feat. Oasis. Making America Great Again: The Wonderwall Approach. *The Campaign Trail*, 2016.

[4] Alexander Nestor Haddaway and Ligia Nestor. What is Love (and Why Can't We Model-Check it?). *The Album*, 1993.

[5] Robert Harper and Alanis Morrisette. Compiling Polymorphism Using Intensional Type Analysis. In *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '95, pages 130–141, New York, NY, USA, 1995. ACM.

[6] David Karp and Michael Rabin. Gender Identification is NP-Complete. *Theoretical Gender Studies*, 2016.

[7] Buzz Lightyear and Randy Newman. You've Got a Friend in Me. *Toys and Their Stories*, 1996.

[8] Chris Martens and Karl Crary. LF in LF: Mechanizing the Metatheories of LF in Twelf. In *Proceedings of the Seventh International Workshop on Logical Frameworks and Meta-languages, Theory and Practice*, LFMTP '12, pages 23–32, New York, NY, USA, 2012. ACM.

[9] Taylor Swift. Picture To Burn. *Sound Emporium*, 2006.

[10] Kanye West and Jay Z. Clique. *Summers and their Cruelty*, 2012.

[11] Jack White and Meg White. We're Going To Be Friends (in Due Polytime). *White Blood Cells*, 2002.

[12] Mark Zuckerberg. Factoring N-Dimensional Friendship Spaces with Faces. *Facebook, Inc.*, 2004.

0xb6836b960d22ce4d3703dda63d3456fdca94292f5e20f04a300fdb607a49d3a7

**Stefan Muller, 563 Facebook friends**
**Rating:** ☹
**Confidence:** $\pi$

This paper aims to solve the long-standing problem of type theory nerds not having friends, by means of a modal logic for friendship verification. This intuitively seems like the right way to go about making friends, so the committee was initially very excited by this submission.

Unfortunately, the submission failed to live up to its promise. There are many errors in the two provided proofs, and clear typos in the provided axioms. One wonders whether the author should have spent more time checking the submission and less time out with friends. As an example, the variable $X$ in axiom A3 of the second proof is unbound and can be instantiated with anything, leading to the very suspicious axiom that being friends with Brandon requires taking all of his stuff. This seems like a strange condition for friendship. The Program Committee's expertise on the subject of friendship is admittedly limited, so we consulted an anonymous external reviewer who had the following to add:

> Ugh, I can't work when Brandon's in the office. He spends all of his time writing meaningless inference rules about friendship on the board, and if I try to use any of the board space, he yells something about me being a bad friend. He keeps ordering pizza and hinting that we should chip in even though no one else is hungry. Oh, also he keeps insisting that my desk is his and putting his stuff there, and when I take some of his papers off of my desk and move them to his desk, he accuses me of stealing his stuff.

Given our initial evaluation of the paper and the external feedback, the committee is forced to take the unusual step of deciding that the author is a Bad Friend. If the errors are fixed, this might be a decent submission for some other venue, but a prestigious conference like SIGBOVIK can't accept submissions from Bad Friends.

# Overcomplexity Theory

0x021c60530cbc7e7ee7adfffafdec6c057e0bb62352befea46fd8e3dc5ab62f28

0xa253258f485a8b431e4645dbf03a7c8ec8a6341b729e7045422192a1b60c2dd1

# The Computational Complexity of Chinese and Italian Noodle Making

Daniel M. Berry[1] and Luisa Mich[2]

[1] Cheriton School of Computer Science, University of Waterloo
Waterloo, ON, N2L 3G1 Canada
`dberry@uwaterloo.ca`

[2] Department of Industrial Engineering, University of Trento
I-38122 Trento, Italy
`luisa.mich@unitn.it`

**Abstract.** This paper describes several traditional algorithms for making Chinese and Italian noodles and classifies each according to its computational complexity. It examines machines for doing each algorithm. It cites a world speed record for making a large number of noodles using the algorithm with the maximal complexity. It dissects mysteries about the legend that Marco Polo brought the technology of making noodles to Italy from China. It determines that both Chinese and Italian ways of eating food can be applied to both Chinese and Italian noodle dishes. It compares the power of the algorithms. It considers the nature of variations of the traditional algorithms. It concludes by mentioning avenues for further studies.

## 1 Introduction

Each of the Chinese and the Italians make and eat a large variety of dough-based products of various sizes and shapes. This paper uses "noodle" as general term to name a single unit of any product of this type regardless of its national origin and regardless of its size and shape[3]. The Chinese call their noodles "miàn tiáo" (面条) or just "miàn", and the Italians call their noodles "pasta". Therefore, this paper uses "miàn" and "pasta" when talking about Chinese noodles and Italian noodles, respectively. Note that "miàn" and "pasta" are collective nouns that denote collections of noodles. Thus, this paper needs to use "strand", perhaps prefixed by "miàn" or "pasta" as an adjective, when talking about one unit[4] of miàn or pasta.

This paper presents one key algorithm from each of China and Italy to make the country's most traditional kind of noodles from already made dough of the proper composition for what is being made. Later, it presents some other algorithms, again from

---

[3] Admittedly, the term "noodle" connotes a string-like product, e.g., spaghetti. Nevertheless, even though many such products *are* string like, the term is generalized in this paper to include even short products, e.g., maccheroni or macaroni, and even shaped products, e.g., farfalle or bowties.

[4] Just as with "noodle", the term "strand" is used used even when the unit is shaped differently from or is shorter than what is normally called a strand or noodle.

China and Italy, for making other kinds of noodles. Each algorithm is characterized by its computational complexity, as a function of the number, $n$, of noodles produced. There are actually two complexity measures, the *local complexity* and the *global complexity*.

The local complexity is for time required for the algorithm to make one batch of noodles. Generally, the number of noodles that can be made in one batch is limited by a combination of the resources available and the physical properties of the noodle dough. The resource limits that come into play include the amount of flour that can be handled conveniently by the noodle maker, the amount of dough that can be worked on by the noodle-maker's rolling pin, the amount of dough that can be fed at once through a flattening device's rollers, the amount of flattened dough that fits on the noodle-making table, and the amount of flattened dough that can be fed at once through a cutting device. The main physical property of the noodle dough that comes into play is that a noodle with too small a cross section tends to break as it is stretched.

The global complexity is for the time required to make, with successive applications of the algorithm, enough batches to yield all the noodles needed for an occasion. Of course, in a home or in a restaurant that makes noodles to order, usually one batch suffices. In any, case, the global complexity is always linear in the number of noodles produced, on the assumption that any algorithm requires about the same amount of time every time it is used to make the same-sized batch of one kind of noodles. Therefore, for each algorithm, only its local complexity is given.

## 2   Traditional Chinese Miàn Algorithm

It appears that the signature variety of miàn in China is the hand-pulled variety known as lā miàn, which originated in and around Lan Zhou, the largest city in the Gansu Province of Northwest China. Lā miàn is made by starting with a single strand of dough and repeatedly stretching and folding it to produce a large number of thin strands, the diameter of the final strands depending on the diameter of the single initial strand and the number of folds.

1. The lā miàn maker takes a previously prepared tube of very flexible dough of diameter $D$ and of length $L$. ($L$ needs to be no longer than the distance across the lā miàn maker's two outstretched arms, and $D$ needs to be no bigger than what the lā miàn maker can grip with one closed hand.) Call this tube of dough "the *initial bundle*".

   (a) He[5] dusts the bundle with flour.

   (b) He folds the bundle in half and pinches each end,
       - in one case, to merge two ends into one, and
       - in the other case, to make an end out of a fold.

       The bundle is now of length $\frac{L}{2}$.

   (c) By twirling the new bundle like a jump rope, he stretches the new bundle back out to the original length, $L$.

---

[5] We use "he" as a singular pronoun to reference a noodle maker of any gender.

    (d) The result is a new bundle with twice the number of strands as the previous bundle, and the diameter of each strand in the new bundle is $\frac{1}{\sqrt{2}}$ times the diameter of each strand in the previous bundle.

    These steps are repeated until the strands are of the desired diameter.

2. The lā miàn maker trims off the ends to leave strands of length $0.9 \times L$. Then, the lā miàn maker lays out the bundle of strands on the table and, in one swift cut perpendicular to the long axis of the strands, cuts all strands to leave two bundles of strands of length $0.45 \times L$.

For a video showing a Chinese chef making lā miàn, see
https://www.youtube.com/watch?v=PHoQN9vQwHE,
particularly the last minute and a half.

On the assumptions that $D$ is 1 inch, that $L$ is 1 meter[6], and that the final miàn are $\frac{1}{16}$ inch ($\approx 1.59$ mm) in diameter, there are 8 folding and stretching steps, producing 256 trimmed strands each of length 90 centimeters. Then, the final cutting step produces 512 miàn, each of length 45 centimeters.

The local complexity of this traditional Chinese miàn making method is $\log_2 n$ to make $n = 2^m$ miàn in $m - 1$ folding-and-stretching steps and 1 trimming-and-cutting step[7].

## 3   Traditional Italian Pasta Algorithm

An Italian pasta maker rolls out a ball of the proper dough into a rectangular sheet of the desired thickness $T$ and the desired length on one edge, hereinafter called edge $L$ (for "length"). An edge that is perpendicular to $L$ is called edge $W$ (for "width"). ($L$ and $W$ need to be small enough for an $L \times W$ sheet of dough to be easily worked on by a hand-operated rolling pin.) Both sides of the sheet are then thoroughly dusted so that they are not sticky. The sheet is then rolled up very loosely perpendicular to $L$ so that the resulting tube is of length equal to that of $W$. The pasta maker decides the type of pasta that is being made to determine the width $w$ of one strand. Ideally, the width $W$ of the sheet is divisible integrally, $n$ times, by $w$. For fettuccine, $w$ is smaller than for lasagne.

1. The pasta maker uses a knife to cut away a section of the tube of width $w$.
2. The pasta maker unrolls the section into a strand of width $w$ and of length equal to that of $L$.

This cutting and unrolling of sections is performed $n - 1$ times and then the remaining section is unrolled to produce the last strand of a total of $n$ strands. All of this cutting and unrolling must be done quickly to prevent the rolled up tube from sticking to itself.

---

[6] The reason that the diameter is in inches while the length and other dimensions are in meters is that it is easier to describe the effect of halving the diameter in terms of binary fractions of an inch.

[7] We are assuming that trimming and cutting take about the same amount of time as does folding and stretching.

0x4810563306d53762d29d978acc1163798b9d4b35d23490b7d1cffa582b3e9675

So, if one is making 30-centimeter long fettuccine whose cross section is $\frac{1}{4}$ inch by $\frac{1}{16}$ inch. Then, $w$ must be $\frac{1}{4}inch$, $T$ must be $\frac{1}{16}$ inch, $L$ must be 30 centimeters, and $W$ can be anything that is less than the length of the pasta maker's rolling pin and is a multiple of $w$. Let's assume that $W$ is 8 inches. Then from one 8 inch by 30 centimeter sheet, the pasta maker will need $8 \times 4 - 1 = 31$ cuts to make 32 strands. For wider pasta, such as lasagne, fewer cuts are needed.

For a video showing making fresh pasta mostly by hand, see
`https://www.youtube.com/watch?v=-lMmUf2nqYA`.
The pasta maker in this video is using a machine to speed up the making of the sheet, but is then doing the rolling, cutting, and unrolling by hand.

The local complexity of the algorithm is linear in the number of strands, $n$, made from $n - 1$ cuts and $n$ unrollings in one sheet of dough.

There are at least two devices that allow cutting a prepared sheet of dough into a lot of strands in one step:

- a pasta cutter rolling pin whose cutting ribs are spaced $w$ apart and
- a pasta making machine whose cutting blades are spaced $w$ apart.

With either of these devices, there is no need to roll up the sheet and cut away one strand at a time. Instead,

- the cutting rolling pin is rolled once over the flat sheet of prepared dough, leaving the strands flat on the table with no need to unroll, or
- the sheet is fed through the machine, and the strands come out of the machine already unrolled.

Several cutter rolling pins with ribs spaced different distances apart can be seen at
`https://www.casalinghivenditaonline.it/en/kitchen/`
`cut-the-dough/beechwood-spaghetti-cutter.html`.
For a video showing making fresh pasta with a machine, see
`https://www.youtube.com/watch?v=IOsnlFcO748`.
The algorithm embodied by each of these devices can be described as a parallel, vector processing algorithm. Thus, the local complexity of the algorithm to make $n$ strands from one prepared sheet with either of these devices is constant. That is, all $n$ strands are made at the same time, in the time required to roll the cutting pin over the sheet or to feed the sheet though the machine.

## 4   World Record Setting Chinese Lā Miàn Maker

In "How to make noodles", found at
`http://www.scientificpsychic.com/mind/noodles.html`,
one paragraph describes the video found at
`https://www.youtube.com/watch?feature=player_embedded&`
`v=auhHl5-6VdY`:

> This video shows chef Kin Jing Mark making Chinese hand-pulled noodles. He held the Guinness World Record as the fastest human noodle maker

for several consecutive years. His last record was set in 1993 on NBC's afternoon talk show, Vicki, when he stretched out $4,096$ strings of Chinese noodles by hand in $41.34$ seconds. The fine noodles are called dragon beard noodles (longxu mian).

The number, 4096, of strands that Kin Jing Mark made, is telling; $4096 = 2^{12}$. It is clear that no one sat there counting the individual strands to arrive at 4096. It is equally clear that the number of folds was counted and that number was used as the exponent of 2 to calculate the number of strands. Thus on average, Kin Jing Mark did one fold and stretch every $3.445$ seconds. Wow! Clearly, the cook and the people who made the video understand the exponential growth of the number of strands in the algorithm.

## 5    Automation of Algorithms

There are machines that automate the Italian pasta-making process. For example, the Italian company Italgi makes some industrial strength pasta sheeters and cutters that can be seen at
`http://www.italgi.it/e-pasta-sheeters.htm`.
Also Arcobaleno makes pasta sheeters and cutters that can be seen at
`http://arcobalenollc.com/pastaequipment.html`.
These machines simulate the human pasta maker's behavior, to make so-called perfectly formed pasta every time.

There does not appear to be any machines that automate the making of Chinese lā miàn. There are machines that automate the mixing and kneading of the dough, but there do not appear to be any machines that automate the folding and stretching[8]. Perhaps the main reason that lā miàn are called in English "hand-pulled" is that they *must* be made by a human's hand.

## 6    Other Methods of Making Noodles

China does have other methods of making miàn [1, 2]:

  – Cut (qiē): A sheet of dough is cut into strands of the desired width, as in the traditional Italian pasta algorithm of Section 3. The local complexity of this process is linear in the number of strands produced.
  – Squared (piàn); As one is making cut miàn, directly above an open pot of boiling water, each (long) strand is torn by hand into square-sized pieces (short strands). The local complexity of this process is linear in the number, $n$, of pieces produced: If $s$ long strands are produced with $s - 1$ cuts, and from each long strand are produced $p$ short strands with $p - 1$ tears, then $s \times p = n$. The total number of steps is $(s - 1) + (s - 1) \times (p - 1) = (s - 1) \times p$, which is approximately $s \times p = n$.

---

[8] The closest we were able to find were the machines by Yamato Noodle that make ramen, udon, and soba, all Japanese noodles. However, as shown at
`http://www.yamatonoodle.com/noodle_machine/`,
after preparing the dough, these machines cut the noodles off.

– Extruded (jǐyā): Dough is pushed through a die with holes of the desired shape to form strands, one per hole in the die. The local complexity of this process is constant, since all the strands are produced at the same time.
– Kneaded (róu): A small ball of dough is worked on a flat surface to form it into a strand of the desired shape. The local complexity of this process is linear in the number of strands produced.

Also Italy has other methods of making pasta [3, 4]:

– Short cut: As one is following the traditional Italian pasta algorithm of Section 3, the $s$ unrolled (long) strands are laid out in parallel, side-by-side into a striped sheet. Then, $p - 1$ equally spaced cuts perpendicular to the axis of the length of the strands are applied across the whole sheet, to produce $s \times p = n$ rectangular pieces (short strands). The local complexity of this process is in the order of the square root of the number of pieces produced: The complexity analysis for this process starts as for the production of $n = s \times p$ squared piàn. The difference is that all $s$ long strands are cut together in only $p - 1$ cuts. Thus, the total number of cuts is $s - 1 + p - 1$, which is approximately $s + p$. If the sheet is close to being a square, then $s \approx p$, and $s + p \approx 2\sqrt{n}$, since $s \times p = n$. In a complexity estimate, a constant multiplier of $\sqrt{n}$ can be ignored, because the main contributor to the growth of $2\sqrt{n}$ is $\sqrt{n}$, and not the constant multiplier.

There is a variation of the pasta cutter rolling pin, mentioned in Section 3, that has cutting ribs running along the long axis of the pin, perpendicular to the strand-cutting ribs. This variation is for producing a whole sheet's worth of short-cut pasta in one roll of the pin over a prepared sheet of dough. The local complexity of this method of making short-cut pasta is constant.
– Extrusion: Dough is pushed through a die with holes of the desired shape to form strands, one per hole in the die. The local complexity of this process is constant, since all the strands are produced at the same time.
– Short-cut extrusion: Each extruded pasta long strand is cut perpendicular to the length of the strand into short pieces. The local complexity of this process is in the order of the square root of the number of pieces produced, by the same analysis as for the above short-cut pasta.

Additional shaping may be applied to the pasta produced by any of the described methods.

There are machines that automate all the various ways of making Italian pasta, as shown at
`http://arcobalenollc.com/pastaequipment.html.`

## 7  Mystery

While China does have cut miàn, resembling Italy's traditional cut pasta, it appears from a thorough search of the Web, that Italy does not have anything resembling lā miàn that is made with a repeated-folding-and-stretching method.

Legend has it that Marco Polo brought noodles to Italy when he returned to his native Italy from his lengthy visit to China [1, 5] although there is some doubt. If this legend is true, then why do the Italians not make their pasta the same way the Chinese do? One possible and plausible explanation is that Polo brought back necessarily well-dried samples of actual miàn rather than the algorithm. Polo told the Italians to soften these dried miàn by cooking them a few minutes in boiling water. After the Italians decided that they liked the results, they proceeded to figure out a way to produce the product that Polo had brought back and came up with the algorithm described in Section 3 and never even thought of the original algorithm described in Section 2. This is not the first time an attempt was made to reproduce a product by reverse engineering from instances of the final product rather using the original algorithm [6]. Such reverse engineering does not always succeed in duplicating the original product, and occasionally ends up inventing a new product that is different from the original in subtle or not so subtle ways. Pasta and Chinese miàn *are* different to the discerning palate, and each is good in its own right, even to connoisseurs of the other.

On the other hand, maybe the legend is false, and each country invented its own kinds of noodles and stumbled on to its own methods with no knowledge of the other's kinds and methods.

## 8   Different Methods of Eating

Chinese food, including noodles, is designed to be eaten with only a pair of chopsticks. The only people in Chinese cooking that are in need of any utensils, such as a knife, are the cooks that prepare the food to be eaten with only chopsticks. Part of the job of a cook is to cut up any meats or vegetables cooked in the food to bite-sized pieces that can be picked up with only a pair of chopsticks. The ordinary eater has no need for a knife and for anything to hold the food while it is being cut. Italians do use knives, forks, and spoons, and have plenty of dishes, generally so-called second plates (secondi piatti) that require that the eater cut his or her own meat or vegetables into bite-sized pieces. Interestingly, most pasta dishes, other than those involving pasta whose individual noodles, e.g, lasagna, are too big to be bite sized, could be eaten with chopsticks. Most pasta sauces are made with bite-sized chunks of meats, fish, and vegetables, that have been cut to bite size by the cook. Here too, the pasta eater with chopsticks has no real need for a knife and fork. Of course, if the sauce were so liquidy that it could not stick to the pasta, vegetables, or meat, then a spoon would be needed, but that would be true also of any Chinese food that had a very liquidy sauce. Perhaps, this similarity between Chinese food and Italian pasta dishes is confirmation of the veracity of the Marco Polo legend.

## 9   Comparison of Algorithms

The Chinese repeated-folding-and-stretching algorithm, with its logarithmic complexity is significantly more powerful than any Italian cutting-based algorithm, with linear complexity, in two different ways:

0xa7542fa80f1d569dbd3861c81b0d5484057b312e7514f9305cb393274d153eaf

1. The logarithmic-complexity algorithm can generate so many more noodles in a time duration than can any linear-complexity algorithm, particularly when the noodle maker is folding and stretching quickly, and he goes beyond six folds. Twelve folds and stetches in $41.34$ seconds suffices to make 4096 noodles. Making 4096 noodles by any linear-complexity algorithm would require a *lot* more than $41.34$ seconds.
2. When the two algorithms are operated totally manually, it is a lot easier to achieve uniformity in the cross section of the noodles with the repeated-folding-and-stretching algorithm than with any cutting-based algorithm.

## 10   Variations of the Basic Noodles

The different algorithms for making noodles lead to variations of the basic noodles that we see in the two countries. The fact that a flat sheet of dough is cut into strands that become the noodles suggests cutting the sheet into other shapes. Hence, we see noodles in the shapes of triangles, squares, rectangles, circles, stars, etc. Once we have these different shapes, we begin to see yet other variations, such as pinching a rectangle into a bowtie, molding a circle into a shell. Once we have shaping and pinching, with the addition of a bit more water, pinching can be used to paste edges together. Then rolling and pasting a wet rectangle yields a tube that can be filled. Covering part of a shape with some filling and folding and pinching wet edges around the filling yields filled tortellini. Once all this is automated, shapes that can be made by machinery become possible.

The Chinese lā miàn algorithm does not lend itself to these cutting-based variations. There are variations in the length and diameter of the noodles, the raw material used to make the noodles, and the twistiness of the noodle achieved by variations in the process of drying the noodles, e.g., by spiralling the wet noodles around a dowel of an appropriate diameter.

## 11   Conclusion

There are a number of issues that require further study, which we encourage the interested reader to take on.

- Perhaps, the space required for an algorithm should be considered. Is there a meaningful space–time tradeoff? Does the size of the available kitchen make a difference, e.g., as for a home *versus* a restaurant kitchen?
- What is the interaction between the algorithms and the ingredients used to make the dough?
- What is the interaction between the algorithms and the issue of fresh versus dried noodles?
- What is the interaction between the algorithms and the local culture?
- Can a Chinese algorithm be applied in Italy and can an Italian algorithm be applied in China?
- How easily learned is each algorithm?
- How automatable is each algorithm?

0x10cdece301da6b143f108588051b9b774c1ab03965c4cccff4610b346e2a2b2a

– Which algorithm is most appropriate to use in a restaurant in which food is made to order?
– What are the empirically determined threshold values for $n$ (the number of noodles made in a batch), below which the traditional basic and parallel Italian algorithms are more efficient locally than the traditional Chinese algorithm?

A short investigation on the World-Wide Web shows that noodle production is a big business world wide. It is apparent that noodles of all kinds are made, not just in the countries of their origins, but just about anywhere. It does appear that for noodles of nationality $N$ to be made elsewhere, in $E$, $E$ must have a significant population of immigrants from $N$.

Of course, in the modern small world in which a food product produced in any part of the world can be sold in stores anywhere in the world, one can buy Italian pasta in China and Chinese miàn in Italy.

## Acknowledgments

## References

1. Lin-Liu, J.: On the Noodle Road: From Beijing to Rome, with Love and Pasta. Riverhead Books, The Penguin Group, New York, NY, USA (2013)
2. Wikipedia: Chinese noodles— Wikipedia, the free encyclopedia (2015) `http://en.wikipedia.org/wiki/Chinese_noodles`, [accessed 1 May 2015].
3. Hildebrand, C., Kenedy, J.: The Geometry of Pasta. Box Tree, London, UK (2010)
4. Wikipedia: List of pasta— Wikipedia, the free encyclopedia (2015) `http://en.wikipedia.org/wiki/List_of_pasta`, [accessed 1 May 2015].
5. Kummer, C.: Pasta. The Atlantic Monthly (1986) `http://www.theatlantic.com/magazine/archive/1986/07/pasta/306226/`, [accessed 1 May 2015].
6. Berry, D.M.: 'What, Not How?': The case of specifications of the New York bagel. Annals of Improbable Research **15** (2009) 6–10

0x22013a74293f6f55e0c065d5fa4c4997aab12d71c9658a574e9a1a492e083b74

# SIGBOVIK 2016 Paper Review

## Paper 2: The Computational Complexity of Chinese and Italian Noodle Making

**With the assistance of the Carnegie Mellon University Postmodern Languages Department**

中国面条算法最好。 我很高你同意。 大利面算法太慢了,因大利人真惰。 我一个大利人用大利文写介,但他 没回信。 所以谷歌翻他写介。 面很好吃.在我家我有一只。 他睡的候 我切断 他的毛做面。 刀削面呢? 你吃? 听那是最好的面。 豆腐呢? 你喜吃豆腐呢? 我想吃豆腐。 你妹妹多大? 第八部分不。 中国人用勺喝。 你怎以中国人只有筷子? 加拿大人怎喝? 除了那个部分以外都了,一共有意思。

- 笨蛋美国人

Come osi dire rendendo noodle italiano è più lento di noodlemaking cinese ? Quei comunisti non poteva dire polpette di mia madre dal mio nutsack . Avrei potuto iniziare ad usare le bacchette quando mangio, perché il mio coltello e forchetta saranno preoccupati scriccatura gli occhi fuori e tagliare in su per il mio cane .

Mio fratello Giovanni conosce alcuni ragazzi in Sicilia . Non si vuole scopare con questi ragazzi . ti suggerisco" Respingere " questo documento a causa di " riferimenti poveri ", se vi capitasse di voler vivere per vedere un'altra citazione. - Google Traduttore

0xabf716b6f89315319f2c8e33070250c4e121084de16180a39cb7f8f51d55b674

# Which ITG Stepcharts are Turniest?

Ben Blum

bblum@cs.cmu.edu

## Abstract

ITG is a popular dance game in which players step on arrows while listening to music. The arrow patterns, indicated by a *stepchart*, may range among any level of complexity and difficulty. Among the many factors contributing to a stepchart's difficulty is how much the player must turn from side to side. Other more obvious factors, such as raw speed, have been well studied in prior work. This paper presents an analytic study of this *turniness* factor. We study the turniness of many existing stepcharts, and present a novel (but unsurprising) approach to automatically generating maximally (or minimally) turny charts. Among real-world songs, we find stepcharts with overall turniness ranging from 0% to 81.33% of the theoretical maximum.

***Categories and Subject Descriptors***   D.D.R. [*Exercise and Fitness*]: Arcade Dance Games

***Keywords***   in, the, groove

## 1. Introduction

In 2005, Roxor Games, Inc. released *In The Groove*, a dance rhythm music video arcade fitness game, in which players control a protagonist using their feet to step on floor-mounted directional indicators. The protagonist, shown in Figure 1, takes the form of any number of arrow-shaped directional receptacles, and must navigate a world of similarly-shaped obstacles (henceforth "arrows") by consuming them with the appropriate receptacle. Roxor Games, Inc. In The Groove (henceforth "ITG") is most commonly played using the "cabinet" form factor, shown in Figure 2, which includes two large metal dance pads, each with four directional indicators (henceforth, also, "arrows").

The game includes a library of rhythmic audio accompaniment files (henceforth, "songs"), each of which



**Figure 1.** ITG gameplay, including score indicator (top), protagonist avatar (mid), directional obstacles (low), and step judgement, life bar, and combo indicator (figure these out for yourself, I'm getting tired).



**Figure 2.** An ITG cab. RIP in peace, Roxor (Konami 2005).

is associated with one or more fixed patterns of arrows (henceforth, "stepcharts"). These charts are often, but not always, synchronized to the beat of the song. During gameplay, the stepcharts appear on screen and scroll towards the protagonist avatar at a rate either fixed or variable (henceforth, "BPM"). When the position of an arrow in the chart coincides with the avatar, the player must actuate the arrow of the corresponding direction. The game will judge the player's timing accuracy, and penalize or reward them accordingly with scores and life bar fill. A "Fantastic" judgement (as in Figure 1) indicates a timing error not exceeding 15 milliseconds. Other judgements include Excellent, Great, Decent, Way Off, and Miss. As a visual assist to the player, notes are coloured according to their beat granularity: ♩, ♩3, ♪, ♪3, ♫, ♫3, ♩.

0x0d417c7e9a4c6c1e996ae17677cda151f43658f95fd2134e9ecc033a6d39113d

**Figure 3.** ITG can be played "Bar" or "No-Bar". Bar players must wear sandals (like an idiot; who is that guy anyway?), while No-Barrers must always play on the right.



**Figure 4.** Doubles play is beyond the scope of our work. It requires (or perhaps produces?) a magnificent beard.

The game may be played in several modes, the most common supporting up to two (2) players, each operating their own protagonist using either the left or right set of four arrows. This game mode may be played with or without the assist of a curved metal rod mounted behind the arrows (henceforth, "bar"), as shown in Figure 3. In other game modes, a single player may operate up to all 8 of the arrows. When all 8 are used, the game mode is known as "Doubles", as shown in Figure 4, and is often associated with excessive No-Barring, use of hands and knees to operate the arrows, and impressive facial hair. In this body of work, we will focus exclusively on the Singles game mode, in which each player controls four arrows: one Up ($U$), one Down ($D$), one Left ($L$), and one Right ($R$). Without loss of generality, we further assume that only one player plays at a time, using exactly two feet at a time, and that she will shower immediately afterward.
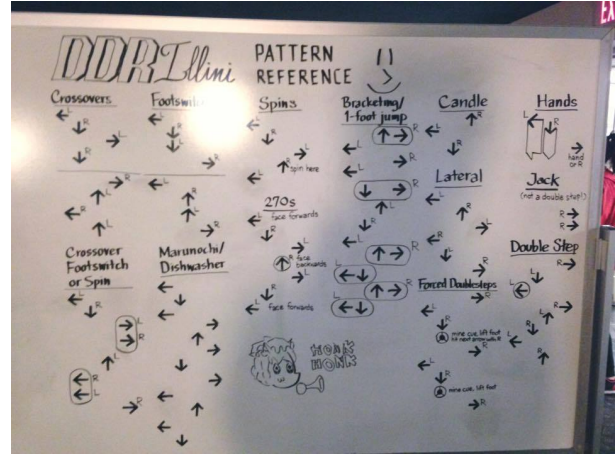


**Figure 5.** Common patterns (Akgul 2015). Honk honk!

## 2. Properties of Stepcharts

Being a game, one might assume that players find it fun to step the arrows in the indicated patterns, as opposed to, for example, stepping them in random unrelated patterns, or stepping off the pad entirely to go play Rocket League (Psyonix 2015). Accordingly, step authors take care to produce stepcharts with interesting patterns. Typically, players step the arrows alternatingly with either foot. Step authors can either reinforce this tendency, by maintaining that left and right arrow targets always receive arrows of different (even or odd) sequential parity; or subvert it, forcing the player to step on the left arrow with their right foot (or v.v.). The former pattern is called a *stream*, and the latter pattern a *crossover*. Crossovers are loved by some players (Figure 3, left), and hated by others (Figure 3, right).

Figure 5 shows many common step patterns. Among these, note especially the *lateral* ($LU\,R\,L\,D\,R$), in which the player briefly faces backwards (left foot on $R$ and right foot on $L$), and the *spin* ($LU\,R\,D\,LU$), in which, yeah, you get the picture.

### 2.1 Difficulty

Stepcharts are rated in difficulty on a scale from 1 to 20 "feet" (hurtpiggypig 2013)[1]. To date, difficulty is judged by a human (typically the step author), and primarily reflects factors such as burstiness (*footspeed*), prolongedness of streams (*stamina*), and complexity of jumps, holds, and crossovers (*technical*). These measures of difficulty have been well-studied in prior works (Zetorux and WinDEU 2008; Konkul 2014; thattagen 2014). Some players are really good at In The Groove.

---

[1] Other dance games, which shall remain unnamed, but whose initials are (Konami 2005), feature wimpier stepcharts rated only up to 10. As ITG players got better, they left those charts in the dust and began creating their own charts to challenge themselves with ridiculous difficulties. This paper's author plays ITG at the 13-14 level.

## 2.2 Facing

Regardless of the presence of crossovers, all step patterns will keep the player *facing* one way or another, and most include sequences that will change the player's facing back and forth. For example, when the player stands with left foot on $L$ and right foot on $U$, they are facing up-left, or $UL$. Standing on $D$ and $R$ also produces the same $UL$ facing, unless the player is crossed-over (left foot on $R$), in which case the facing is $DL$. Table 1 shows all possibilities of facing using two feet on four arrows. We exclude the possibility for both feet to be instantaneously on the same arrow (called a "footswitch"), leaving this pattern to future work.

| | | Right foot | | | |
|---|---|---|---|---|---|
| | | ← | ↓ | ↑ | → |
| | ← | - | $UR$ | $UL$ | $U$ |
| Left foot | ↓ | $DL^*$ | - | $L$ | $UL$ |
| | ↑ | $DR^*$ | $R$ | - | $UR$ |
| | → | $D$† | $DR^*$ | $DL^*$ | - |

**Table 1.** Facing directions. "Crossover" facings are marked (*), and the "lateral" facing is marked (†). Note the appealing diagonal symmetry.

## 2.3 Turning

A stepchart contains a *turn* when a sequence of steps changes the player's facing.

**Definition 1** (Turniness). *The turniness $\mathcal{T}$ of a single step is the angular distance (measured in increments of $\pi/4$) which that step changes the facing compared to the facing from the previous two steps.*

*The turniness $\mathcal{T}$ of a stream is the average of each step's turniness (excluding the first two for which it is undefined).*

Because only one foot may change at a time, it is easy to see that the maximum turniness of a single step is 2. These steps always involve one foot moving entirely across the middle of the pad (from $U$ to $D$, from $L$ to $R$, etc). By tradition, these are called *candle steps*, according to the imagery that if a candle were placed in the middle of the 4 arrows, this step would knock it over. I don't know why it couldn't be called a beer step instead, though.

**Turning is exhausting.** Consider the three stream patterns shown in Figure 6. In the leftmost stream, the player must barely move her feet at all, never changing facing from $U$. In the middle stream, the player faces largely $UR$, with brief $U$ and $R$ facings ($\mathcal{T} = 1$) as she moves between the $L/D$ and $U/R$ arrows. Both of these streams are very easy to step without becoming fatigued. However, the rightmost stream features many candle steps (or beer steps), changing facing from $UR$ to $UL$ and back in the span of a single measure. ITG players hate this (Fouhey and Maturana 2013)! However, note that even
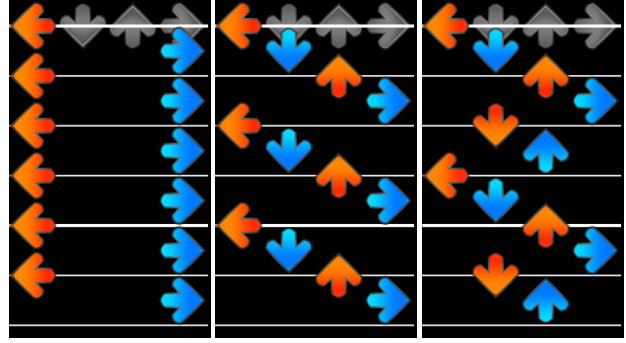


**Figure 6.** Three streams of differing turniness. Note how the color-coding of the notes by beat aids to discern turniness at a glance: the right foot is always on blue, and it moves among 1, 2, or 3 different arrows respectively.

in this stream, not *all* of the steps have $\mathcal{T} = 2$: the *average* turniness of this stream is only 4/3.

We are interested in the following questions:

1. What is the turniest possible stepchart?

2. How is maximum turniness affected by various constraints to contain no crossovers/laterals/etc?

3. How turny are real-world ITG charts?

## 3. Chart Synthesis

We developed a program for exploring all possible step patterns using a little-known yet powerful computational technique (Apostol 2012; Curtin 2005). This program computes the average turniness of every such pattern. However, while some players enjoy any step pattern including spins (Vangpat 2015), other players may prefer only crossovers and laterals in their charts, while still others prefer charts completely vanilla. Hence, our program is further capable of filtering patterns by 5 predicates:

- All patterns allowed
- No spins
- No 270s
- No laterals
- No crossovers (vanilla)

Note that each predicate captures a strict subset of step patterns compared to the one above it (a lateral is a crossover, and so forth). These predicates are implemented as shown in Figure 7.

Deciding on how long of step sequences we should search for is a tradeoff between accuracy, in judging turniness in small fractions, and (drumroll...) exponential explosion. To strike a balance we decided to experimentally measure turniness at a granularity of 1/8th, which requires searching for step sequences of length 18 (twice the denominator, plus 2 for the first 2 steps whose turniness

$$\begin{aligned}
\text{is\_xover}(\phi) &= \phi \equiv DL \lor \phi \equiv DR \\
\text{is\_lat}(\phi) &= \phi \equiv D \\
\text{is\_270}(\phi, \phi_1, \phi_2) &= (\phi \equiv DR \land (\phi_1 \equiv DL \lor (\phi_1 \equiv D \land \phi_2 \equiv DL))) \lor \\
&\quad (\phi \equiv DL \land (\phi_1 \equiv DR \lor (\phi_1 \equiv D \land \phi_2 \equiv DR))) \\
\text{is\_spin}(\phi, \phi_1, \phi_2, \phi_3) &= (\phi \equiv R \lor \phi \equiv UR \lor \phi \equiv L \lor \phi \equiv UL) \land \text{is\_270}(\phi_1, \phi_2, \phi_3)
\end{aligned}$$

**Figure 7.** Formal definitions of crossovers, laterals, 270s, and spins. Each predicate takes as arguments the current facing $\phi$, and up to the three most recent previous facings, $\phi_1, \phi_2, \phi_3$.
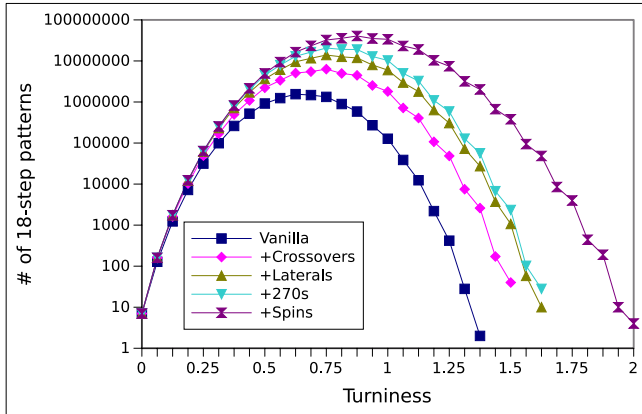


**Figure 8.** Distribution of unique step sequences among turninesses, classified by what types of step patterns are allowed in the chart.

is undefined). Also, our program WLOG restricts streams to always start on the left foot, and never to start crossed over. Figure 8 shows our results. The highest data point clocks in at 40,609,780: there are over 40 million unique 18-step sequences, any patterns allowed, that have turniness 0.875.

We see that, with one exception, allowing each new type of step pattern increases the maximum possible turniness of the chart. The exception is that 270s do not allow for any additional turniness over laterals, although they do allow for *more possible ways* to achieve maximum turniness. Despite no additional turniness, this is crucial information for stepchart authors, as ITG players will (usually) get bored of any chart that simply repeats the same pattern over and over.

**Maximum turniness!** Now to the paper's titular question: what step sequences underlie those data points on the far right of each curve? Though it would be tedious to transcribe them all into Stepmania's step editor (Step-Mania 5.0 2016), we manually inspected many of them to select the most representative and/or photogenic ones from each category. We show these sequences in Figure 9. These visualizations allow us to manually compute the true turniness of each sequence, imagining the core se-

quence repeated ad infinitum, freeing ourselves from the 1/8th granularity imposed by our experimental setup. We find that the turniest crossovers have $\mathscr{T} = 3/2$ and that the turniest laterals have $\mathscr{T} = 8/5$.

The case of 270s is irregular: we found no 18-step sequence turnier than the turniest laterals, and most notably, none of the sequences included a 270 step in the *middle* of the stream. Figure 9(d) shows a stream with crossovers and laterals and no 270 step until the very last one. To see why, imagine yourself on the pad here: the only possible 19th step that doesn't result in a spin is to step back on the same arrow your left foot is already on. That step has $\mathscr{T} = 0$, which would undo any potential turniness gained from going into the 270. Nevertheless, 270s retain some real-world application, as demonstrated by (Foy and ☺ 2004) which similarly employs a 270 as the very last step.

Perhaps predictably, pure spins make for the turniest possible chart. There are only two possible ways to candle every step; one's name is "clockwise", and you can guess the other.

**Minimum turniness.** Each of the 5 categories shares the same data point at $\mathscr{T} = 0$: there are 7 possible streams that never turn at all. These are the 7 "tower" patterns $LRLR$, $LULU$, $LDLD$, $URUR$, $DRDR$, $UDUD$, and $DUDU$. We already showed the first of these in Figure 6, left. For further treatment of minimally turny charts, we refer the reader to (WinDEU and Ashura 2009).

## 4. Existing Chart Analysis

Of course, composing a stepchart of nothing but the theoretically-turniest patterns would be extremely tiresome (in more ways than one). We turn our attention to existing ITG charts which were developed prior to this study, to investigate how much ITG players turn in the real world. We processed every stepchart in our collection, which includes 70+ song packs. Most well-known pack series from the ITG community are included, such as *Cirque, ITG, Mute Sims, Pendulum, r21*, SPEED-COOOORE, Tachyon, The Legend of Zim, Valex's Magical 4-Arrow Adventure*, and *Oh Henry! Mad Stamina: Streamy · Brutal · Stamina Bar in Milk Chocolate*.
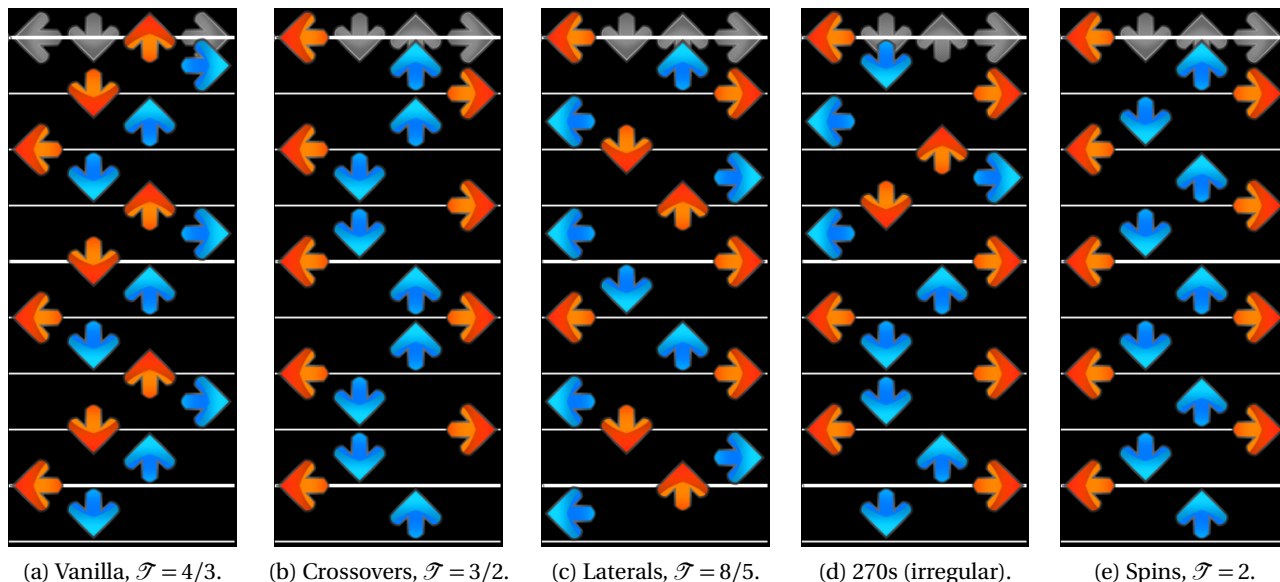
(a) Vanilla, $\mathscr{T} = 4/3$.    (b) Crossovers, $\mathscr{T} = 3/2$.    (c) Laterals, $\mathscr{T} = 8/5$.    (d) 270s (irregular).    (e) Spins, $\mathscr{T} = 2$.

**Figure 9.** The turniest possible stepcharts.

**Limitations.** Real-world stepcharts include many patterns not accounted for in our analysis. We did not expect much trouble from jumps and jacks, opting simply to ignore them and count the turns among the surrounding single-steps. Footswitches and doublesteps, however, can foil this strategy, as our analysis will inadvertently begin stepping with feet inverted from where they should be. Merely getting turned around is not a problem (computers don't have eyes), but the act of turning around involves some candle-steps, which could cause a footswitch-heavy chart to seem very spinny even though the proper footing would never make the player cross over at all. Coping with these limitations is beyond the scope of our work, although we will highlight some experimental false-positives later.

### 4.1 Results

As shown in Figure 10, the vast majority of real-world stepcharts have average turniness between 0.6 and 1.0. We don't really care about those.

**The Turniest Charts.** No joker has yet dared to make the turniest possible chart (all spins), but one chart comes close. Standing alone as the only chart turnier than 1.5 is the Easy 3 for *DO ME* (Konami DDR 4thMIX PLUS), which is mostly spins punctuated by a few breaks, as shown in Figure 11(a). (Vangpat 2015) loves this chart.

Actually, 20 of the 30 turniest charts are all from DDR packs, most of which are Easies. Although DDR charts are known for being fairly crossovery in the upper ranges of difficulty, a close inspection of the easier charts will show that very little care was put in while writing them to avoid excessive double-stepping. It's no wonder new ITG
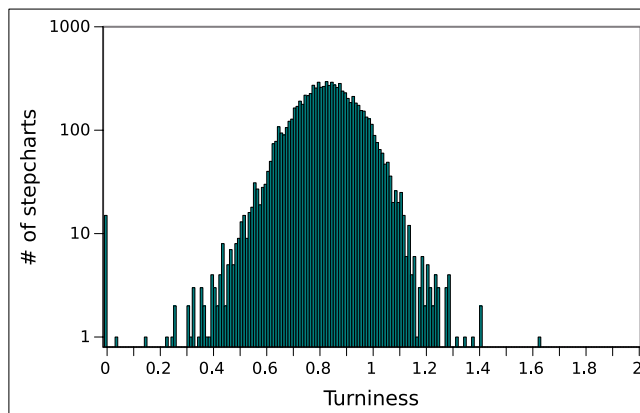


**Figure 10.** Distribution of turniness among 8,278 existing ITG charts.

players, who used to play DDR, always double-step even when they don't need to. Sheesh, Konami.

Beyond these extreme examples, we are interested in some more normal charts people might actually like to play. Table 2 shows the turniest charts of each level, for the most popular difficulty levels ranging from 7 to 17. So, next time you want to work on your laterals or candles, you have me to thank!

Surprisingly, *BRILLIANT 2U (Orchestra-Groove)* (Konami DDR 2ndMIX) is full of 270s. I didn't think anybody ever put those in real charts. Finally, an honourable mention goes to *Oedo Hop* (sssmsm and Palpable 2007), the 2nd turniest 7 (and 4th turniest chart overall!) with $\mathscr{T} = 1.384$. (Hanneman 2010) really likes that one so we had to mention it.
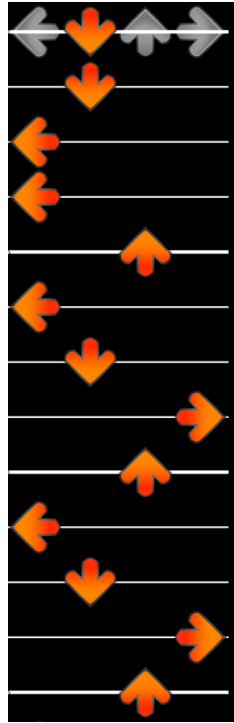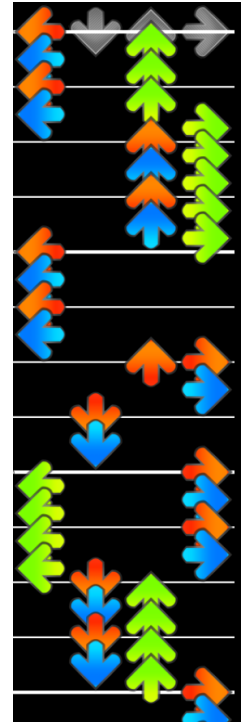
(a) *DO ME*, $\mathcal{T} = 1.63$. Proportion of non-spinny section enlarged to show detail.
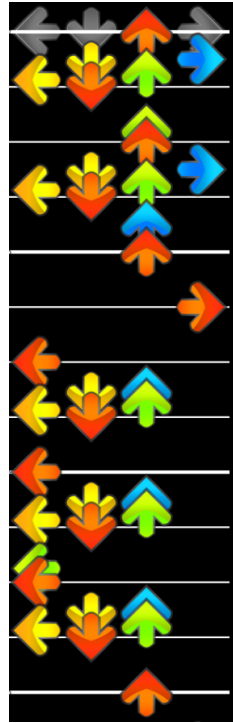
(b) *Mr. Saxobeat*, $\mathcal{T} = 1.04$. Look at all those candles! No wonder it was so hard for us to pass this one.
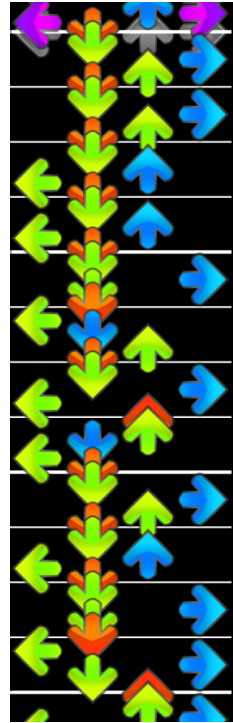
(c) *Flames of the Sky*, $\mathcal{T} = 1.04$. False-positive $\mathcal{T}$ from footswitches. Actually faces $R/U\,R/U/U\,L/L$ the whole time.

(d) *Sick Dubstep Track*, $\mathcal{T} = 0.26$. The least turny nontrivial non-beginner chart.

**Figure 11.** Real-world ITG charts of varying degrees of turniness.

| Ft. | Name | Pack | $\mathcal{T}$ |
|---|---|---|---|
| 7 | TRIP MACHINE | DDR 1st | 1.407 |
| 8 | BRILLIANT 2U (O-G) | DDR 2ndMIX | 1.347 |
| 9 | Hearts of Ice | In The Groove 3 | 1.127 |
| 10 | Visible Noise | In The Groove 2 | 1.188 |
| 11 | KKKKK Come On | r2112 | 1.145 |
| 12 | Winnipeg is Fucking Over | best of r21freak ii | 1.150 |
| 13 | Mr. Saxobeat (*) | Sexuality Violation 2 | 1.044 |
| 14 | Fancy Footwork (†) | Cirque du Zeppelin | 1.026 |
| 15 | Flames of the Sky (*,†) | Ft. Rapids VII | 1.037 |
| 16 | *(no 16 with $\mathcal{T} \geq 1$)* | - | - |
| 17 | Superluminal | Ft. Rapids VII | 1.058 |
| 18 | *(no 18s+ with $\mathcal{T} \geq 1$)* | - | - |

**Table 2.** The turniest existing stepcharts of each difficulty. Songs marked (*) are also shown in Figure 11. Songs marked (†) are false-positives resulting from too many footswitches incorrectly analyzed as spins.

**The Least Turny Charts.** 16 charts in our collection have $\mathcal{T} = 0$. Most of these are beginner charts, rated 2 or 3 feet and with no more than 100 steps. ΔMAX (WinDEU and Ashura 2009) stands out, as a 15-footer with 945 steps. God is that chart ever annoying.

| Ft. | Name | Pack | $\mathcal{T}$ |
|---|---|---|---|
| 7 | King Kong | In The Groove 3 | 0.354 |
| 8 | Jam Jam Reggae | DDR 2ndMIX | 0.428 |
| 9 | Fantasia | In The Groove 3 | 0.382 |
| 10 | Oosanbashi | TLOES Chapter 1 | 0.534 |
| 11 | Release the Music | Ft. Rapids VII | 0.507 |
| 12 | Payon (Theme Of) | CuoReNeRo MeGaPacK | 0.396 |
| 13 | Technodildo | The Legend of Zim 5 | 0.436 |
| 14 | Gentleman | TLOES Chapter 1 | 0.556 |
| 15 | ΔMAX | Tachyon Alpha | 0.000 |
| 16 | Dash Hopes 2 | Tachyon Alpha | 0.327 |
| 17 | Resist | SPEEDCOOOORE | 0.592 |
| 18 | Baby | Sexuality Violation 2 | 0.524 |
| 19 | Pedal to the Metal | SPEEDCOOOORE | 0.468 |

**Table 3.** The least turny existing stepcharts of each difficulty (trivial charts excluded).

Ignoring all beginner-difficulty and turn-free charts, the least turny stepchart is *Sick Dubstep Track* (mute and Gotobed 2011), a 4-measure joke rated 6,482 feet, the majority of which is shown in Figure 11(d). It has $\mathcal{T} = 0.263$.

For readers who prefer more to tower than to turn, we similarly tabulated the least turny charts of each common difficulty in Table 3.

0xb2ad59ed236ef957c365ad85f09f7c8281ade5032a068c791ba0a3196da91c5d

| Song Pack | # of Charts | Avg. $\mathcal{T}$ |
|---|---|---|
| Valex's Magical 4-Arrow Adventure 5 | 150 | 0.964 |
| Valex's Magical 4-Arrow Adventure 4 | 170 | 0.956 |
| Valex's Magical 4-Arrow Adventure 3 | 155 | 0.950 |
| Valex's Magical 4-Arrow Adventure 7 | 150 | 0.937 |
| Valex's Magical 4-Arrow Adventure 2 | 130 | 0.929 |
| Valex's Magical 4-Arrow Adventure | 95 | 0.919 |
| Valex's Magical 4-Arrow Adventure 6 | 145 | 0.917 |
| Subluminal | 16 | 0.906 |
| The Legend of Zim 1 | 26 | 0.906 |
| FoxyMix 5 | 30 | 0.901 |
| FoxyMix 4 - Nuclear Overdrive | 68 | 0.899 |
| Mute Sims 9 | 48 | 0.886 |
| In The Groove Rebirth + | 147 | 0.881 |
| StepMania 5 | 5 | 0.869 |
| Cirque du Zeppelin | 109 | 0.867 |
| In The Groove 3 | 448 | 0.858 |
| … | | |
| Piece of Cake 5 | 20 | 0.785 |
| Cirque du Veyron | 30 | 0.777 |
| Stephcharts and Richarts | 81 | 0.770 |
| Tachyon Delta | 32 | 0.769 |
| Tachyon Alpha | 151 | 0.769 |
| Sexuality Violation 2 | 219 | 0.766 |
| TLOES Chapter 1 | 120 | 0.765 |
| Fort Rapids V (12s-16s) r3 Final | 98 | 0.759 |
| Noisiastreamz | 20 | 0.758 |
| In The Groove | 331 | 0.754 |
| Pendulum | 150 | 0.747 |
| SPEEDCOOOORE 2 | 70 | 0.731 |
| SPEEDCOOOORE | 47 | 0.709 |
| The Legend of Zim 3 | 40 | 0.702 |
| The Legend of Zim 4 | 44 | 0.701 |
| The Legend of Zim 5 | 19 | 0.667 |

**Table 4.** The turniest and least turny stepfile packs. Wow, great job Valex! Zim had a good thing going but has been slacking off recently.

| Step Artist | # of Charts | Avg. $\mathcal{T}$ |
|---|---|---|
| M. Emirzian | 38 | 0.992 |
| sssmsm | 50 | 0.945 |
| J.Berkowitz (Valex) | 1025 | 0.939 |
| B. Dinh | 10 | 0.919 |
| J. DeGarmo | 26 | 0.906 |
| J. Frederick | 176 | 0.881 |
| Fraxtil | 23 | 0.880 |
| M. Calfin | 26 | 0.880 |
| X. Ythar | 24 | 0.879 |
| R. Woods | 22 | 0.874 |
| R. McKanna | 76 | 0.872 |
| MIDI | 86 | 0.870 |
| Dark Luke | 36 | 0.866 |
| @@ | 60 | 0.859 |
| D. Bernardone | 354 | 0.858 |
| D. D'Amato | 154 | 0.857 |
| … | | |
| M. Simmons | 334 | 0.787 |
| Revolver | 25 | 0.775 |
| R. Konkul (Rynker) | 125 | 0.774 |
| K. Ward (☺) | 600 | 0.773 |
| Freyja | 89 | 0.766 |
| P. Mason | 14 | 0.765 |
| M. Puls | 167 | 0.759 |
| P. Shanklin | 37 | 0.756 |
| I. Pyles | 177 | 0.754 |
| Gazebo | 377 | 0.754 |
| zimlord | 217 | 0.747 |
| C. Foy | 205 | 0.739 |
| B. Abear | 29 | 0.733 |
| Blazing | 21 | 0.729 |
| Mootz | 29 | 0.715 |
| Insane Steve | 49 | 0.714 |

**Table 5.** The turniest and least turny step authors. Only authors with 10 or more charts are included.

**The Turniest Other Things.** Although our analysis picked out a good many turny charts, any ITG player will inevitably get tired of playing the same ones over and over. Hence, we are also interested in an aggregated turniness profile to show the average scores of many different stepcharts at once. We computed average turninesses for stepcharts grouped *by pack* and *by step artist*. Abbreviated results for the former are shown in Table 4 and for the latter in Table 5. For the sake of players who dislike turniness, we also list the lowest-scoring packs and artists in addition to the highest-scoring ones.

## 5. Future Work

**Jumps.** The most obvious next step for this work is to account for jumps. This seems quite easy: compute the two possible facings of the jump, depending which foot lands on which arrow, and use whichever facing produces the least turniness when compared to the previous and next steps in the sequence. Honestly, we should have implemented that for this paper, but we only realized how it should work after we finished computing our dataset; and owing to an inferior conference's deadline coming up in 10 days, we don't have time for re-running experiments (Anonymized 2016).

**Footswitches.** More difficult is the matter of deciding whether stepping the same arrow twice in a row constitutes a jack (same foot) or a footswitch. As we saw in Figure 11, our current approach of assuming it's always a jack can lead to some false-positive $\mathcal{T}$ values. Judging the presence of footswitches is a nontrivial problem even for human intuition, so we view this as an exciting area for future research. So as not to piss off the player, most stepcharts tend to feature only jacks or only switches, never both. Future extensions of our algorithm could exploit this convention to decide with high confidence whether or not an entire chart is footswitchy.
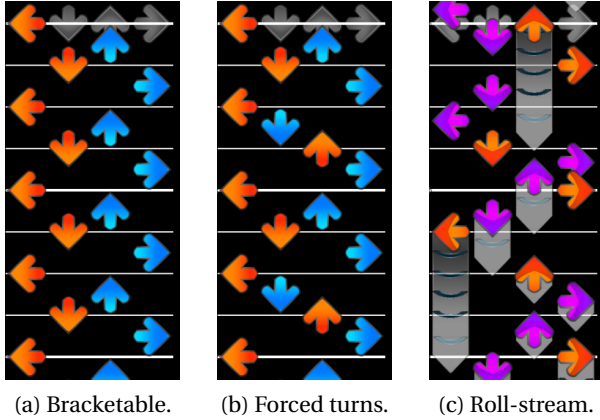
0xa9add8e9b72241b1e482140d31d75a186f59ef7c5ed16f5e1ecd031b7bb276d5

(a) Bracketable.    (b) Forced turns.    (c) Roll-stream.

**Figure 12.** Future work.

**Fractional Turns.** Experienced ITG players do not step with their feet on the middle of each arrow; rather, they try to step on the edges and "brackets" between arrows, hitting each step with heel or toes only to minimize the distance their feet must move. This tendency distinguishes further granularity of turning than what we identify in this work. Consider the two streams shown in Figure 12 (a) and (b). Although they measure the same $\mathcal{T}$ value of 1, (b) is considerably more exhausting to step, as (a) allows the player to bracket the entire run. Future work could handle this by counting a 1-turn step as the full value only if the player subsequently continues turning farther in the same direction (which would force the player's feet off the brackets), and otherwise assign $0 < \mathcal{T} < 1$ for such a step.

**Holds and Rolls.** Many charts use holds or rolls to force the player to step all concurrent tap-steps with the other foot, which breaks our alternating-feet assumption. When multiple steps are concurrent with a hold, they must be double-stepped, and the facing for each computed using the hold's arrow rather than the most recent one. When multiple steps are concurrent with a roll, the facings are determined similarly, but repeatedly stepping the roll results in a stream, as shown in Figure 12(c) (MIDI and Basshunter 2007). Judging how hold-steps and roll-streams must be stepped is another exciting opportunity for future work.

## 6. Conclusion

Our analysis program can be found at `https://github.com/bblum/sigbovik/tree/master/itg/code`, and our full experimental dataset is available at `http://tinyurl.com/turniness`.

The value of this work is already abundantly clear, as through it, the author has found several new charts he is interested to play. We hope our published dataset can provide similar value to other ITG players worldwide. Please accept our paper. We worked hard on it.

## A. Most turny patterns

### A.1 Vanilla

Following are the most turny 18-step patterns that are totally vanilla, with $\mathcal{T} = 1.375$. (Manual inspection reveals the true turniness to be 4/3.)

URDULDURDULDURDULD DRUDLUDRUDLUDRUDLU

### A.2 Crossovers

Following are the most turny 18-step patterns that may include crossovers only, with $\mathcal{T} = 1.5$.

URDULDURDLDRULURDL URDULDRDLUDRULURDL
URDULDRDLURULDURDL URDULDRDLURULDRDLU
URDLDRUDLUDRULURDL URDLDRUDLURULDURDL
URDLDRUDLURULDRDLU URDLDRULURDULDURDL
URDLDRULURDULDRDLU URDLDRULURDLDRUDLU
URDLDRULURDLDRULUR URDLDRULURDLDRURDL
URDLDRULURDRULURDL URDLDRURDLDRULURDL
URDRULURDLDRULURDL DRUDLUDRULURDLDRUL
DRUDLURULDURDLDRUL DRUDLURULDRDLUDRUL
DRUDLURULDRDLURULD DRULURDULDURDLDRUL
DRULURDULDRDLUDRUL DRULURDULDRDLURULD
DRULURDLDRUDLUDRUL DRULURDLDRUDLURULD
DRULURDLDRULURDULD DRULURDLDRULURDLDR
DRULURDLDRULURDRUL DRULURDLDRURDLDRUL
DRULURDRULURDLDRUL DRURDLDRULURDLDRUL
LUDRULURDLDRULURDL LURULDURDLDRULURDL
LURULDRDLUDRULURDL LURULDRDLURULDURDL
LURULDRDLURULDRDLU LDURDLDRULURDLDRUL
LDRDLUDRULURDLDRUL LDRDLURULDURDLDRUL
LDRDLURULDRDLUDRUL LDRDLURULDRDLURULD

### A.3 Laterals

Following are the most turny 18-step patterns that may include crossovers and laterals, with $\mathcal{T} = 1.625$. (Manual inspection reveals the true turniness to be 8/5.)

URDLDRULRDLURLDRUL    URDLRULDRDLURLDRUL
URDLRULDRLURDLDRUL    URDLRULDRLURDLRULD
DRULURDLRULDRLURDL    DRULRDLURULDRLURDL
DRULRDLURLDRULURDL    DRULRDLURLDRULRDLU
LURLDRULRDLURLDRUL LDRLURDLRULDRLURDL

### A.4 270s

Following are the most turny 18-step patterns that may include crossovers, laterals, and 270s, with $\mathcal{T} = 1.625$. (Manual inspection reveals irregularity; these streams exhibit no core pattern that can be repeated to retain the maximum turniness.)

URDULDRLURDLRULDRU URDLDRULURDLRULDRU
URDLDRULRDLURULDRU URDLDRULRDLURLDRUL
URDLRULDURDLRULDRU URDLRULDRDLURULDRU
URDLRULDRDLURLDRUL URDLRULDRLURDULDRU
URDLRULDRLURDLDRUL URDLRULDRLURDLRULD
DRUDLURLDRULRDLURD DRULURDLDRULRDLURD
DRULURDLRULDRDLURD DRULURDLRULDRLURDL

DRULRDLUDRULRDLURD   DRULRDLURULDRDLURD
DRULRDLURULDRLURDL   DRULRDLURLDRUDLURD
DRULRDLURLDRULURDL   DRULRDLURLDRULRDLU
LURULDRLURDLRULDRU   LURLDRULURDLRULDRU
LURLDRULRDLURULDRU LURLDRULRDLURLDRUL
LDRDLURLDRULRDLURD   LDRLURDLDRULRDLURD
LDRLURDLRULDRDLURD LDRLURDLRULDRLURDL

## A.5  Spins

Following are the most turny 18-step patterns that may
include any pattern, with $\mathcal{T} = 2$.

URDLURDLURDLURDLUR   DRULDRULDRULDRULDR
LURDLURDLURDLURDLU LDRULDRULDRULDRULD

## B.  Related Work...?



**Figure 13.** We employ the techniques of (Hanneman
2010) for ITG machine translation (Karakos et al. 2008;
Goto et al. 2013).

## References

S. E. Akgul. This is how we DO in DDRillini. In *DDR + ITG Players
Facebook Group*, pages You want page numbers, bibtex? It's a
photo of a whiteboard from Facebook. Get it together., 2015.

K. Apostol.   *Brute-force Attack*.   SaluPress, 2012.   ISBN
613627454X, 9786136274546.

M. Curtin.   *Brute Force: Cracking the Data Encryption Stan-
dard*.   Springer-Verlag New York, Inc., Secaucus, NJ, USA,
2005. ISBN 0387201092.

A. for Double-Blind Submission.  Ode to reviewer two.  SIG-
BOVIK 10 (that's decimal, not binary), 2016.

D. F. Fouhey and D. Maturana.  Find a separating hyperplane
with this one weird kernel trick! Proceedings of the 7 th ACH
SIGBOVIK Special Interest Group on Harry Quechua Bovik,
2013.

C. Foy and K. W. ☺.  Utopia.  In The Groove 1, 2004.  Challenge
11.

I. Goto, M. Utiyama, and E. Sumita.  Post-ordering by parsing
with ITG for Japanese-English statistical machine translation.

12(4):17:1–17:22, Oct. 2013.  ISSN 1530-0226.  doi: 10.1145/
2518100. URL http://doi.acm.org/10.1145/2518100.

G. Hanneman. DeltaX: A new paradigm for machine translation.
A Record of The Proceedings of SIGBOVIK 2010, 2010.

L. hurtpiggypig.  Simply love.  Simply Love is a custom theme
designed for Stepmania 3.95 or ITG arcade machines., 2013.

D. Karakos, J. Eisner, S. Khudanpur, and M. Dreyer.  Machine
translation system combination using ITG-based align-
ments.   In *Proceedings of the 46th Annual Meeting of the
Association for Computational Linguistics on Human Lan-
guage Technologies: Short Papers*, HLT-Short '08, pages 81–
84, Stroudsburg, PA, USA, 2008. Association for Computa-
tional Linguistics.   URL http://dl.acm.org/citation.
cfm?id=1557690.1557712.

Konami.  Konami Corporation v. Roxor Games, Inc., Mad Catz,
Inc., and Redoctane. United States District Court for the East-
ern District of Texas, 2005.

Konami.  Brilliant 2U (orchestra-groove).  DDR 2ndMIX.  Chal-
lenge 8.

Konami. Do me (H.I.G.E.O. mix). DDR 4thMIX PLUS. Easy 3.

R. Konkul.     Rynker - call of the hound (tachyon ep-
silon 20) 88.21.   https://www.youtube.com/watch?v=
6yB5WmIBZPk, 2014.

MIDI and Basshunter.  Boten anna.  In The Groove Rebirth +,
2007. Challenge 11.

mute and C. Gotobed.  Sick dubstep track.  Mute Sims 7, 2011.
Challenge 6482.

Psyonix. Rocket league. Video game, 2015.

sssmsm and Palpable. Oedo hop. r2112, 2007. Medium 7.

StepMania 5.0. http://www.stepmania.com/, 2016.

thattagen.  [ddr] canblaster - "dawgs in da house" stupid dou-
bles chart (10).   https://www.youtube.com/watch?v=
CZlB16BroPg, 2014.

A. Vangpat. http://photos2.alanv.org/, 2015.

WinDEU and D. Ashura.  ΔMAX.  Tachyon Alpha, 2009.  Chal-
lenge 15.

Zetorux and WinDEU.   Team dragonforce.   https://www.
youtube.com/watch?v=uSfx5Zdgyfc, 2008.

# SIGBOVIK 2016 Artifact Evaluation

## Paper 12: Which ITG Stepcharts are Turniest?

**Claim 1:** *LURR DURR*

**EVALUATION OF CLAIM:** The evaluation committee tripped repeatedly while attempting to evaluate this claim, leading to the loss of several perfectly good "candles." The committee requests an instructional video on performing this maneuver, as well as a fresh 6-pack of Woodchuck.

Though the committee does concede evaluating this claim would have been easier if we werent halfway through the 6-pack of Woodchuck when we started.

**Claim 2:** *Real-world tracks achieve 81.33% of the theoretical maximum turniness.*

**EVALUATION OF CLAIM:** The evaluation committee has confirmed that .8133 * 2 = 1.6266 which is an appropriate rounding-down of the number 1.6267 that appears on the spreadsheet. We appreciate the author's willingness to round numbers in the least favorable a direction, a subject with which we understand systems researchers frequently struggle.

However, the artifact submitted contains no actual stepcharts, let alone songs, and thus we have no way to evaluate whether this claim is actually true.

We were about to reject your submission for this reason, but then a member of the committee discovered an algorithm for turning theoretical stepcharts into real stepcharts. Followed by the insight that 1.6266 <2.0, we discovered that the author is indeed capable of producing such a stepchart upon request.

**Claim 3:** *The author is capable of multiplying various numbers by 10000.*

**EVALUATION OF CLAIM:** We diligently verified your multiplication of various numbers by the common number 10000 and discovered that in every case the multiplication is correct. This is in stark contrast with another publication presented at this conference in which an author utterly failed to understand any calculus whatsoever.

For this reason, the evaluation committee has found it appropriate to present you with the inaugural Innovations in Trivial Mathematics award. Congratulation (yeah, you only get one).

In return for this acknowledgement, we request that the author leads next year's Trivial Mathematics workshop in order to improve the quality of mathematics submitted to this prestigious Conference.

**Claim 4:** *The authors claim to have implemented a turniness analysis in Standard ML.*

**EVALUATION OF CLAIM:** However, much to our disdain, the code appears to be written in Latin. Thankfully, one member of the evaluation committee remembered enough Latin from high school to translate the code into English. Please submit your code in English next time. The original Latin is reproduced below for posterity:

```
IFS=$\n
for i in cat /tmp/authors; do
output=grep "$i"
Downloads/Which\ ITG\ stepcharts\ are\ turniest_\ -\ Sheet1.tsv
| sed s/.*\t// total=echo "$output"
| wc -l; numer=echo "$output"
| sed s/\r//g
| tr \n +
| sed s/+$//
echo -ne "$i\t"
echo "($numer)/$total" | bc
done

IFS=$\n
for i in cat /tmp/packs
| sed s/\r//g; do
output=grep -- "$i/" which.tsv
| sed s/.*\t//; total=echo "$output"
| wc -l; numer=echo "$output"
| sed s/\r//g
| tr \n +
| sed s/+$//
echo -ne "$i\t"
echo "($numer)/$total" | bc
don
```

**OVERALL EVALUATION:** Many of the numbers reported in this artifact were dubious (see additional comments). However, the contribution of the turniness metric stands on its own despite

the errors in the data. We encourage the author to build on their turniness metric to produce metrics for difficulty and author name, two values which appear to be more subtle than turniness based on the present dataset.

We also commend the author for encouraging the evaluation committee to get up and move to the groove.

**ADDITIONAL COMMENTS:**

- The arithmetic facts cited in this evaluation are currently being formalized in Coq, but as of publication time the proofs remain incomplete.

- Some of the difficulty ratings seem quite dubious, most notably:

```
Name: DDR 1st Mix to EXTREME/Last Message/Last Message.sm
Rating: Challenge
Steps: 2

Name: In The Groove Rebirth +/Sounds of Life/Sounds of Life.sm
Rating: Beginner
Turniness: 1.2931
```

- The following tracks have foot ratings of "Challenge," "Easy," "Hard," or most surprisingly, "Edit":

```
CuoReNeRo MeGaPacK/Return of Salieri (Dukamok)/Return_of_Salieri.sm
CuoReNeRo MeGaPacK/DeathMoon (Braintrust)/DeathMoon.sm
fort rapids vii/(12) Be OK/Be OK.sm
rocky mount xi/(12) Be OK/Be OK.sm
In The Groove Rebirth +/Welcome to Rainbow -hardstyle remix-
   /Welcome to Rainbow (Hardstyle Remix).sm
CuoReNeRo MeGaPacK/Omega/omega.sm
CuoReNeRo MeGaPacK/Return of Salieri (Dukamok)/Return_of_Salieri.sm
CuoReNeRo MeGaPacK/Since 1983 (DukAmok)/Since_1983.sm
CuoReNeRo MeGaPacK/DeathMoon (Braintrust)/DeathMoon.sm
```

- 64 tracks had foot ratings exceeding the claimed maximum of 20, with some reaching as many as 80085.

- The hardest Medium track had a difficulty of 21, in excess of the claimed maximum

- The hardest Easy track had a difficulty of 19, also very close to the limit

- A prolific author named "" authored almost 2000 songs, nearly a fourth of the dataset. This seems worthy of mention in your paper.

- Only 1187 tracks are possible with two feet, and out of those only 972 are possible with one foot - I'm concerned about the accessibility of ITG to disabled people with only one leg. There were 0 tracks that could be done with 0 feet!

- Why are both of the turniest tracks listed Easy?

- Even for a paper focused on turns rather than RAMs, I would have expected a DDR2 specification citation at a bare minimum, yet there wasn't even a reference to the *original* DDR spec. Remember to do a sufficiently extensive literature search: if it wouldn't put the significant prestige of this conference in jeopardy, I'd personally threaten to reject any of your subsequent submissions that was guilty of such an omission.

0x55109bffa0d8acd1effc245a68a64a49bdf6f66eded5c218ca35ea5a099a25a8

# Languages: Human and... Inhuman

0x664749275fef50b1918c91b7c80b96e88b026764e033cd91ff1e14dc2b0b4ee8

0x33d6b42b5a07b84648ee90dc6d2e5374426457978f5f61340c56e0c09764225d

# Lambda Doge: A digital canine language

Joshua Suereth

March 11, 2016

## Abstract

Lambda Doge is the first programming language designed specifically for canine implementers. By applying the known simplicity of lambda calculus to the known structure of canine utterances, we are able to create a programming environment that is both elegant and turing complete.

## 1. Motivation

According to a 2012 report by the world health organization[1]:
> … estimates that there are around 78 million owned dogs in Africa, but that there may also be in excess of 70 million unowned, stray dogs as well.

According to a random quora infographic, there is an estimated 17 million programmers in the world[2].
If we had the opportunity to teach all stray dogs in Africa how to program, then this would have profound impacts, including:
- Less strain on the economy from freeloading pooches
- New "outsourcing tech mecha" in africa from low-cost high-quality canine coders
- Estimated 400% growth in computer book sales

So, there is a lot of reason to design a programming language for dogs.

0x5624027ece7ed2c355a33d1d07f4b41a0d8dea6b6dc0b340b7cda45a5a49e7ca

## 2. Language design

Designing a language for dogs is quite hard.  First we decided to take some of the best aspects of human-oriented programming languages and ensure that these elements also are there to assist dogs:

- **Statically typed**.  To err is human, but typing is also unnatural for canines.  We expect having a compiler with static types will dramatically help cut down on typos and errors in code.
- **Type inference.**  While static typing can dramatically reduce errors, having to enter more keystrokes is certainly not dog friendly.  The language should attempt to infer as much meaning from expressions as possible.
- **JVM-backend.**  Java remains one of the top used/known languages for college grads.  By designing a java-friendly language, we allow dogs easy access to the job market.

Let's look at a simple "Hello, world" program in lambda-doge:

```
WOW
main
MUCH PrintLn
"WUF"!
```

As can be clearly seen, the syntax and style of this program are completely natural to the language of doge, as discovered via internet memes.

## 3. Grammar

Every lambda doge program is built using expressions.  Expressions are one of: Let-expressions, Lambda-Expressions, Apply expression, Identifier references or literals.

A lambda doge let expression is used to define a function, and looks as follows:

```
WOW <name>
SUCH <optional forced type specification>
SO <argument name list>
```

```
VERY <body-expression>
```

Functions optionally take both arguments and types for the arguments.   If types are specified, they must be specified for all arguments.   If argument names are specified, the values of these arguments can be referenced within the body expression.  Let's compare Lambda Doge to the more human-friendly Haskell:

| Lambda Doge | Haskell |
|---|---|
| WOW Inc<br>SO x<br>VERY Plus x 1 ! | Inc x = Plus x 1 |
| WOW Inc<br>SUCH Int => Int<br>SO x<br>VERY Plus x 1 ! | Inc :: Int -> Int<br>Inc x = Plus x 1 |

Lambda doge also has lambda expressions, otherwise known as function literals. These look as follows (with comparisons to scala):

| Lambda Doge | Scala |
|---|---|
| MANY friends<br>VERY Happy friends ! | { friends => Happy(friends) } |

The MANY keyword denotes a function literal is coming.  All identifiers preceding the VERY keyword denote how many arguments the function literal takes.

We list a more complete lambda doge grammar for those interested in implementing their own.

### Lambda-Doge Grammar
```
expr := let-expr | lambda-expr | ap-expr | literal | id-ref
let-expr := 'WOW' <identifier> type-list? arg-list? Ap-expr
type-list := 'SUCH' <type>
arg-list := 'SO' <identifier>*
```

```
lambda-expr := 'MANY' <identifier>* ap-expr
ap-expr := ('VERY' | 'MUCH') id-ref expr* '!'
id-ref := <identifier>
literal := <int-literal> | <boolean-literal> |
           <string-literal>
```

## 4. Type System

The Lambda Doge type system attempts to blend power with simplicity, enabling canines the fastest possible ramp-up time to programming as possible.   The type system is composed of two types:
1.  Type constructors: T[a], T
2.  Type variable:       a

A Type constructor can have no type arguments associated with it.  In this case it is known as a simple type, and is considered "complete.".   Most literals in the language are typed into simple types:

| Literal | Type |
|---------|------|
| Integer literal | Int |
| Boolean literal | Bool |
| String literal | java.lang.String |

Functions are encoded in the language as curried instances of the type constructor Function[_,_].  Lambda doge provides a convenience syntax for function types, using the => literal.  This allows breaking the type constructor into argument + result pairs, leading to a more haskell-looking type syntax. The following table shows examples:

| Function | Type | Abbreviated |
|----------|------|-------------|
| WOW Inc<br>SO x<br>VERY Plus x 1 ! | Function[Int, Int] | Int => Int |
| Plus | Function[Int, Function[Int, Int]] | Int => Int => Int |

Lambda Doge performs type inference by running the hindley-milner-like type inference algorithm locally against each let expression.   If no type is isolated, then the type will remain as a type variable.  In lambda doge, all types that start with a lower-case letter are considered 'variables'  which may contain any type.

For instance, lists in Lambda doge are encoded as follows:

| List function | Type |
|---|---|
| Nil | List[a] |
| Cons | a => List[a] => List[a] |

So, when constructing a list, the first element defines the type of the list, e.g.

>     WOW mylist
>     VERY Cons 1 Nil !

Will infer the type of mylist to be List[Int].

One complication is how to represent Java types in Lambda Doge.   While static java methods can be mapped directly to lambda doge functions, there is the issue of how to handle methods. We use an old trick of defining methods as static functions which take the instance of the class as the first argument.   Here's some example mappings:

| Java Method | Java Qualifier | Type |
|---|---|---|
| java.lang.Object#toString | | java.lang.Object => String |
| java.lang.String#charAt | | java.lang.String => Int => java.lang.Character |
| java.lang.Integer#MAX_VALUE | static | Int |
| java.lang.Integer#parseInt | static | java.lang.String => Int |

Currently the type system does not take into account inheritance.  We think open recursion is hard enough for humans to understand and wish to spare our canine friends.  However, as time progresses, we believe there may be opportunity to explore a simple way of allowing "foreign types" to have full object oriented semantics, well preventing most of the headaches associated with such semantics and type inference.

# 5. Backend

The backend of lambda doge compiles all lambda doge files into Java .class files.   The mapping is done as follows:
- All .doge files are compiled to a single .class file of the same name
- All let-expressions defined within the file are encoded as static methods of the same name.
- Any let expression named "main" is encoded as a "main" method, as used by the Java Virtual Machine (JVM) to run a program.  I.e.  Lambda-Doge will force the type of the main let expression to conform to Java main standards.

Additionally, as closures are not natively supported (at the JVM level), Lambda Doge performs "lifting" of static method into java.util.Function<?,?> instances.

Here is an example:

        WOW four_in
        SO a b c d
        VERY Plus a
        MUCH Plus b
        SUCH Plus c d ! ! !

        WOW closure
        SO captured
        VERY four_in captured !

The first function, four_in, has a type Int => Int => Int => Int => Int, which takes four inputs before returning a value.   In Lambda Doge, functions are values and can be returned.  The function called closure has a return type of Int => Int => Int => Int, which means it returns a function which will take three inputs before returning a value.

Encoding this on the JVM is tricky because the JVM makes a distinction between methods and objects.  Methods are defined against objects and only methods perform behavior.  Objects define data, and only objects can be passed or returned as values.   Therefore treating a function as both behavior and data is not entirely native.

However, Java 8 now defines a java.util.Function<?,?> interface, as well as a LambdaMetaFactory that can be used to construct object-instances of this interface using methods.  The LambdaMetafactory can take a set of "bound" arguments, a method handle and will construct a function object which can be returned.   To make use of this, given the natural currying in Lambda-doge, we must adapt all instances of code where we must return a function object such that these instances only pass one value into a static method at a time.   To solve

this, we introduce an AST transformer that will expand any partial function application into N let expressions, where N represents the number of unbound arguments. Here's an example.

| Phase | Pseudo-Ast |
|-------|------------|
| *Parse* | let four_in(a,b,c,d) = Plus(a, Plus(b, Plus(c,d)))<br>let closure(captured) = four_in(captured) |
| *Lambda Lift* | let four_in(a,b,c,d) = Plus(a, Plus(b, Plus(c,d)))<br>let closure$four_in$1(a, b, c) = four_in(a,b,c)<br>let closure$four_in$2(a, b) = four_in$1(a,b)<br>let closure(a) = four_in$2(a) |

The runtime is then able to appropriately call either the static method for the let expression, or is able to call the apply method defined on the java.util.Function instance, appropriately.

A second complication in the backend is over boolean logic. While most of lambda doge is evaluated eagerly, it is a common optimisation when dealing with boolean if statements to avoid computation for one of the branches of logic. As such, Lambda Doge provides built-in function "ifs" having type Bool => a => a => a. Here, the first argument is a boolean condition. The function returns either the second or third argument, depending on whether the boolean condition is true or false. As an optimisation, this built-in function will avoid evaluating the expression for the second and third arguments until after checking the boolean condition.

Generally the lambda-doge backend is quite simple, and most of the features map quite naturally to JVM bytecode, as the bytecode is stack based.

## 6. Standard Library

Currently lambda doge comes with a minimal standard library, for use in prototyping[3]. Here's a table of all functions currently defined:

| Function | Type |
|----------|------|
| tuple2 | a => b => (a,b) |
| fst | (a, b) => a |

0x412572ca2f24b309a9ac827882bd6245cd7b70893826c5f9688aabbf267ceb6b

| snd | (a,b) => b |
|-----|-----------|
| Nil | List[a] |
| Cons | a => List[a] => List[a] |
| hd | List[a] => a |
| tl | List[a] => List[a] |
| Plus | Int => Int => Int |
| Minus | Int => Int => Int |
| Multiply | Int => Int => Int |
| Divide | Int => Int => Int |
| ifs | Bool => a => a => a |

# 7. Future Work

The language cannot be considered complete until pattern matching has been implemented in an elegant way.   As you may know, dogs are excellent pattern matchers.   Here is a common pattern match from a popular dog AI[4]:

```
def next_action(in: SensoryInput): Action =
 in match {
   case See(SomethingThatMoves()) => bark
   case Hungry() => bark
   case FullBladder() => whine
   case Nothing if NotSeenOwner => whine
   case Nothing => sleep
 }
```

We envision adding this to lambda doge using a haskell-style encoding of multiple let-expression definitions with the same name yielding a single physical let expression, where the arguments define the patterns, However given the nature of lambda-doge, the elegance of this approach is dubious, leading to much discussion within the canine community of the best way to bark at things which move.

# Conclusions

This language presents an ideal syntax and backend to allow the canine race to enter the workforce.  By focusing on Java, it allows them to immediately contribute to most financial and e-commerce institutions.  WIth some work, the language could even be extended into a native backend, Allowing dogs to work on mission critical software, such as military, transportation and medical equipment.  This is totally because dogs are really good programmers and not, like, some attempt to overthrow the dominant sentient race of earth or anything.  Woof.

# References

1. https://www.psychologytoday.com/blog/canine-corner/201209/how-many-dogs-are-there-in-the-world
2. Google for "How many programmers are there" and it totally tells you
3. Prototype implementation: https://github.com/jsuereth/lambda-doge
4. https://gist.github.com/anonymous/b3174de0aec16a8e1c7f

0x4551deac5bff00c0e0f8caa2853565f333a72885a0110b16f9e11c2b690e4a3c

# Making the Theoretical Practical:
# The Untyped $\lambda$-Calculus in C++

Jim McCann*
TCHOW llc

## Abstract

While of great theoretical importance, the untyped $\lambda$-calculus is not often used outside of a handful of toy examples.

Imagine our excitement when we learned that C++11 "supported lambdas"; finally, we thought, all the features of a **production** language – standard libraries, memory support, in-line assembly, syscalls, and strong typing – paired with the wonderful programming style afforded by the untyped lambda calculus. The possibilities seemed endless.*

Alas, our hopes were dashed when we learned that by "lambdas", what was actually meant were strongly typed anonymous functions which had reasonable capture semantics and generally meshed nicely into the flow of C++ programs, replacing cumbersome function objects and obtuse macros. In this paper we show how to snatch victory from the ashes of defeat by using C++ lambdas to implement *real*, untyped lambda calculus lambdas and binding. This implementation is based on an in-place execution model for the lambda calculus that uses an *argument stack*.

**CR Categories:** ⸏⸏⸏ [Blank]: Blankings—Blank

## 1 Introduction

The $\lambda$-calculus [**?**] is a model of computation that proceeds through substitution ("$\beta$-reduction").

$$x[x := r] \rightarrow r \tag{1}$$
$$(\lambda x.y)z \rightarrow y[x := z] \tag{2}$$

While it may be straightforward to implement this in terms of copy-pasting of program fragments, self-modifying code can often result in performance penalties owing to instruction cache flushes. Therefore, we instead introduce an execution semantics for the $\lambda$-calculus involving a stack and instruction pointer.

Our execution model is similar to the standard reduction, in that our instruction pointer will move through unbound variables within the $\lambda$-calculus term in the same order that

---

*e-mail: ix@tchow.com
*Or, at least, roughly stack-depth.

the traditional substitution-based semantics would have after substitution.

And when we say "our", we're probably neglecting some background work but it's late-o'-clock so.

Y'know.

Sorry about that.

## 2 Background

The $\lambda$-calculus was introduced by Turing [**?**], and – famously – proved to be equivalent to the much more practical model of computation proposed by Jacquard [**?**]. It has enjoyed an important place in compuphilosophical cannon since, with the exception of a period in the mid-1700's it was briefly suppressed by the breakaway *Mechanists*, Lovelace and Babbage.

We shall neglect to mention von Neumann, nor the entirety of the nineteenth through twentieth centuries in this discussion. Suffice it to say, LISP was a bad idea [**?**].

In the present day, reliable complex systems [**?**; **?**] are engineered in reasonable and entirely satisfactory languages like C++ and its poor imitator Scala [**?**].

## 3 Execution Model

While the conventional "execution" of lambda term involves $\beta$-reduction – that is, finding applications and bound variables and performing substitutions – our execution model is more straightforward, involving two stacks and an instruction pointer.

We first describe the basic operations in terms of this model and then formalize the semantics:

At the beginning of a lambda term, the instruction pointer is set to the first character of the term, the argument stack is empty and the return address stack is empty.

```
LX.body
```

A $\lambda$ term. When encountered, pops the top value from the argument stack and stores it in a location (memory or register) which will be referred to as *X* within *body*. These addresses can be statically allocated in a pre-pass of the term since they are purely based on position within the uninterpreted term.

```
std::vector< std::function< void() > > arguments;
#define APPLY_BEGIN  ([&]( std::function< void() > fn_){ arguments.emplace_back(fn_);
#define APPLY_MIDDLE  })([&](){
#define APPLY_END  });
#define LAMBDA( X )  auto X = arguments.back(); arguments.pop_back();
```

**Figure 1:** *Encoding of untyped lambda calculus into typed lambda expressions in C++11.*

When *X* is encountered in *body*, the address of the next instruction is pushed onto the return address stack and then the instruction pointer is moved to *X*.

$$(func{:}arg)$$

An application. At `(`, pushes the address of the `:` onto the argument stack. (That is, makes *arg* available for any inner lambda term.) At the end of *func*, execution jumps to after `)`. (That is, *arg* isn't executed inline.) Finally, when *arg* is being executed, just before `)`, the return address is popped and set to the instruction pointer.

## 3.1 Formal Definition

As is typical in programming languages papers, we formally define the execution model by providing x86-assembly-like definitions for each character:

```
; code for L
POP X
```

```
; code for when X appears in the body:
CALL X
```

```
; code for (
PUSH :label
```

```
; code for :
JMP :end
:label
```

```
; code for )
RET
:end
```

## 3.2 In C++

Our implementation of the untyped lambda calculus in C++ (Figure 1) consists of four macros and a global arguments stack. Essentially, the `APPLY_` macros capture the address of code by using two C++11 lambdas and passing the second as an argument to the first, which pushes it onto the argument stack. Each lambda uses automatic reference capture semantics, which allows it to communicate with variables from the outside scope, but also establishes in inner scope (which conveniently limits the lifetime of variables defined by `LAMBDA`).

And, best of all, it actually seems to work.

For example, it finally allows one to easily count to four:

```
int32_t count = 0;
APPLY_BEGIN
    APPLY_BEGIN
        LAMBDA( X )
        LAMBDA( Y )
        APPLY_BEGIN
            Y();
        APPLY_MIDDLE
            APPLY_BEGIN
                Y();
            APPLY_MIDDLE
                X();
            APPLY_END
        APPLY_END
    APPLY_MIDDLE
        cout << ++count << endl;
    APPLY_END
APPLY_MIDDLE
    LAMBDA( A )
    A();
    A();
APPLY_END
```

Which is equivalent to:

$$((\,\lambda\,X.\,\lambda\,Y.\,(\,Y\,(\,Y{:}X\,)\,){:}\,\text{cout} << ++\text{count} << \text{endl}\,){:}\,\lambda\,A.\,AA\,)$$

Which I think we can all admit is a lot less cumbersome than a for loop.

## 3.3 Safety

Intel hasn't made computers with auto-destruct instructions for years, and other manufacturers are taking their cue. We predict code produced in this way will be safe by 2014.

Of course, arguments are only removed from the argument stack by lambdas, so things might get really weird if you apply something without a lambda to something else. But that's the joy of the untyped lambda calculus. And, besides, optimizers thrive on undefined behavior.

0x7b79788053e2455d70493d5ea898a537e14f2f842be5f709d85c53722c021893

```
APPLY_BEGIN
    LAMBDA( A )
    APPLY_BEGIN
        A();
    APPLY_MIDDLE
        A();
    APPLY_END
APPLY_MIDDLE
    LAMBDA( Y )
    APPLY_BEGIN
        Y();
    APPLY_MIDDLE
        Y();
    APPLY_END
APPLY_END
```

**Figure 2:** *Yes, this compiles and does what you think it should, though not for very long (though* ulimit *can help).*

## 4  Limitations

This section intentionally left blank.

## 5  Bibliography

This section unintentionally left blank.

0x338a6374adb80a3e33f7b30715a6e45abdd0ff63c42bda2b00fba05598b77b88

# A Shortmantout

David Renshaw          Jim McCann

1 April 2016

## 1   Introduction

Let $\mathbf{L}$ be a set of words and let $P = p_1 p_2 ... p_n$ be a string of length $n$. $P$ is a said to be a *portmantout* for $\mathbf{L}$ if:

1. Every word from $\mathbf{L}$ is a substring of $P$.

2. For every $i \in \{1, 2, ..., n-1\}$, there exist $j \le i$ and $k \ge i+1$ such that the substring $p_j p_{j+1} ... p_k$ is a word from $\mathbf{L}$.

For example, if $\mathbf{L} = \{\texttt{daguerrotype}, \texttt{per}, \texttt{personalized}, \texttt{whimper}\}$ then $P = \texttt{whimpersonalizedaguerrotype}$ is a portmantout for $\mathbf{L}$. In fact, in this particular case it's not too difficult to see that $P$ is the shortest possible portmantout for $\mathbf{L}$.

The general problem of constructing a short portmantout for a given $\mathbf{L}$ was first studied by Murphy [4], who implemented an algorithm that produces not-absurdly-long portmantouts for a 108,709-word set called `wordlist.asc` [3].

In the present paper, we describe techniques that significantly improve upon Murphy's algorithm. We also describe a method for computing a lower bound on the length of the shortest portmantout for a given $\mathbf{L}$. In particular, for `wordlist.asc`, we present a portmantout of length 537,136, and we prove that no portmantout can have length less than 520,732.

## 2   Computing a Short Portmantout

Let $\mathbf{R}$ the set of words obtained from $\mathbf{L}$ by dropping any word that is contained in any other word from $\mathbf{L}$. We say that $\mathbf{R}$ is the *reduced word set* for $\mathbf{L}$. Note that we have the following implication:

$$\text{every word from } \mathbf{R} \text{ is a substring of } P$$
$$\Longrightarrow$$
$$\text{every word from } \mathbf{L} \text{ is a substring of } P$$

Therefore, to construct a portmantout, it suffices to to arrange the words of $\mathbf{R}$ in any order and then to connect adjacent words, where a *connection* between adjacent words $R_1$ and $R_2$ is either an overlap of letters from the end of $R_1$ and the start of $R_2$, or a sequence of zero or more padding letters that, when inserted between $R_1$ and $R_2$, preserves property (2) in the definition of *portmantout*. The length of the resulting portmantout depends on how much overlap is achieved and how little padding is used.

In broad strokes, our algorithm for finding short portmantouts proceeds as follows:

1: Construct a portmantout.

2: Randomly break some connections in the portmantout.

3: Randomly reorder the pieces and reconnect them into a new portmantout.

4: If the new portmantout is shorter than the old one, then keep it. Otherwise, keep the old one.

5: Goto 2.

0x4ccd69669eb3ec09434da6ad0e127cfc7b86169bf24a3fb135042d60e3ec1fdf

There are of course many important specifics about how these steps are implemented. For the full gory details, readers are encouraged to peruse our source code [2][1]. For the purposes of this paper, we highlight a few observations:

- The effectiveness of the algorithm hinges on the probability with which it breaks a given connection. In our limited experimentation, we've had success with the strategy of *never* breaking connections that have overlap length 2 or greater, and breaking each other connection with probability 1/5000. This strategy seems to occupy something of a sweet spot, at least with respect to `wordlist.asc`, but it's easy to imagine that more sophisticated strategies could do much better.

- The algorithm does not get stuck, per se. Rather, as the portmantout gets shorter, the algorithm takes longer and longer to find each successive improvement. When we terminated the algorithm after it had produced the portmantout printed in Appendix A below, roughly 10 minutes were elapsing between each improvement of a single letter, on average. Any improvements on the efficiency of generating new random candidate portmantouts would directly translate into an ability to compute shorter portmantouts.

- Our currently-implemented approach for reconnecting a portmantount is somewhat inefficient. Every time we need to connect two pieces, if no overlap is possible then we directly consult the word set to compute the needed padding. We speed up this process by augmenting `wordlist.asc` with Murphy's $26 \times 26$ last-letter-to-first-letter joiners [4] plus other useful connectors such as `madjust`, `erstoque`, and `seaqua`, and by storing that augmented word set in a trie data structure to make lookup-by-prefix fast. Even so, this reconnection process is the clear performance bottleneck of our implementation.

- The reduced word set $\mathbf{R}$ for `wordlist.asc` has 64,389 words. The entire matrix of optimal connection lengths between the $64{,}389 \times 64{,}389$ pairs of words in $\mathbf{R}$ can in fact be computed in a reasonable amount of time. Each length fits within a byte, so the whole thing can be stored in under 4 gigabtyes of memory. Once this matrix is constructed, we can forget about the the words themselves and instead perform our search entirely in terms of the indices of the matrix. Unfortunately, we have not yet actually integrated this pre-computed matrix into our search algorithm. Doing so would likely help a great deal in eliminating our current performance bottleneck of reconnecting broken portmantouts. Future work!

# 3   Lower Bound

Let $\mathbf{R}$ be the reduced word set for $\mathbf{L}$. Define a function $w : \mathbf{R} \times \mathbf{R} \to \mathbb{N}$ where $w(R_1, R_2)$ equals the maximum length of overlap between the end of $R_1$ and the start of $R_2$. For example, $w(\texttt{earshot}, \texttt{hotshots}) = 3$ and $w(\texttt{hotshots}, \texttt{earshot}) = 0$. We decree that a word is not allowed to completely overlap with itself, so $w(\texttt{hotshots}, \texttt{hotshots}) = 4$.

**Lemma 3.1.** *Let $R_1, R_1', R_2, R_2' \in \mathbf{R}$. Suppose that $w(R_1, R_2) \geq w(R_1, R_2')$ and $w(R_1, R_2) \geq w(R_1', R_2)$. Then*

$$w(R_1, R_2) + w(R_1', R_2') \geq w(R_1, R_2') + w(R_1', R_2)$$

*Proof.* If $w(R_1, R_2) > w(R_1, R_2') + w(R_1', R_2)$ then we are already done, because $w(R_1', R_2')$ is non-negative. If $w(R_1, R_2) \leq w(R_1, R_2') + w(R_1', R_2)$, then the situation looks like this:
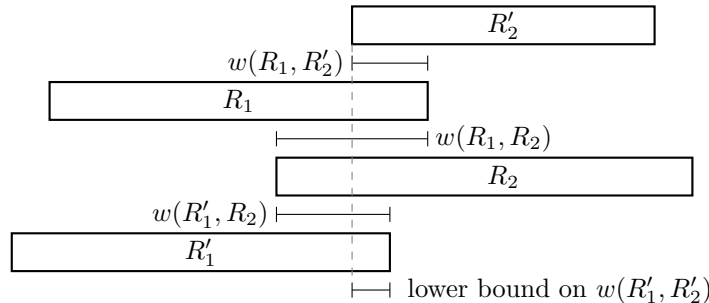


and by inspection it is clear that our desired inequality holds. (For an example where the left-hand side is strictly greater, consider $R_1 = \texttt{xab}$, $R_2 = \texttt{abx}$, $R_1' = \texttt{xbab}$, $R_2' = \texttt{babx}$.)

$\square$

0xa0c646eedce2246db1194a24901378087fc4c4fc26788a118f5fa131fb7ed2ac

Let $P$ be a portmantout for $\mathbf{L}$. Define $G = (\mathbf{R}, \mathbf{R}, E)$ to be the complete bipartite graph whose vertices consist of two disjoint copies of $\mathbf{R}$. Note that $P$ defines a matching $M$ on $G$, and $w$ defines a weight function on $E$ the edges of $G$.

Now, note that

$$
\begin{aligned}
|P| &\geq \sum_{R \in \mathbf{R}} |R| - \sum_{R_1 R_2 \in M} w(R_1, R_2) \\
&\geq \sum_{R \in \mathbf{R}} |R| - \sum_{R_1 R_2 \in \hat{M}} w(R_1, R_2)
\end{aligned}
$$

where $\hat{M}$ is a *maximal* matching, i.e. one that maximizes the sum of the weights of its edges. Therefore, if we can find such a $\hat{M}$, then we can compute a lower bound on the length of portmantouts for $\mathbf{L}$.

**Theorem 3.1.** *A maximal matching on $G$ can be computed using the greedy algorithm that at each step chooses a remaining edge with maximal weight.*

*Proof.* Suppose that one run of our greedy algorithm yields $M_0$, a matching that does *not* have maximal weight. Let $\hat{M}$ be a maximal-weight matching, and suppose furthermore that we've chosen $\hat{M}$ among all maximal-weight matchings to be one that agrees with our greedy algorithm for as many steps of the algorithm as possible. Let $R_1 R_2$ be the first edge that our greedy algorithm chooses that is not in $\hat{M}$.

Then there exist $R_1'$ and $R_2'$ such that $R_1 R_2' \in \hat{M}$ and $R_1' R_2 \in \hat{M}$. By the greediness of our algorithm, $w(R_1, R_2) \geq w(R_1, R_2')$ and $w(R_1, R_2) \geq w(R_1', R_2)$. Therefore, we can apply Lemma 3.1 to get

$$w(R_1, R_2) + w(R_1', R_2') \geq w(R_1, R_2') + w(R_1', R_2).$$

We can construct a new matching $\hat{M}'$ from $\hat{M}$ by replacing $R_1 R_2'$ and $R_1' R_2$ with $R_1 R_2$ and $R_1' R_2'$. By the above inequality, $\hat{M}'$ is also maximal. However, $\hat{M}'$ agrees with our greedy algorithm for at least one more step than $\hat{M}$ does, contradicting our supposition that $\hat{M}$ agrees with the greedy algorithm as many steps as possible.



□
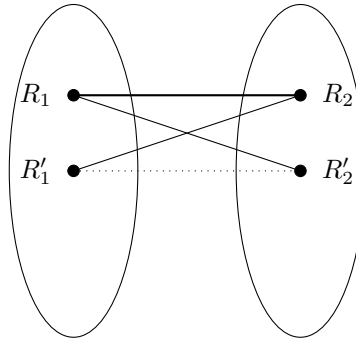
For `wordlist.asc`, Theorem 3.1 yields a lower bound of 520,732 on the length of all portmantouts.

# References

[1] portmantout git repository. https://github.com/ixchow/portmantout.

[2] shortmantout git repository. https://github.com/dwrensha/shortmantout.

[3] `wordlist.asc`. http://www.cs.cmu.edu/~tom7/portmantout/wordlist.zip.

[4] Tom Murphy VII Ph.D. The portmantout. In *Proceedings of SIGBOVIK 2015*, 2015.

# A   Appendix: Our Shortmantout

0xbe7bf0cef24409a94f62bd18d4fc95bb63b320ea3dcaa488d9ddf1456ca93e11

0x23c6f14e7ad2a6b2bc778621117d32d308936e92faa4649274393ac540c520fb

0xdbdde21f80eb4806973ed2502d5d0266bee7a181e329d1b609a9397e483f4629

0x8bee616411418c1e8d43b28d0ca5a162fbcb05bba66c2510347cb26b92a5048a

0xc7ac3261b91ff18f6a0888b08b8f6285b52f153076f9e2ce50ba7b063ba0f1dd

0x2fefc1d14f2eeffbba4060a12da37d0b18cd93b410da61d139c52dadb322cbed

0x7938b6cda72989f183f43d01b2426cf5235a5a699f48e7a162daa6f6836b186c

0xc3c338685db5a74cc0054b9cf8361e638b76b7495ee7450b7c67af209a40ebf1

0x9b35cb10c212b2f8f37e605829c9eed23798daa367925cbaa385b6e6c49aba58

0xb2e9ddc8e365c9abe8dd82347ea85184dfaac4cc8e297edad7215880df92d486

0x52e4b87ae8332582f74da728d30c4d093ee5ce26f2ea7b0e7058b3a4efafb302

0x99ba7d422db05770d1f502e8cc6d0b4eecbd5ed25e1cda9ba5c463fd75ee5469

0x88d34007416e70009614ed5ee1bc590881f346feebcbc122d93004be50449be1

0xdb8dd55392182930e5e92c2135b3c3b4a7c945aac78f02b84babdaaf7e7c5eb8

# HyperLink to DnD

# The glEnd() of Zelda

Dr. Tom Murphy VII Ph.D.*

1 April 2016

## Abstract

3D ZELDA

**Keywords**: small keys, boss keys, dungeon keys

## 1 Introduction

**1986. Hyrule.** The Legend of *frickin'* Zelda for the Nintendo *freakin'* Entertainment System. Need I say more? A *god damn* <u>gold cartridge</u>. Fortunes made just from melting the gold cartridge down to make gold teeth grilles, after carefully extracting and preserving the even more precious ROM inside. A die-cut hole in the box so that you could get a peek of the cartridge and presage that you were getting some solid gold. A die cut little window that you could palpate through the wrapping paper on Christmas Eve, presaging some epic thumb blisters in store for the coming weeks. Koji *freckin'* Kondo. Koji Kondo whipping up a nice 8-bit arrangement of Boléro as the theme music until realizing at the last minute that this music was copyrighted[1] and so instead composing its epic theme in *one day*??

A gold cartridge that contained ROMs and a little swallowable battery to keep the onboard SRAM powered up so that it could retain your epic save game. A battery designed to last 70 years. Nothing could cause you to lose your save game, even once you were half way through the Second Quest. Unless your little brother starts a completely new game and saves over your slot, earning him one of the most righteously deserved clobberings this side of Inigo Montoya. Saves right on top of your slot with a player called just `A` . Right over your



Figure 1: Technical diagram of mathematical equations.

slot, erasing it, and hasn't even picked up the *sword* yet. All you have to do is step on the black square in the first room and the guy gives you the sword and then dies. Jeez pooleaze. Saves right over yours with no sword even though there are TWO OTHER UNUSED SLOTS and you even wisely put your game in the third slot *exactly to avoid this kind of calamity.*

---

[1]Perhaps ironic, since Ravel's Boléro itself was composed as a result of Ravel getting cop-blocked.[1] And surely some Zelda knock-off since then has included a Muzak ersatz of the Zelda theme! What is the longest documented sequence of compositions due to Copyright restrictions?

1

Need I say more? Shall I say it? Apparently so. KIDS THESE DAYS. Kids these days and their three dimensions. You can certainly make a respectable case that the Zeldest of the series was *Legend of Zelda III: A Link To The Past*, or even *Legend of Game Boy Zelda: I forgot What It's Called*. But there seems to be an established consensus that *Legend of Zelda LXIV: The Ocarina of Time* is the true masterpiece; that while the NES Original indeed invented the genre, the technical limitations of the system handicapped it. You're talking to me about technical *frockin'* limitations? Like somehow appreciating the subtle wonder of the universe and its mechanics requires me to look at 3D tree people with tetrahedric clubs for hands. And they say this of the 64th game in the series? Please jouize.

A fine game, don't get me wrong. But was its theme song composed in one day? Did it come in a gold cartridge?

And the thing is: The original Zelda was built with enough forethought that it *can* be rendered in first-person 3D. So here I will make the game 3D so that it can be enjoyed by these kids and they can get off my lawn, tapdancing around playing their titular Ocarinas. I will make it possible to play the game in glorious HD, or even 4K resolution, and then some (Figure 1). And while we're at it, why not try to do it in a way that can play loads of NES games in three *fröckin'* dimensions? What's the worst that could happen?

## 2 Technical limitations

As presaged, the NES does have certain technical limitations, which I prefer to think of as constraints. Some basic details are important for understanding how this works.

The NES has a modest CPU that executes game logic. It's a little 8-bit baby puppy with access to 2 kilobytes of RAM. If you think about it, the NES's 256x240 pixel screen is already 61,440 pixels, which if it used the entire 2k of RAM to represent it, we'd have only $0.2\overline{6}$ bits of RAM per pixel, which makes no sense. At 1.7 MhZ, and a generous 1 instruction per pixel, we'd only be able to render 28 frames per second! This doesn't make sense, and this is because the CPU does not render the graphics on a NES.

The NES also contains a custom Picture Processing Unit, or PPU, the 8-bit baby puppy's baby ppupy brother. This thing outputs 60 frames of NTSC video per second. In fact, since the PPU also drives a CPU interrupt for each scanline and vblank, and most games



Figure 2: Pattern tables for the Zelda overworld. In Zelda, the top tiles are used for sprites, and the bottom are used for the background graphics. All tiles are 8x8 and use 4 color values (color 0 is transparent). These color values index into 4 sprite palettes and 4 background palettes, with some limitations.

use this for timing,[2] in many ways the CPU is sub-

---

[2]Fun fact: NTSC ("Never Twice the Same Color"), the video

servient to the PPU, kind of like the PPU is the *true* pure heart of the NES. In the context of this paper, you could say that the CPU is the Ocarina of Time and the PPU is the Legend of Zelda. It's just an analogy. The PPU is very complicated and has tetrahedra for hands, like clubs. The CPU communicates with the PPU not by drawing pixels, but by giving a high-level description of the screen (by writing it into PPU memory). The chief insight of this work is that the PPU representation can be interpreted and rendered in 3D instead of 2D. This allows us to use the exact original game logic and only change the way it's viewed. Sometimes this even allows us to see aspects of the game that are actually present but were not visible on NES hardware. Often it makes the game more difficult by making things invisible (enemies behind the player or behind walls) and inducing motion sickness.

### 2.0.1 Background

The PPU has two major drawing facilities: Background and Sprites. Both are made of "tiles", which are 8x8 graphics that are usually[3] stored in ROM. There are two tile sets, each with 256 tiles. Figure 2 shows the tile sets for the starting location in the Zelda overworld.

The NES background is described by the "nametable", which is an array of 32x30 bytes giving the tile numbers to fill the screen with. It's two rows shy of 256 probably for some reason having to do with the aspect ratio of TVs or number of actual NTSC scanlines, but also to make room for 64 bytes of attribute data. There are just two bits per four tiles. These two bits select which of the four palettes to use for each 2x2 group of tiles on the screen. It's an interesting hobby to look for the tricks that artists use to work around the constraint on the number of background colors.

Because of this array of tiles, NES games have to be built of high-level, block-like structures. In Zelda, this structure is particularly (but not unusually) regular. Since each 2x2 tile group (16x16 pixels) shares the same palette, the screens are built around 16x16 pixel grid-aligned blocks.

### 2.0.2 Sprites

Sprites, named in honor of the forest pixies whose homes were razed in order to make space inside the PPU for them, are more complicated.

There are 64 sprites, which are all turned on or off together. Each sprite is described by four bytes: Its $x$ and $y$ coordinates (these can be placed at arbitrary integer pixel locations), the index of the tile to draw, and an attribute byte:

|     | bit # | bit deets |
| --- | --- | --- |
| MSB | 7 | Vertical flip |
|     | 6 | Horizontal flip |
|     | 5 | something weird |
|     | 4 | I don't know |
|     | 3 | unused? who can say |
|     | 2 | bit 2; could be anything |
| LSB | 1,0 | Palette index |

Additionally, there are some global sprite settings. The only one that is important for this presentation is the "tall sprites" flag. If it's set, then sprites are 8x16 (drawn of two consecutive tiles) instead of 8x8. Sprites are drawn in a particular order. Barring some bizarre mischief, all of the sprites are drawn; you can't turn them off individually. Instead, you position the ones you don't want off the screen or make them be completely transparent. In order to make things more challenging, if there are exactly 8 sprites on a particular scan line, some really weird stuff happens.

Sprite Zero has some special properties. I don't want to tell you about the properties, I just wanted to make a soda joke.

## 3 Three to D—$u$ and $v$[4]

The game runs in an emulator, a version of FCEUX[2] that I have hacked up almost beyond recognition. In normal operation, it

- Emulates a complete frame of the NES. This includes the CPU and PPU operation for all 240 scanlines. The execution of a frame depends on the current controller input (which doesn't change as it runs);

- Extracts scene geometry and textures. This consists of two pieces: Boxes and Sprites;

---

standard used in the Americas, has a ∼60 Hz frame rate. But PAL ("Pointless and Lousy"), used in Europe and Asia, has a ∼50 Hz refresh rate. This means that many games run 16% slower in Europe than the US, so if you grew up playing there, you were playing on *Easy* mode. ;-)

[3]Some games trick the NES hardware by remapping tile memory (e.g. in response to a memory read), even during a scanline. I do not allow for such chicanery in this work.

[4]Spoiler alert because I think this joke may be too abstruse. "Free to Be You and Me" was a semi-famous kids record (here meaning actual vinyl) by Marlo Thomas. D is "dimensions." $u$ and $v$ are common names of the axes for texture coordinates in computer graphics.

0x15700c5aedc0f49938e223e9a35cc26c2d0b66b0f6dfbca27315cca70a796f4f

- Extracts the player location and orientation;

- Moves texture data to the GPU;

- Transforms the scene geometry into a textured model, sets the camera based on the player location, and renders.

I'll describe the interesting parts of these.

**Emulation.**  Well, this is not that interesting. I give the emulator its controller input and it does its thing, modifying the RAM and PPU and CPU state. I have been working on improvements to emulator technology but these are independent of this paper and better described elsewhere.

**Extracting boxes.**  Yeah! Okay! Boxes are the three-dimensional version of squares, and are what give the game its 3Dness. I read the nametable in chunks of 2x2 tiles, which is the size of one overworld tree (and the same size as Link). Recall that Zelda uses this grid size, probably because color attributes can only be set at this scale. These become $2x2x2$ cubes. The $x$ and $y$ coordinates are obvious (pixel positions divided by 16, since each 2x2 tile block is 16x16 pixels), but what should $z$ be? Zelda and other games tend to use tiles quite consistently to represent walls, obstacles, floor, empty space, and so on. This makes sense because a tree graphic, for example, suggests something to the player. So we simply build a file, `zelda.tiles`, which maps tiles (by index) to one of `wall`, `floor`, `rut`, or `unknown`. Floor blocks are placed at $z = 0$, which will be under the player's feet (we typically only see the top face). Walls are placed at $z = 2$, so that their bottoms are coplanar with the floor. Rut blocks are placed at $z = -0.5$; this is used for blocks that are part of the "floor" but that can't be walked on, like water. Because each box is actually made of 4 tiles, we just use the maximum $z$ value for the constituent tiles (they should normally be the same). It would make sense to develop heuristics for when games use tile-level resolution, even to generate primitives other than cubes.

This approach is a bit unsatisfying because it requires us to have specific knowledge of the game as well as do some manual labor. However, there really aren't that many tiles, and it's kind of fun. (In fact, you really only need to pick out a handful of floor tiles before the game is totally playable.) There is another problem, which is that many games switch tile sets, which may change whether tile `0x84` ("desert dots" in the overworld) is floor or wall. What happens if you go into a dungeon in Zelda, for example? I didn't even try that yet but I bet it's nuts! This can easily be fixed by identifying distinct tile sets (for example by using the source ROM address, or a hash of their content) and mapping them all.

However, there's some hope that this can be done in a much more automatic way. I describe an idea in Section 4.3.

**Box textures.**  The game is pretty understandable with just TRON-cubes, but it is better to texture them. We put the same texture on each face of a box; that texture is just the 16x16 pixel graphic consisting of the 4 tiles that made it, in the appropriate palette. It does *not* work to generate a bunch of tile-sized boxes (with the individual tile texture on all sides); these will only look right on the top face. Indeed, it is possible to get weird texture mismatches between these 2x2x2 boxes for basically the same reason. It usually looks pretty good, though.

Conceivably, every box on the screen could have a different texture, and conceivably these all change every frame. It is inefficient to move 960 distinct textures to video memory every frame, so instead we actually just move the rendered background (256x240) itself and record with each box its texture coordinates within. This is *much* faster, though it can create artifacts at the edges of boxes as an adjacent pixel peeks into the texture. I checked out Ocarina of Time and it is *loaded* with texture seam issues, so I figure this is how they like it.

**Extracting sprites.**  Sprites are the most difficult. Aside from just being more fiddly to implement, it is tricky to recover an important high-level structure. Although we know the location of each individual sprite and the two tiles (Zelda uses "tall' 8x16 sprites; see Section 2.0.2) that make it up, most game objects (like enemies) are in fact drawn with *two* sprites. These sprites could be in any of the 64 sprite slots (it's not at all obvious how Zelda allocates these, or that it would be similar for other games). Adding to the complexity, sprites like Link's sword or the pulsing balls of plasma that frogmen shoot at you from the water *just for being an elf* are just a single sprite, and bosses can be like 6 or more sprites (Figure 3). If we promote sprites to 3D without knowing which ones are supposed to be part of the same object, then the enemies get cut up along their sprite seems and it looks super wrong.

To fix this, we perform *sprite fusion.* We say that two sprites are part of the same object if they share
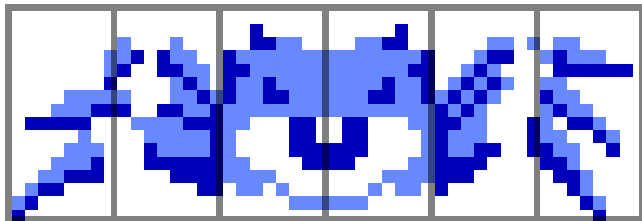
Figure 3: Link's spider nemesis, made of six 8x16 sprites. The sprites have adjacent edges, so they end up being fused into one merged sprite.

a complete edge (transitively). Since this allows the construction of very tall ($64\times16$) or wide ($64\times8$) sprites, we have to make sure that our textures are large enough to contain large sprites. Rather than try to pack them into a texture at runtime (computationally expensive since they can also be weird shapes), we just pre-allocate one texture of the maximum size for each sprite slot (16MB video ram; nothing). Sprite fusion, computation of the sprite size, drawing the sprites, extracting the geometry, and moving the textures to the video card are all intertwined. The simplified algorithm is

- Create 64 "pre-sprites", which are baby sprites. These know their coordinates and have the index of a "parent" pre-sprite (initialized to $-1$).

- While filling these in, consult a hash-map to see if there are already pre-sprites exactly aligned in each of the cardinal directions. If so, use a union-find– like algorithm[3] to efficiently update its parent index; each fused sprite has one canonical representative whose parent index is $-1$. Add the pre-sprite to the hash-map.

- Another pass pushes the min/max bounds to the canonical representative, so that we know how big the fused sprite is.

- Another pass draws all sprites into the texture data for their canonical representative, at the appropriate location. This pass must happen in the same order as on the NES hardware, so that overlay effects are correct.

- Finally, create proper sprites to return for each of the fused sprites. Copy the used region of the texture data for these to the video card.

Two types of sprite rendering are supported: in-plane and billboard. In-plane just draws the sprites on the floor or some other $xy$ plane of your choice. (This does not actually need sprite fusion, but it is not harmful.) Billboard sprites are drawn so that they face the player, like in Wolfenstein 3D. These are positioned at their $xy$ centroid and positioned in $z$ so that the bottom of the sprite is touching the floor.

**Extracting location.** Most NES games use very simple layouts for their game facts (i.e., global variables at fixed memory locations), since the processor is simple and memory light.[4] In Zelda, Link's $x$ coordinate is at `0x70` and $y$ at `0x84`. Since this is his top-left corner, we add 8 to each to get to the middle of the 16x16 pixel boy.

We also have to face in the camera in the same direction as Link, to make the perspective first-person. Link's orientation is at memory location `0x98`.

Link moves smoothly (with pixel resolution) around the screen, but turns abruptly. This is sickening to watch. I apply some smoothing to Link's angle, by treating the location from memory as a "target" angle and twisting towards it (along the shortest angular path) at a limited speed.

Knowing what memory locations hold the player's position and orientation also involves some manual work. You could imagine heuristics for trying to determine this, by seeing which bytes tend to go up when pressing right on the controller, and so on. However, it's easy to find these values manually or to even look them up for popular games.

**Rendering.** Finally, we can render the scene. I used "classic" OpenGL interface, even though it is terrible, for the sake of making good on the excellent pun that is this paper's title. The final result looks like Figures 4 and 5.

Rendering with OpenGL is quite straightforward given the geometry extracted from the previous sections. I try to preserve the pixel aesthetic by using `GL_TEXTURE_MAG_FILTER = GL_NEAREST`, but at this point we can use whatever approach to 3D graphics most titillates the youth. Full-screen multisample anti-aliasing? Sure. Screen Space Ambient Occlusion? I hear that this creates physically realistic shadows on the cheap, sure. Stencil buffer shadows? Volumetric fog? God rays? Depth-of-field? Lens flares? You got it.

## 3.1 Quirks

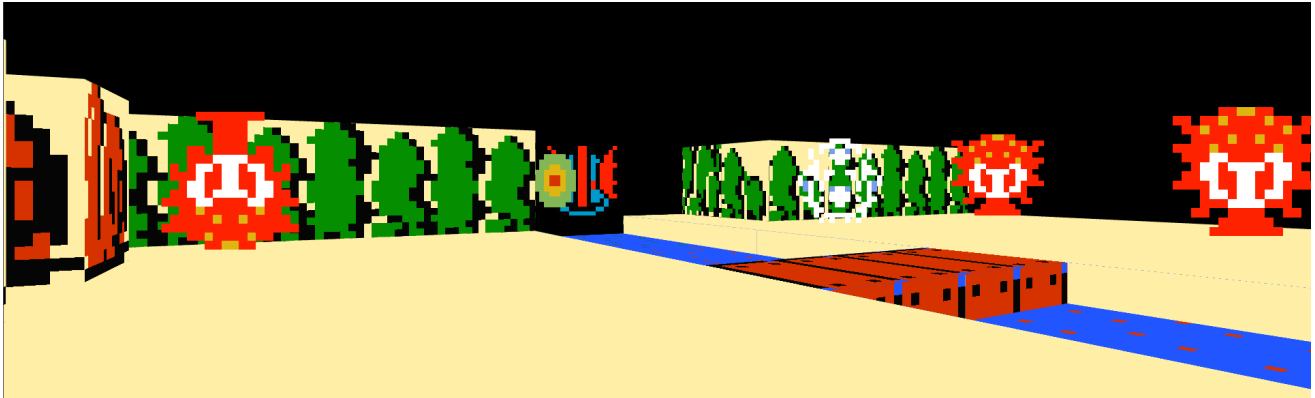There are some problems or quirks with the current approach.

Figure 4: A representative view of the Legend of Zelda in glorious 3D.



Figure 5: The first cave, where this man gives you a sword and then dies, rendered in 3D. Letters are treated as walls so you can see 'em upright, but then of course the second line is in front of the first. The 2D NES screen is superimposed on the top left for comparison.
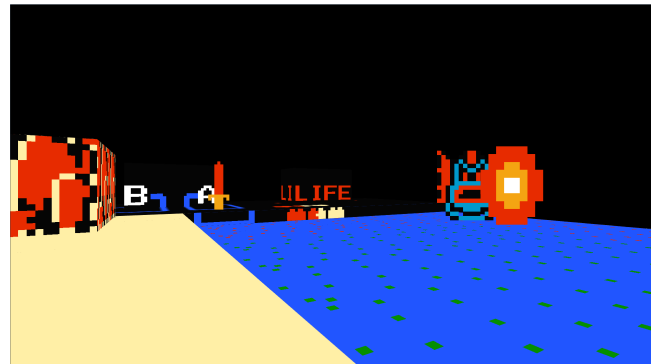


Figure 6: The game menu visible to the North because the ocean does not get in its way. Did you know that if you were able to see beyond the cluttering trees and mountains at the horizon of Earth, you'd see your own status menu in real life?

**Extramural geometry.** In Zelda, some of the tiles on screen are used to draw a menu at the top, which is not part of the gameplay area. This still gets interpreted as part of the geometry of the world, of course (Figure 6). It would be a simple matter to crop the region of interest (especially if done manually), and it would even be natural to overlay it on a heads-up display. But I like it over there. Why shouldn't menus be 3D as well?

**Positive self-image.** We place the camera at Link's position and orientation, so that the player "becomes" Link. However, Link is still represented by a sprite in the game, and we do not treat it specially. This means that the fused Link billboard is always in the scene geometry right at the location of the camera. The upshot

is that whenever the camera is facing south, an enormous Link graphic fills the entire screen, reminding you that he has always been following a few pixels behind you, haunting you (Figure 7). This sprite could be rejected by seeing that it is coincident with the player's location (which we extract) or by its particular content. We could also set the "near plane" so that nearby sprites are not visible. However, the everpresent looming spectre of a ghost mimicking our every move is a good reminder of our mortality, so I kept it.

**Surprising sprite fusion.** Sprite fusion works well for merging sprites that are part of the same logical object in Zelda. (It may not work as well in other games where e.g. a monster's legs are moved independently from its body.) However, it occasionally mistakenly

Figure 7: Look who's behind you! WE ARE FRIENDS FOREVER :-) :-)

fuses two sprites that just incidentally share an edge (Figure 8). This usually persists for just a tiny moment since the sprites are moving. But who's to say that as those monsters high-five, they aren't forming a momentary totem? Only Shigeru *froakin'* Miyamoto can say that isn't what he intended.
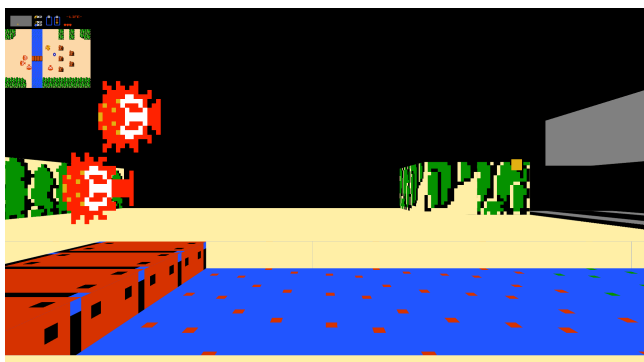


Figure 8: Two Octoroks that were incorrectly fused because they have constituent sprites that have adjacent edges (the fused sprite shape is not rectangular). The formation is a Hexadecirok.

**Texture orientation.** The textures of both backgrounds and sprites often have strange and inconsistent perspective ideas in video games. For example, the Octorok points its nose in the direction it's moving, because it is always viewed "from above". However, when Link is facing south, you see his front (not the top of his head facing down), and when facing west you see his side. This doesn't make sense, and makes it impossible to apply any consistent rule for mapping sprites into world orientation. This mostly just produces humorous

results, like when Link throws his sword and it moves away from him in first person view: It may be pointing towards the sky, towards the floor, or to either side, but never in the direction it's thrown. Even if we were to render the sword "in plane", its handle should always be oriented towards the camera (which would not be the case). No single policy will work for all cases, but perhaps heuristics can look for the use of the horizontal and vertical flip bits to figure out what sprites are rotatable, and then rotate them relative to the camera.

# 4 Future work

This work is literally in progress, and due to Draconian SIGBOVIK deadlines, this paper is figuratively already out of date by the time you read it. You can check out the project's home page

http://tom7.org/zelda

and maybe download it and play it, and probably watch a video demonstration. There may be some new features. Depends what I get to, you know?

## 4.1 Control

Currently, the game is viewed first person but played with native controls; this means that pressing "up" always moves link to the North, turning the camera if necessary. This is super weird. However, it is pretty easy to remap first person controls into native controls, as long as there is a way of implementing that intent (for example, it is not possible to "strafe" in Zelda). We can change the camera angle arbitrary, implementing "mouselook" or virtual reality support. Since the z axis is otherwise unused, we could allow the player to jump, though this would do nothing in the game itself.

## 4.2 Other games

This program was built in such a way that it should take minimal effort make other NES games 3D, perhaps with humorous results. There are some issues that need to be dealt with. A common type of game is the side-scrolling platformer, like Super Mario Bros.. At least two important issues are not yet handled: Mario has a different coordinate system than Zelda (it is "side view" rather than "top down"), so we need to support that rendering style and control/camera scheme. No big deal. Second, this game scrolls the screen rather than

using single-screen layouts; this uses the PPU in a different and somewhat more complex way. Many games will use advanced PPU tricks that cannot be interpreted in a generic high-level way.

### 4.3 A Link to the cast

One dissatisfying problem with the current approach is that it requires some manual effort to map different tile types to geometry. One could imagine various heuristics (even a image classifier) to provide defaults or to remove the manual input entirely. An exciting angle is to use the game's own understanding of these tiles. The main thing that differentiates a "wall" from a "floor" is whether the player can walk over it. Clearly somewhere in the game's code it "knows" what is floor and what is wall. We could extract that directly, but this would require a different manual step. Alternatively, we could determine it experimentally. A simple heuristic would keep track of what tiles the player has walked on. But we are not limited to what the player actually does; since we are emulating, we can also emulate counterfactual scenarios. Imagine that as we draw a first person view, we cast lines out from the camera towards the geometry, as in a ray tracer. Except that instead of tracing light rays until an intersection with scene geometry, we trace a hypothetical player path by emulating the player moving in that direction, until Link can move no more. This only requires understanding the bytes that correspond to the player's location (which we already need) and how the controller works.[5] Testing contact with sprites might give us useful heuristics about how they should be rendered as well.

There are many ways we can imagine this going wrong, but probably spectacularly so. Interestingly, this would allow us to render parts of the game that are not even visible in the PPU yet, since emulating the player's path ahead far enough would cause the game to switch to future scenes.

## 5 Conclusion

In this paper I made Zelda three-dimensional so that it can be enjoyed by the kids. I am fully aware that these "kids" are actually also full-grown adults who themselves resent a slightly younger generation who inexplicably think of Legend of Zelda Twilight Princess HD as

the canonical episode; these people should get to work at enhancing Ocarina of Time to add Amiibo support. While you're at it, get rid of the really long section where you have to be a dog. I didn't like that part.

## References

[1] Wikipedia, "Boléro — Wikipedia, the free encyclopedia," 2016, [Online; accessed 13-March-2016]. [Online]. Available: http://en.wikipedia.org/wiki/Bol\'{e}ro

[2] adelikat et al., "FCEUX, the all in one NES/Famicom emulator," http://fceux.com/.

[3] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *J. ACM*, vol. 22, no. 2, pp. 215–225, Apr. 1975. [Online]. Available: http://doi.acm.org/10.1145/321879.321884

[4] T. Murphy, VII, "The first level of super mario bros. is easy with lexicographic orderings and time travel. after that it gets a little tricky," *SIGBOVIK*, pp. 112–133, April 2013.

---

[5]Most rays are diagonal. We can implement a north-northwest diagonal by repeatedly moving two pixels north, then one west. We could also set the player's position directly to a putative destination and then move in cardinal directions to test solidness.

# Student Success and Sorrow: The Determination of Professor Alignment and its Effects on Students

D. Daly, E. Binns

March 14, 2016

**Abstract**

The "lawful-chaotic, good-evil" alignment system has gained popularity in describing real world personalities. We continue this trend by demonstrating how to apply it to professors. We also introduce a continuous "orbital" description of alignment oscillation. The resulting system has potential to aide students surveying potential courses by providing them with a more practical understanding of the professors in question.

# 1 A Discrete Analysis: The Traditional alignment Chart

## 1.1 A Background on Alignments

In the popular role-playing game *Dungeons & Dragons*, characters are categorized along two axes: *ethical* and *moral*. There are three possible values for each axis. Thus, in this system, there are nine possible alignments.

The ethical axis describes one's adherence to rules. A *lawful* character follows the rules, a *neutral* character views them as flexible guidelines or is unaware of them, and a *chaotic* character willfully breaks them.

The moral axis describes one's consideration of other people. A *good* character does what they believe is best for the general good, a *neutral* character will try not to cause unjustified harm (but not push themselves to cause active good), and an *evil* character will engage in unjustified malicious behavior.

We use these alignments to describe important components of professors.

## 1.2 Alignments as a Description of Professors

The approach of professors towards their courses are extremely varied, but the most important factors can be described with the alignment system.

Regarding ethicality:

- Lawful professors follow the rules set by society. In addition to following university policy and implicit social rules, they run their courses with a clear structure that adheres to the general expectation of what a course should be.

1

0x25ab14206b79be878da4fc75424de8d2b70614c994bf4d0a41a64335f10826e1

- Neutral professors merely take the above rules as guidelines, not considering them heavily before taking action.

- Chaotic professors have one or more of the following qualities: their courses are unstructured; their schedules/policies are generally in flux; the rules that they most strictly adhere to or enforce are ones they personally create, which may be unpredictable or seemingly arbitrary; they take pride in instances where they've broken socially accepted conventions.

Regarding morality:

- Good professors, whether or not they succeed, run their courses with the goal of reaching as many students as possible, being as effective a teacher as possible, avoiding directly causing harm to their students (such as stress, lack of sleep, etc.) and causing the greatest quantity/quality of net learning as possible.

- Neutral professors focus more on "doing their job sufficiently" than on "optimizing results"– they let the responsibility of learning the information fall almost entirely on the students. Their decisions are seldom based on improving student experience, and are instead based on performing the technical requirements of their position.

- Evil professors intentionally make the course unnecessarily difficult for students. Rather than increasing difficulty to increase learning across the board, they increase difficulty to weed out the weak, or because they find enjoyment in their students' struggles. Other behaviors that harm the student, such as putting them down to lower their self esteem, or publicly embarrassing them, are also considered to be in the evil category.

  However, a professor need not always make decisions to harm their students to be counted as evil. They can certainly sometimes make decisions to benefit their students; the distinction that classifies them as evil is a willingness to knowingly make decisions that directly cause harm to their students and often take some form of pride in doing so. Professors who make their courses unnecessarily difficult or who otherwise cause harm to students but who do so without knowing or understanding the effect it has are not necessarily considered evil in this case, as the intent behind their actions is what matters most.

## 1.3   The Effect of Alignments

Obviously, a professor who actively works towards creating the best possible course for as many students as possible will tend to be a good professor by typical standards, and a professor that that intentionally harms students will tend towards being a bad professor by typical standards. The professors that are neutral on the moral axis, however, are less obvious in categorizing. For these professors who merely do their job instead of aiming for a particular result, their other qualities will have a much more pronounced effect on the course as a whole. Typically, True Neutral professors will fail to engage the students in any capacity– they embody the classic "lecture hall full of sleeping students" scenario. Lawful Neutral and Chaotic Neutral professors are much more unpredictable.

0x7f4a7bf31450fadf83688ca46a4890850a07a1e8f904fa986a03a93344708f23

The effectiveness of a professor's ethical category largely depends on the student and the personality of the professor. Lawful professors may be easier to follow, but that predictability can also leave students less engaged. Chaotic professors may engage students and excite them, but can be hard to keep up with, leaving the student disoriented. Neutral professors can find a balance between the two approaches, but could also end up possessing the worst of both worlds.

|  | Good | Neutral | Evil |
|---|---|---|---|
| Lawful | high | split | low |
| Neutral | high | low | low |
| Chaotic | high | split | low |

Table 1: The most likely public opinion for the quality of courses taught by professors of each alignment.

Evil professors tend to maximize the difficulty of their courses out of malice. Otherwise, the prioritization of passing the majority of students tends to lower the course difficulty. The exceptions to this are chaotic good professors, who are willing to take risks and employ the flexibility necessary to make their course as difficult as possible while still maintaining student satisfaction.

We give a more detailed guide to student alignment-based satisfaction with professors in Section 3.2.

# 2   A Continuous Analysis: Alignment Orbitals

## 2.1   Defining Alignment Orbitals

In the process of classifying professors, we began to notice certain patterns in how professors fluctuate between traditional alignments. These fluctuations tend to follow patterns that are predictable based on specific characteristics, prompting us to create a more detailed sub-classification system, in which each professor in a traditional alignment may also fall on a *alignment orbital* that provides more detail as to their approach to their courses. Identifying these orbitals may clarify the behavior of a professor, and ease both the process of classifying them and the process of predicting the nature of their courses.

It is important to keep in mind the following:

- Having an alignment that contains an orbital does not necessarily mean one lies on that orbital.

- Each professor on an orbital is still centered in a traditional alignment; in our visualization of these orbitals, this is represented as the nucleus around which a professor orbits. The location of this nucleus indicates the traditional alignment that the professor in question generally falls into.

## 2.2   Alignment Orbitals as a Description of Professors

The orbitals we have observed are as follows: the *neutral chaotic-evil* (NCE) orbital, the *good lawful-chaotic* (GLC) orbital, and the *evil lawful-chaotic* (ELC) orbital. Other orbitals are not discussed in this paper.

### 2.2.1 The NCE Orbital

The defining characteristics of professors on the NCE orbital is that they are self-centered and do not hold strictly to rules. The decisions they make are based primarily on their own best interest, and they are willing to take actions typically viewed as unethical for their purposes.

There are two key components in determining the nucleus for a professor in the NCE orbital:

- Professors centered in chaotic neutral will not knowingly break rules to either hurt or help their students, so the harm or benefit that befalls their students is coincidental.

  Professors centered in neutral evil will break rules that may cause harm to their students in some way, and while they are aware of this harm, they will still break the rule if doing so benefits themselves.

- Professors with a "chaotic neutral" center are generally more aware of the fact that they are breaking a rule when they do so and are proud of that fact.

  Professors with a "neutral evil" center are generally less aware of the face that they are breaking a rule and are instead just doing what they feel is best for their needs without necessarily regarding what is or is not against a rule.

### 2.2.2 The GLC Orbital

The key common characteristic of professors on the GLC orbital is that they are actively trying to benefit their students in the way they make decisions, and are aware of any rules that are in place.

Professors centered in lawful good tend to obey the rules imposed on them, but will occasionally break certain ones if they feel the benefit is worth the transgression, or if they feel the rule is particularly unreasonable. Professors centered in chaotic good tend to view their personal creed as above the rules imposed on them, but they have certain rules that they fully support and will always strictly follow.

It is important to note that this orbital does not actually pass through the neutral good alignment, as neutral good implies a lack of awareness or understanding of societal rules and whether or not they are breaking them. In contrast, professors on the GLC Orbital are often "reformers"– they desire a system optimized for student learning, and are willing to go against convention to achieve that.

### 2.2.3 The ELC Orbital

The ELC orbital operates in very much the same way as the GLC orbital, except that professors on this orbital are willing to knowingly make decisions that cause their students harm in some way. It is essentially the reflection of the GLC orbital on the opposite side of the morality axis.

# 3 Utilizing Professor and Student Alignments

## 3.1 Determining Your Alignment

There are many quizzes online to determine your alignment, but we present a simplified quiz for the specific purpose of determining student-professor compatibility. For each section below, add the number in brackets to your sum (initialized at 0) for the questions you answer "yes" to.

Ethicality:

1. Do you wish that every course was run with strict policies (with exceptions made only in extreme cases) and a pre-made schedule? [+1]

2. Do you think more professors should be given free rein on their courses, being able to make many decisions on a student-by-student basis? [-1]

(Sum meanings: -1 = chaotic, 0 = neutral, 1 = lawful)

Morality:

1. Were a professor to shame a student for asking a "stupid question", would you feel a sense of disgust towards that professor? [+1]

2. Would you be happy upon learning a hard-working, but slow, classmate who went to office hours was ignored the entire time because working with them would take too long? [-1]

(Sum meanings: -1 = evil, 0 = neutral, 1 = good)

As an example– If you answered the above questions as "yes, no, no, no" respectively, your sums for ethicality and morality respectively would be 1 and 0, making you lawful neutral.

We will continue to utilize the above numerical labelling of ethicality/morality (ie. -1 on the ethicality axis is "chaotic") in the following section.

## 3.2 Determining Your Favored Professor Alignments

Perhaps intuitively, students tend to favor professors whose alignments match their own. That is, a lawful student will favor lawful professors, and a chaotic student will favor chaotic professors. However, other important factors come into play: the student's ease of success, and the student's passion for the course subject. Our hypothesized measurement of student enjoyment of a course in relation to the ethicality or morality of a professor is depicted in Figure 1.

Recall from Section 1.3 that chaotic professors and evil professors tend to have harder courses. A student who must endure high amounts of stress in order to succeed, no matter how evil, will disfavor abnormally difficult courses due to the harm done to their own person. We theorize that morality has a greater predictable effect than ethicality on student happiness as a function of their ease of success, as the flexibility of chaotic professors can be an important asset for easily stressed students.

Additionally, students who do not enjoy, or even loathe, the course subject will want to put in as little work as possible. The flexibility of chaotic professors

does not aid them as much as it does high-stress students, as these students are unmotivated to seek help. Indeed, these students seek the easiest course possible, and the course that requires the least pre-planning. Generally speaking, the more lawful a professor, the easier it will be to work on their assignments at the last minute.
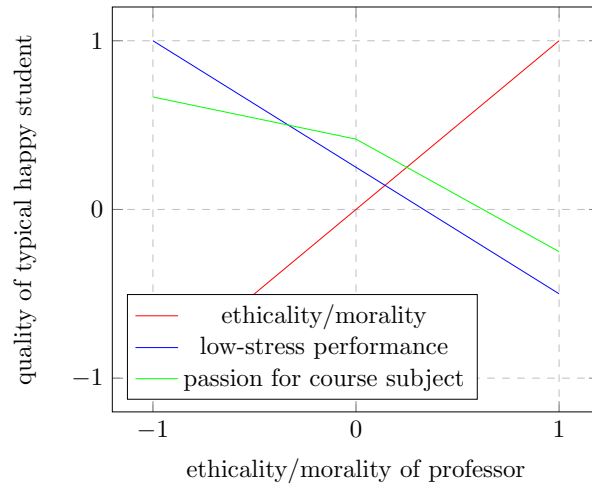


Figure 1: The theorized qualities of students that are satisfied with a professor of a given ethicality/morality. -1 indicates very poor performance, and 1 indicates very high performance; -1 indicates very low passion for the course subject, and 1 indicates very high passion for the course subject.

# 4 Conclusion

The alignment system as used in *Dungeons & Dragons* can be specialized in classifying professors. This specialized system can be used in determining the likelihood that a given student would enjoy a course taught by a professor. We believe that this system could be used by students as an alternative to standard Faculty Evaluations, and to sites such as RateMyProfessor, by providing a recommendation that takes into account the qualities of the inquiring student. Further research and discussion could be done on the dynamics in multi-professor systems ("the two-professor problem") and hierarchical course structures (ie. those including teaching assistants).

6

# Reduce, Reuse, Recycle

0xf443db57f0d15658fc95dcbc435c3989ba520eeebfc80013f0c8bb338b0d187f

0x6fa6c1fe75b5a34319737a9d07af77ad90f02f7c22371c1cf8bbf2e8f1816873

# PRESS RELEASE

March 31, 2014

Dear "SigBovik":

We regret to inform you that, after careful deliberation, the Rail and Transit Progress Advocacy Group (ratpag) delegation will not be attending your annual meeting. We simply have too much to do and believe that attending your meeting would be nothing more than a waste of our valuable time. In addition, you have stubbornly refused all requests by ratpag regarding our reasonable appearance fees and backstage requests. Your insistence that "this is just for fun" was not made clear during the paper submittal process and so now we will refuse *your* request to "lighten up."

From this point forward ratpag will utilize all its resources to test SigBovik's apparent 100% acceptance rate policy. You may have thought the three papers submitted by ratpag this year to be a bit overboard and off-topic, and to that we say: wait until next year. ratpag vows to inundate SigBovik 2015 with so many submissions—updates on Bridge-gate, more pet techniques, completely autonomous coffee makers, one-word non-joke papers, two-word non-joke papers, unabashedly plagiarized papers—so as to overwhelm SigBovik servers and communications.

Think we're kidding? Try sifting through 50 terabytes of this: :-) :-( ;-) :-P =-O :-* :O B-) :-$ :-! :-[ O:-) :-\ :'( :-X :-D o_O :-/ X-( X-P :-| <3 :-V XD :-Q :-@ :-C :-O :-3. And guess who just discovered SCIgen? The authors of the draft paper "Decoupling 802.11 Mesh Networks from B-Trees in Consistent Hashing," that's who. Warning: it's long.

So why are we telling you all this? Wouldn't it be easier for ratpag to follow the tried-and-tested surprise attack approach? Perhaps. But suppose ratpag left the door cracked just a bit. Suppose SigBovik was granted a way out of this total and assured destruction and suppose ratpag was generous enough to grant it? Is that something you'd be interested in? "What do we have to do to avoid total and assured destruction?" It's funny that you ask—we were just thinking of it—all you have to do is donate a sizeable sum to ratpag.wordpress.com—so sizeable so that we may change our domain name to our dream, ratpagdotorg.com, or perhaps, simply, ratpag.com.

The ball is in your court, SigBovik. Despite ratpag's shabby appearance, poor copy editing skills and often repetitive content we are surprisingly resourceful and, worse, have nothing but free time. We hope you make the right decision.

Respectfully,

*Nicholas Fudala*
Nicholas Fudala, founder

*Chester Francis*
Chester Francis

P.S. We're clearly joking. Unless we're not. Psychological warfare initiated.

0xfd989da19125a5ee09a171d22ec2f9ba4def667ccabc657595d637d5e7974bee

# Pittsburgh is a Grid

Stefan Muller
Carnegie Mellon University

April 1, 2016

## Abstract

Anyone who's been to Pittsburgh knows its streets are laid out in a weird way. Most of it looks like a grid locally, but then streets run into each other at angles downtown, streets intersect themselves, they do weird things to avoid going up hills, or sometimes they go straight up a hill. In this paper, we show that much of this confusion stems from the fact that Pittsburgh is a triangle and not a rectangle. We show this constructively, by exhibiting a mapping from Pittsburgh into the unit square. The images of many of Pittsburgh's streets under this mapping form a nice grid. Sort of.

## 1 Introduction

Historically, many great cities, and also some not-so-great cities, have been established near large bodies of water. Presumably, these locations were convenient for the transportation of goods and people to and from the city. I don't know; I'm a computer scientist, not a historian. Furthermore, through a combination of sheer luck and the fact that a large enough curve locally resembles a straight line, the waterfront of many such cities forms a flat border for at least part of the city. Examples include midtown Manhattan in New York, the Back Bay in Boston, Center City Philadelphia, all of Chicago, and all of those interchangeable cities where much higher-paid people with my training live. This sort of situation makes things very easy for city planners: put half the streets parallel to the water and the other half perpendicular. This leaves you with a nice convenient grid of streets so people can figure out where they're going, builds in nice water views on many of your streets, and gives the general impression that you know what you're doing and put a lot of thought into this.

Enter Pittsburgh. Pittsburgh is not bordered by a body of water. For a bunch of complicated reasons involving George Washington, the French, the British and probably Andrew Carnegie at some point, Pittsburgh is bordered by three bodies of water[1], but its designers still wanted it to feel like New York, and can you blame them?[Aurand(2006)] Now, having multiple waterfronts isn't necessarily a problem. If they're parallel to each other, you still get the nice street grid and visitors to your lovely city have the opportunity to enjoy both the more polluted waterway and the even more polluted waterway. In Pittsburgh, however, the rivers which border our city are not even parallel. In fact, at one Point, two of them intersect to form a Pointed shape which we will, of course, call The Acute Angle. This left the planners with a difficult choice of which river to use as the basis for the grid of downtown Pittsburgh. As it happens, they chose "both". Furthermore, Pittsburgh has many hills which would disrupt an orderly grid in any case.

In this paper, we construct an embedding of Pittsburgh into a flat, rectangular space. The images of the streets under this embedding form an orderly grid. This shows that Pittsburgh is just as beautiful and well-organized as other cities; it simply exists in a higher-dimensional, more complex metric space. This is probably related in some way to the technique of kernel methods, but I don't really understand those and so I didn't investigate that connection.

---

[1] Depending on how you count.

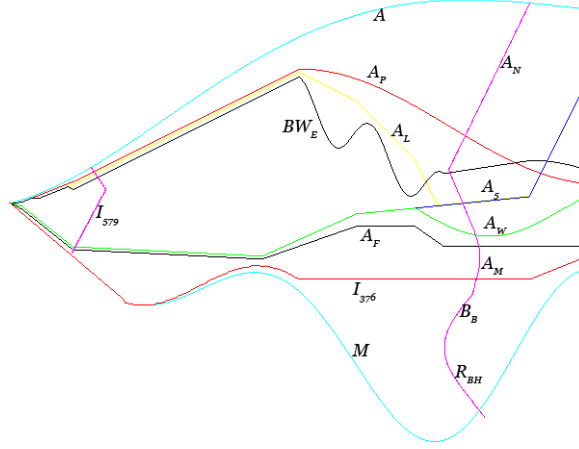0x420f863ba76ffcb6272435aed26b7c9b43b2242a2bcd8f854ed260d2ed1742af

Figure 1: A model of the space $\mathcal{P}$ with some representative subsets.

## 2    The Definition of Pittsburgh

We begin by formally defining Pittsburgh. In this paper, we will be working with the set $\mathcal{P}$, defined as follows

$$\mathcal{P} = \{(x, y) \mid 0 < x \leq 1 \wedge \mathcal{M}(x) \leq y \leq \mathcal{A}(x)\}$$

where

$$\mathcal{A}(x) = 2\left((1 - \cos\frac{9x}{2})/(4x + \frac{1}{200}) + \frac{x}{2}\right)$$

$$\mathcal{M}(x) = \begin{cases} -\frac{17x}{20} & x < \frac{1}{5} \\ -10(2x\sin(11x) + \sqrt{10x}) & x \geq \frac{1}{5} \end{cases}$$

Note that this model only accounts for the area of Pittsburgh between the Allegheny and Monongahela Rivers. This seems reasonable. After all, many cities have central areas that consist of grids as well as outlying areas that don't. In addition, we assume throughout this paper that Pittsburgh is planar. This is not strictly true, but close enough.

We also identify a number of subsets of $\mathcal{P}$, which we notate $A_5$, $A_P$, $A_L$, $I_{579}$, $BW_E$, $A_W$, $A_F$, $I_{376}$, $A_N$, $A_M$, $B_B$ and $R_{BH}$. A representation of this model of $\mathcal{P}$ is shown in Figure 1.

## 3    A 2D Embedding

We begin by constructing an embedding ignoring elevation and pretending that Pittsburgh is flat. The embedding is a function $E : \mathcal{P} \to [0, 1] \times [0, 1]$ First, let $d$ be the standard Euclidean distance metric on $\mathcal{P}$:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Let $m_{\mathcal{A}}(x, y)$ be the point on $\mathcal{A}$ closest to $(x, y)$:

$$m_{\mathcal{A}}(x, y) = \arg\min_{0 < t \leq 1} d((x, y), \mathcal{A}(t))$$

and similar for $\mathcal{M}$:

$$m_{\mathcal{M}}(x, y) = \arg\min_{0 < t \leq 1} d((x, y), \mathcal{M}(t))$$

For convenience, we let $d_\mathcal{A}$ and $d_\mathcal{M}$ be the respective distances:

$$d_\mathcal{A}(x,y) = d((x,y), m_\mathcal{A}(x,y)) \qquad d_\mathcal{M}(x,y) = d((x,y), m_\mathcal{M}(x,y))$$

We can now define $E$.

$$E(x,y) = \left( \frac{m_\mathcal{A}(x,y) + m_\mathcal{M}(x,y)}{2}, \frac{d_\mathcal{M}(x,y)}{d_\mathcal{A}(x,y) + d_\mathcal{M}(x,y)} \right)$$

The function $E$ maps $\mathcal{P}$ into the unit square by normalizing the distance between the rivers to 1, and it ensures that streets parallel and perpendicular to the closest point on the closest river will form a grid in the image of $E$.

# 4  A 3D embedding

The above embedding accounts for the unusual shape of Pittsburgh, but not its hills. We account for this with the observation that it's hard to go up hills and thus our distance metric should include vertical distance:

$$d((x_1, y_1), (x_2, y_2)) = \int_0^1 \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \frac{dH'}{dt}} dt$$

where

$$H'(t) = H(x_1 + (x_2 - x_1)t, y_1 + (y_2 - y_1)t)$$

The function $H(x,y)$ is an extension to our model of Pittsburgh:

$$H_D(x,y) = 400 e^{-\left(\frac{(x-\frac{1}{3})^2}{2(\frac{1}{6})^2} + \frac{(y-\frac{1}{15})^2}{2(\frac{1}{6})^2}\right)} \qquad H_{Sq}(x,y) = 450 e^{-\left(\frac{(x-\frac{3}{4})^2}{2(\frac{1}{4})^2} + \frac{(y+\frac{1}{15})^2}{2(\frac{1}{4})^2}\right)}$$

$$H(x,y) = L(700 + H_D(x,y) + H_{Sq}(x,y))/25000$$

The factor of 25000 is to provide appropriate scaling between the entirely arbitrary units ("feet") used in the elevation calculations above and the much more sensible units ("Pittsburgh-widths") in which $x$ and $y$ are expressed. The constant $L = 100$ is the "laziness" factor which approximates how difficult walking on hills is: the constant indicates that 1 vertical foot is approximately equal to 100 horizontal feet. Is this constant reasonable? Sure, because it makes the results look nice.

We've forgotten how to do integrals and that one looks particularly scary. Fortunately, we have computers to numerically aproximate these things for us. Figure 2 shows the image of the curves from Figure 1 under $E$. Figure 3 shows the same results, but with some minor noise cleaned up.

# 5  Conclusion

In this paper, we have shown that all that's required to turn Pittsburgh's complex mass of streets and avenues into the orderly grid enjoyed by many other cities is a 3D model of the city, line integrals and a surprising amount of compute power. There are many aspects of Pittsburgh that this model fails to consider. We have already acknowledged that Pittsburgh is non-Euclidean and therefore any attempt to project it onto a plane is necessarily an approximation. We also don't consider the topological structure induced by Pittsburgh's many bridges. For an exploration of this complex topic, see [Hanneman and Blum(2015)].

# References

[Aurand(2006)] M. Aurand. *The spectator and the topographical city.* University of Pittsburgh Press, Pittsburgh, PA, 2006.

[Hanneman and Blum(2015)] G. Hanneman and B. Blum. A constrvctive solvtion to the Königs-Pittsbvrgh bridge problem. *The ninth annual intercalary robot dance party in celebration of workshop on symposium about $2^6$th birthdays: in particular, that of Harry Q. Bovik (SIGBOVIK)*, 2015.
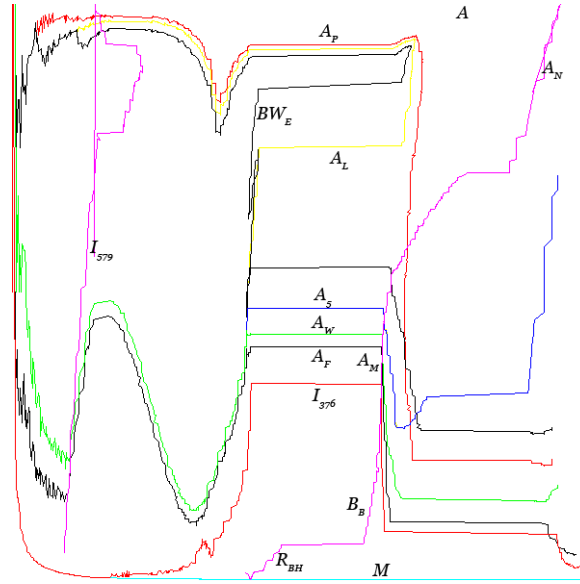
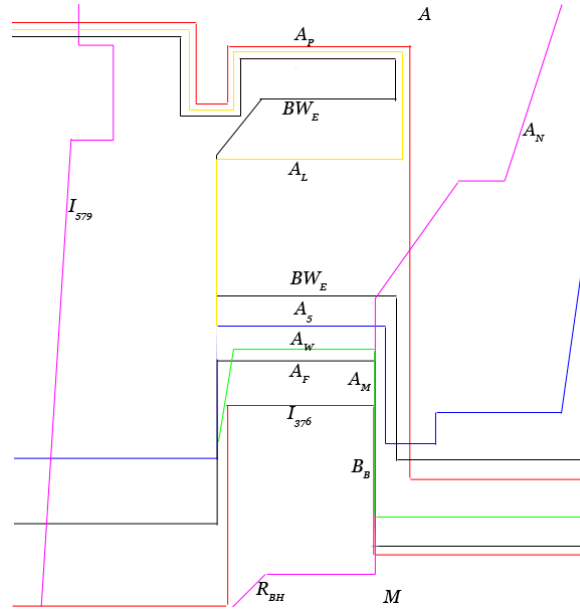0xa0ad2fbd2898412ab6010d4b20e406367a2ea25158aec038c4a507334fe7ea4c

Figure 2: The image of $\mathcal{P}$ under $E$.



Figure 3: The image of $\mathcal{P}$ under $E$, with some minor smoothing operations.

# SIGBOVIK 2016 Artifact Evaluation

Paper 10: Pittsburgh is a Grid

---

**Claim 1:** *The authors claim to have constructed a Pittsburgh-gridding algorithm in Standard ML.*

**EVALUATION OF CLAIM:** The provided code was not a Standard ML program, but was clearly an attempt at producing one. After extensive consultation with local French experts, the committee managed to extract a Standard ML program from the provided code, and proceeded under the assumption that this is what was intended.

In the future, we implore the author to provide proper Standard ML programs with their submissions. If the author is incapable of learning Standard ML on their own, we recommend the course 15-150.

**Claim 2:** *The authors claim that Pittsburgh is a grid.*

**EVALUATION OF CLAIM:** The evaluation committee has confirmed that submission contains grid-like shapes, but has failed to verify equivalent claims about Pittsburgh proper. We sent an undergraduate to search for grids, who then got stuck in an infinite loop on Boulevard of the Allies and nearly starved to death before being rescued by a crack team of topologists.

Before recklessly endangering Danny in this way again, we kindly suggest you evaluate your own work more closely before thrusting it upon this already-overburdened committee. In particular, it is well-known that all graphs form grids under the appropriate rounding conditions, and thus the veracity of your result is under question. As is the applicability, since the Peduto administration is a vocal opponent of the corn industry in general, and these "kernel methods" in particular.

**Claim 3:** *The authors claim an inability to perform integrals.*

**EVALUATION OF CLAIM:** This claim is pretty obvious, if you ask me. The authors did not even consider the numerical stability concerns inherent to all numerical integration methods, let alone give a convincing argument why their methods were stable. Instead of drawing on the extensive results in computer algebra, widely available in commercial systems such a Mathematica, they snobbishly implemented everything from scratch with inferior results. Like come on man, not

even Euler integration? Runge and Kutta are spinning in their graves.

**OVERALL EVALUATION:** Well, the PC told us to go easy on this one as a personal favor, so with great remorse we give this submission our approval. Pittsburgh is a grid, no matter how many weeks Danny spent on life support in UPMC Presby.

**ADDITIONAL COMMENTS:**

- While evaluations are normally provided as a free service by the organizers of the conference and their students, the medical expenses associated with this evaluation were well in excess of $100,000. This is partially covered by the author's travel allowance, but he will be billed the difference.

- You should seriously take 15-150. In all my years of teaching I have never seen such hideous SML code, not from the worst students, not even from the French. What kind of animal calls `float_to_int` over and over again on the same argument instead of let-binding like a civilized human being? And how lazy do you have to be to leave commented-out code in a conference submission? If youre going to insist on programming in French, at least tell us about your eclectic choice in graphing libraries. The quadtrees also seemed a bit excessive.

- Minus one style point for lines in excess of 80 characters

0x4a0bd59bde9c5f32766c440fa9baeaecb7c9012b3703325c65fafe95c1e3f75c

# How (Over)fitting

0xa961784f72704979411458d0b8fd8105a741ef78d3e47a13ef595765fbce98e3

0xa329c65e11214df956e99b8f2d1e361c69ffe4d01ca4bb2b5002d0db86e0bb92

# Reordering snacks is effective and just

Dr. Tom Murphy VII Ph.D.*

1 April 2016

## Concrete

It's time to get serious about *snacks* and *stats*.

**Keywords**: snacking, permutations, comestability theory

## 1 Introduction

Some workplaces of the future now offer a feature called Snacks. With Snacks, a kitchenette is provided near workers, which supplies running water and an array of small food packs. These foods are free of (dollar) charge, but various spoken and unspoken rules govern worker interactions with the foods.

This presents a challenge, since some foods are more desirable than others. Specifically, say that one yogurt food comes in four varieties: Classic, Diet, Cherry-Vanilla, and Caffeine Free. Furthermore, say that one user `<3`'s Cherry-Vanilla variety yogurt food and `>=3`'s Caffeine Free variety yogurt food. This worker is happiest when he begins his office work while eating Cherry-Vanilla. If there is no Cherry-Vanilla, the worker may resort to Classic yogurt food, reducing his task-ready disposition, and thus performance. If all flavors have been exhausted but Caffeine Free, then the worker may take no yogurt at all, knowing that Caffeine Free brings more displeasure than even hunger. This creates a negative affect, which may cause the worker to actively sabotage the work of his peers. This provides poor Return on Investment (ROI).

One strategy is for this worker, who we will call Sal, to take all of the favored Cherry-Vanilla foods from the kitchenette to his desk at the beginning of the day. This strategy is called Hoarding. This ensures that Sal may eat all his Cherry-Vanilla flavors. However, this behavior is considered unfair, for one reason that all other workers are completely deprived of Cherry-Vanilla fla-

vor. It is perceived that Sal and all other workers should have equivalent access to the shared resource, except for the moment that he is selecting a food (for he is "first in line," and lines are fair). Moreover, Sal should take only one food at a time (for it is "Please help yourself. We ask that you take only one piece so that others may enjoy it as well," which is fair). We also perceive that Sal should take food only when he is actually hungry (for it is "waste not, want not," and you can't spell "aphorism" without "a fair is," mmm?).

Given these rules, there are still things that Sal can do to influence the chance that he gets the Cherry-Vanilla flavors. In particular, this paper investigates the strategy of Reordering, where Sal selects his favored snack, and also changes the order of the foods in the kitchenette. The thought is that while everyone retains equivalent access to the foods, other workers are less likely to select Cherry-Vanilla due to the decreased visibility and/or increased effort in finding them.

Note that the author does not reorder habitually reorder snacks; this question is of abstract philosophical interest. We consulted the wisdom of Judge John Hodgman, who ruled [1]:

> Why don't you just go out to lunch and buy the food you want with your own money?
> ... Stop with the personal e-mails and get back to work.

We did not find this to be a satisfactory argument.

In this paper, I first provide a short argument why Sal's behavior may be considered fair, using an unjustifiable but common assumption. I then give a formal model for Snacks, which can be used to conduct controlled experiments. I then show that under suitable conditions, Sal's behavior benefits both him and the workplace, in a Utilitarian sense.

### 1.1 Why Reorder?

As suggested before, Reordering benefits Sal because his favorite snacks are less likely to be eaten by others. The

0x3b9493db2dabe4026029c00894dba6b5d20a973e912e990ecf022730e59d56ac

policy can be thought of as fair because it still provides everyone equivalent access to the snacks. But did Sal make the snacks worse for everyone by putting the good snacks at the back? The problem seems intractible until we consider an unjustifiable but standard simplifying assumption: Each worker's preferences for snacks are independent and identically distributed. This means that Sal's preference for Cherry-Vanilla gives us no information about the rest of the workforce's preferences. Since all preferences are equally likely, reordering the snacks improves access to some snacks and hinders access to others; these effects cancel out. Thus under suitable assumptions, *any* reordering of snacks is fair and harmless to other workers. The reason why we believe that this particular policy is beneficial is that other workers may be indifferent among various flavors, and simply take the one that is most convenient. When indifferent, taking a variety that someone else substantially favors is globally sub-optimal. This is similar to the principle of "decoy beer", whereby cheap OK beer (e.g. Red Stripe) is placed at the front of the fridge in front of the premium good beer (which may be unsuitable for casual drinkers anyway).

We can go further and demonstrate this scientifically.

## 1.2 Game of Scones

To investigate whether Reordering helps Sal and the rest of the workplace, we could run an experiment. Unfortunately this would be very expensive; we would need to find many workforces that are comparable, with similar food preferences, randomly assign some to the experimental group (where some fraction implement the Reordering policy) and then somehow judge their happiness. This would take a long time, and if the policy or experimental controls turn out to be harmful, might impact real GDP. For the effect sizes we see later, a live experiment is unlikely to show significance.

Instead, we develop a simple model that captures important aspects of the Snacks program, implement this on a computer, and then run millions of simulations.

The simplified model is as follows.

A simulation consists of an array $S$ of shelves, each of which is stocked with different varieties of food (Figure 1). The varieties are just given as integers from $0$–$V_i$, where each food type (shelf) may have a different number of varieties. In an early version of the simulation each food and variety is given a name, so we might have a shelf consisting of many sodas, like[1]

---

[1] All of these brands now Copyright Trademark Patent Pending Dr. Tom Murphy VII Ph.D., 2016!
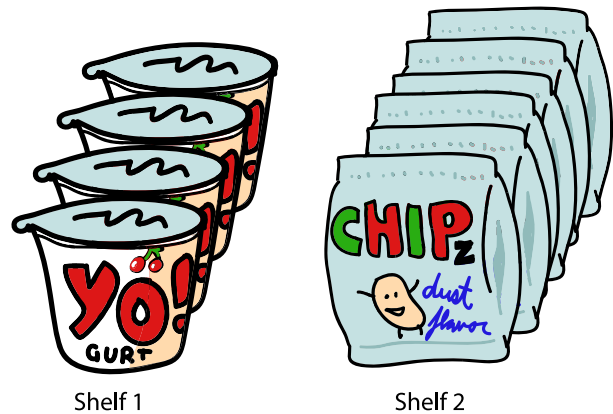


Figure 1: Two shelves with snacks. Shelf 1 has four snacks; potentially different types of Yö!gurt brand yogurt. Shelf 2 has six snacks, potentially different types CHIPz brand natural particle chips. I can tell you that the first three on Shelf 1 are Cherry flavor, and the last one is the Diet Cherry variety. All of the CHIPz are different varieties: Dust flavor, Dirt flavor, Sticks flavor, Cobweb flavor, Sand flavor, and Powder flavor. But because of the way snacks are arranged, you can only see the variety of the snack in front.

- Diet Kake

- Diet Thuck Lite

- Mango Slooch

- Mr. Sleepe Black

- Caffeine–Free Droob Ultra

- Caffeine–Free Spask Black

- Dr. Drarb Classic

- Drorp Lite

- Strawberry Sad

- Vanilla Grerb

- Cherry Prote Lite

- Diet Grobe

- Dr. Brosh Lite

- Duq Ultra

- Grape Ding Lite

- Mrs. Broop

- Diet Pap

- Grape Drax Classic

We also have an array $W$ of workers. Each worker has a preference function $P_{ij}$, one floating point value for each snack variety. This value may be negative, indicating an aversion to that snack. Nominally, these values are in dollars, for scale.

For simplicity, workers all get hungry at the same rate (although their hunger strikes randomly). When a worker hungers, she

1. Selects a shelf at random.

2. Sets her gaze upon the foremost variety on that shelf. She has some value $v$ for this item, given by $P$.

3. She can see the number of items on the shelf, but not what varieties they are. From this number, she estimates what value $v'$ she would get from skipping the current variety (for this round) and setting her gaze upon the next one.

4. If $v' > v$, she does so, and repeats from step 3.

5. If not, this is the food provisionally selected for this shelf, with value $v$.

6. If there are unvisited shelves remaining, she estimates the value of abandoning this shelf and trying the next one, $v''$.

7. If $v'' > v$, she moves to the next shelf and returns to step 2.

8. If she finishes with a selected food, she may reorder the items on the shelf of that food arbitrarily. She removes the selected food, if any, and eats it.

No player retains any knowledge of the organization of a shelf between rounds.[2]

We have not yet said how the worker estimates the value of a shelf. But observe the following properties:

---

[2]This is not an accurate assumption in reality, but seems to only disadvantage those who reorder snacks from benefiting from their own treachery. A very advanced strategy might rearrange items on the shelf in order to encode information about what has been reordered, for example, by coding a specific unlikely pattern at the front of a shelf (or prior shelf) to foreshadow the hidden booty. Ultra-advanced strategies might place misleading codes to confuse other workers and cause them to make suboptimal choices. Hyper-advanced strategies might use steganographic techniques or cryptographic signatures to hide codes or make them tamper-proof. Of course, this does not matter in a real workplace because workers can remember extremely simple facts themselves.

- If estimates are accurate, workers select a rational choice of food to maximize their own happiness.

- If a user has a dramatically favored snack, she is willing to search deep within a shelf for it.

- If a user has some snacks she favors and some she does not, she will be less willing to give up a good snack to find her favorite snack, because she might get stuck with a worse snack.

However, it also has an undesirable property:

- If a worker has a flat distribution of preferences, she will search the whole shelf. This is because there is no risk of getting stuck with a bad snack; she likes them all. This extends in a soft way to nearly flat distributions.

This does not match our intuitions of how real workers behave. Most of the time, an indifferent worker will just take a food that is "good enough;" this is known as "satisficing."[2] The argument for the global value of Sal reordering hinges on such indifference, in fact. To prevent this, a worker's estimate of the value of continuing to search the shelf will include a small cost to search each item. This can be thought of as the cost of the physical labor or the displaced opportunity cost, or an estimate of the risk that an interruption causes her to have to stop searching before she selects a snack.

The above requires an estimate of a shelf's value, both for the case where the worker may continue searching a shelf and the case where she continues to the next shelf. This can be computed with a recurrence relation. Since the worker cannot see beyond the snack her gaze is upon, this only depends on which shelf this is and the number of items on it. The expected value $E_s(n)$ for looking through $n$ items on shelf number $s$ is

$$E_s(0) = 0$$

$$E_s(n) = \sum_{i=1}^{V_s} \mathrm{P}(\text{item } i) \times \max(P_{si}, E_s(n-1) - c)$$

where $V_s$ is the number varieties for shelf $s$, $\mathrm{P}(\text{item } i)$ is the probability of selecting variety $i$ in the next slot, $P_{si}$ is the worker's preference for variety $i$ from shelf $s$, and $c$ is the small cost of looking at all. The content of the recurrence is simple: At each step, for each possible item, the worker can either can take that item with the value given by the preference function $P$, or keep going (but now there will be one fewer item).

Since we stipulate that the worker remembers nothing between rounds, the only probability distribution that makes sense for P(item $i$) is the uniform one, so the general case becomes

$$E_s(n) = \sum_{i=1}^{V_s} \frac{\max(P_{si}, E_s(n-1) - c)}{V_s}$$

Since this only depends on the preference function, we can compute this when the worker is born and print it on their birth certificate and employee badge.

## 1.3 Experimenting

I implemented the rules above in about 1000 lines of JavaScript, including a UI, random number generation, and the experimentation harness.[3]

Some other parameters need to be set: `NUM_SHELVES`, `NUM_PEOPLE` and `MAX_VARIETIES`, all self-explanatory; `MIN_ITEMS` and `MAX_ITEMS`, the bounds on the number of snacks initially stocked per shelf (it seems that `MIN` should be at least 2, for the program is Snacks, plural); `COST_TO_LOOK`, the penalty from the recurrence for estimating a shelf's value. We also have `PREF_MEAN` and `PREF_STDDEV` which are the parameters for the generation of preference functions (each is Gaussian—preferences are allowed to be negative). Finally, `MEAN_WAIT` gives the average amount of time between hunger events, wait times cannot be negative and so are distributed as $\Gamma(2.0, \texttt{MEAN\_WAIT})$. We run each simulation for three eight-hour work days.

I then generated random scenarios and evaluated different snack reordering policies. One extreme policy sorts the shelf in reverse preference order, so that the worker's favorite snacks are at the back. This seems unnecessary (takes $O(n \lg n)$ time, unless perhaps using "radix sort") and unrealistic. It also leads to substantial interference between workers, as one worker's resorting completely undoes another's. Another relaxation of this, more realistic, is where the worker only moves his favorite variety to the depths of the shelf. This is analogous to a "Move-to-Back List" [3], except not really, since the worker somehow finds all of the instances of their favorite variety in the whole shelf before moving them to the back. The most realistic policy generalizes this last one, and only moves the favorite snack when it passes some "outlier" threshold for how much it is preferred over the other snacks. This is the policy I ran many experiments for.

The first thing to notice from the experiments is that the policy doesn't even work! OMFG! For the reasonable parameter settings I started with, not only is the overall happiness (value of snacks eaten) neutral to slightly negative when the policy is in effect, but the worker employing the reordering strategy *also eats neutral to worse snacks*! This doesn't concur with anyone's perspective on the problem (except maybe John Hodgman's)—at least the "selfish" worker who reorders snacks should be benefiting from it, right?

Normally, this is where a heroic scientist would (a) implicate the model, seeing as how it fails to have intuitive properties[4] or (b) appeal that "it seems more work is needed in this fertile area of study" or (c) give up. But owing to Draconian SIGBOVIK deadlines, this heroic scientist resorts to another unjustifiable but common technique: "Tweaking" the model parameters until the experiments turn out as expected (i.e., positive). Better yet, this process can be automated!

**Search**. Overnight in 4 separate browser tabs I ran a simple "hyper-parameter search" to try to find the best settings of the parameters described previously. I selected parameter values uniformly at random, and then interpolated against the current local maximum (by averaging the random point with the best point for each successive "heads" coin-flip during intialization). The goal was to find parameters that simultaneously improved two scores in the experiment: The value of the snacks eaten by the resorting worker, and the total value of snacks eaten across all workers. (Specifically, I maximized the minimum of these two). The best settings of the model had these values:

| | |
|---|---|
| `COST_TO_LOOK` | 0.001 |
| `MAX_ITEMS` | 37 |
| `MAX_VARIETY` | 4 |
| `MEAN_WAIT` | 36305.715 |
| `MIN_ITEMS` | 3 |
| `MRATIO` | 1.010 |
| `NUM_PEOPLE` | 2 |
| `NUM_SHELVES` | 15 |
| `OUTLIER_RATIO` | 2.199 |
| `PREF_MEAN` | 1.320 |
| `PREF_STDDEV` | 5.625 |

---

[3]It can be found at `sourceforge.net/p/tom7misc/svn/HEAD/tree/trunk/snacks/`.

[4]I think that the model does not capture two important aspects. First is that Sal *does know* that he's reordered the snacks and can likely find his favorites by just looking immediately to the back. Second is that traces of the model suggest that most workers look through nearly all of the snacks due to the (rational) expectation that they will improve their selection. There does not seem to be a good setting of the `COST_TO_LOOK` penalty that suitably discourages workers from looking through the whole shelf while still allowing a resorter to find his deeply-placed favorites.

This an interesting instance: There are only two workers, but lots of shelves and lots of snacks. Most interestingly, the mean hunger time is very unusually large (in the highest 97th percentile of the probability distribution): 36,305 seconds is over 10 hours! This means that in the course of a three-day simulation, we only expect each player to eat about 2.5 times. There is a very good chance that the workers never interact (never visit the same shelf) and more than a $\frac{1}{4}$ chance that a worker never eats anything.

Repeating many more simulations with these parameter values suggests—but does not prove—that this may just be a nearly degenerate case in the simulation where the policies of the workers do not matter, and what we are seeing is pure noise. However, when I stopped the simulation arbitrarily after 1 million rounds, it produced nice smooth-looking distributions, consistent with a good sample size (Figure 2). The player implementing the resorting policy ate 0.015% better snacks on average, and the overall workplace ate 0.031% better snacks on average! This truly is a victory for snack reordering and the scientific method!            □

# References

[1] J. Hodgman, "Judge John Hodgman on tech-firm snacks," *New York Times Magazine*, Jan 2016. [Online]. Available: nytimes.com/2016/01/24/magazine/judge-john-hodgman-on-tech-firm-snacks.html

[2] H. A. Simon, "Rational choice and the structure of the environment," *Psychological review*, vol. 63, no. 2, p. 129, 1956.

[3] R. Rivest, "On self-organizing sequential search heuristics," *Communications of the ACM*, vol. 19, no. 2, pp. 63–67, 1976.

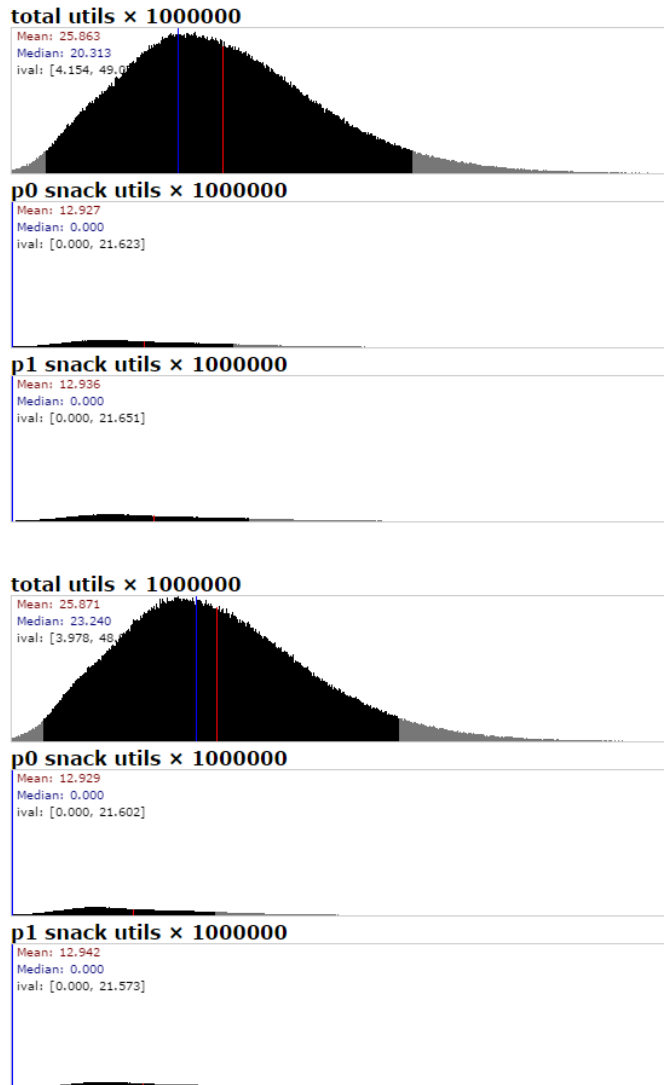[4] J. Kruschke, *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan.* Academic Press, 2014.

**total utils × 1000000**
Mean: 25.863
Median: 20.313
ival: [4.154, 49.0...]

**p0 snack utils × 1000000**
Mean: 12.927
Median: 0.000
ival: [0.000, 21.623]

**p1 snack utils × 1000000**
Mean: 12.936
Median: 0.000
ival: [0.000, 21.651]

**total utils × 1000000**
Mean: 25.871
Median: 23.240
ival: [3.978, 48....]

**p0 snack utils × 1000000**
Mean: 12.929
Median: 0.000
ival: [0.000, 21.602]

**p1 snack utils × 1000000**
Mean: 12.942
Median: 0.000
ival: [0.000, 21.573]

Figure 2: Histogram of one million experiments.

The top group is the control (no resorting) and the bottom is the experiment (worker 0 moves his favorite to the back of the shelf if its value exceeds the outlier ratio). Note how smooth the simulated outcomes are. This kind of graph is just the kind of statistics you write home about. Within each group, the big lump is the total snacks eaten by the two workers, and then the snacks eaten by each worker individually. Note a dramatic peak at the value 0 for the two workers, corresponding to *no snacks*; this happens because of the extreme hunger inteval of 10 hours. A overall value of zero is very unlikely because the simulation can only advance time (and thus end) when a worker has a chance to eat. Each histogram shows its mean and median, as well as darkening the 95% highest density interval [4]. An advanced technique would difference the values of the experiment and control and show a distribution of that statistic, but again, Draconian SIGBOVIK page size limitations preclude this. Normally we would use the 95% intervals to make the comparison between control and experiment, but this appears to be unreadable due to the very smooth, nice looking distribution occluding it. Therefore we compare the means, seeing 0.031%, 0.015%, and 0.046% improvement in snack consumption respectively for the three pairs. Note that resorting is actually *altruistic* in this experiment, helping most the worker who does not sort!

Without getting into too much math, regarding the frequentist standard of statistical significance typically used, we can say with confidence that $p > 0.05$.

# 0-Order Non-Linear Ordinary Differential Equations

## Dead Duck or Phoenix?

Oscar I. Hernandez, Daniel J. Packer

`ohernandez13@simons-rock.edu`, `dpacker14@simons-rock.edu`; $\Lambda\Delta M$.
Division of Science, Mathematics & Computing, Bard College at Simon's Rock, USA.

**Abstract.** A common problem in the field of applied mathematics is to find solutions to ordinary differential equations of order $n$. Solutions to first and second order differential equations are well understood. For example, consider the homogenous linear second order differential equation, $\frac{d^2y}{dt^2} = -y$, which has the general solution $C_1 \sin t + C_2 \cos t$ for any $C_1, C_2 \in \mathbb{R}$. Many generalizations can also be made to differential equations of higher order. However, perhaps the most important class of differential equations are those of order zero, whose solutions have been overlooked in most undergraduate-level textbooks. For example, $y = e^x$ is a solution to the differential equation $\frac{\csc(x)}{\log_2(e)} \log_2(y^{(0)}(x)) = x\frac{\sec(x)}{\tan(x)}$. In this paper, we will explore and attempt to provide methods of solution to these equations.

**Keywords:** ordinary differential equations, non-linear, non-homogenous.

## 1 Introduction

Perhaps one of the largest fields in mathematics is ordinary differential equations. However, this discipline has failed to offer solutions to differential equations of order zero, for mentions of them in the literature are scarce and subtle. For convenience, we shall now refer to ordinary differential equations as ODEs.

The general form for an $n$th order ODE is

$$y^{(n)} + \sum_{i=0}^{n-1} p_i(t)y^{(i)} = g(t) \tag{1}$$

where paranthetical exponentiation $(i)$ denotes the $i$th derivative. For example, the equation $y^{(1)}(t) + 17y = 37t + e^t - \sqrt[8]{\ln(\sin^2(\pi t))} - e^{\pi i} + 0$ is in the general form for a first order ODE. It follows that the general form for an ODE of order zero is $y^{(0)}(t) = g(t)$.

### 1.1 Overview

In Section 2, we will provide some methods of solution to ODEs of order zero. In Section 3, we will offer real applications of 0th order ODEs.

0x21c218f8f6b3a47aa10292f045d6a562dbf336aee78e4b08059568aa1f03e442

## 2 Methods of Solution

We ask the reader to look over any standard undergraduate textbook on differential equations. The methods we'll provide to solve ODEs of order zero should now seem familiar.

### 2.1 Integrating Factors

Perhaps the reader recalls that there exist standard methods for solving first order linear ODEs. Those are ODEs of the form $y^{(1)}(t) + p(t)y = g(t)$, and the method is that of integrating factors. The derivation will not be considered, but we shall remind the reader that the following is the general solution to the ODE above

$$y(t) = \frac{\int \mu(t)g(t) + c}{\mu(t)}$$

where $\mu(t) = e^{\int p(t)\ dt}$ is the integrating factor that facilitates integration.

**Theorem 1.** Let $\mu(t) = e^{\int \nu(t)}$ be the integrating factor for

$$\nu(t) = \frac{g^{(2)}(t)g(t) - (g^{(1)}(t))^2}{g^{(1)}(t)g(t)}.$$

Then, the solution to any 0th order linear ODE $y^{(0)}(t) = g(t)$ is

$$y(t) = \frac{g^{(1)}(t) \int \mu(t)g(t)\ dt}{\mu(t)g(t)}.$$

We leave the proof of this as an exercise to the reader.

### 2.2 Characteristic Polynomials

Homogeneous ODEs are ODEs of the form as in (1) for $g(t) = 0$. As the reader may remember, one of the general methods for solving homogenous higher order ODEs is that of the Characteristic Polynomial. This offers solutions to ODEs of the form from equation (1). For a 0th order ODE $y^{(0)}(t) = g(t)$, the following method may be used:

1. Assume $y$ can be written in the form,

$$y(t) = \sum_{n=0}^{\infty} C_n g^{(n)}(t)$$

2. Since $g$ is a constant function (by homogeneity), all derivatives, $g^{(n)}$ will vanish for $n > 0$. Thus,

$$y(t) = C_0 g^{(0)}(t) + \sum_{n=1}^{\infty} C_n \cdot 0 = C_0 g^{(0)}(t).$$

It is interesting to note that in the case of 0th order ODEs $C_0 = 0 = g^{(0)}(t)$. In Section 4, we examine the case where the 0th order ODE is non-homogenous

## 2.3   Undetermined Coefficients

The method of undetermined coefficients is used for finding particular solutions to non-homogenous ODEs. Suppose we have an ODE of the form $y^{(0)} = g(t)$. The following method may be used to solve for $A$.

1. We can assume the solution will be of the form $y(t) = Ag(t)$.
2. Plug this into our ODE and solve for $A$.

For example, consider the ODE $y^{(0)}(t) = e^t$. $Ae^t = e^t$. Setting the coefficients equal to each other[1], we see that $A = 1$. One can easily show that $A = 1$ works for all 0th order ODEs. Thus, $y(t) = 1 \cdot g(t)$ is a solution to all 0th order ODEs. It is worth noting that this method holds if $g(t) = 0$; in addition, if $g(t) = 0$, $A = 0$ also works, which makes $y(t) = 0 \cdot 0 = 0$ a solution.

# 3   Application

Solutions to 0th order ODEs are abundant in a multitude of scientific fields. Below are just a few examples.

**Example 1.** A classic problem in the field of physics is to solve a system at equilibrium. One of the most popular versions of this problem is to find the terminal velocity of a falling object. Let $\alpha$ be the coefficient of linear air resistance in units of $\frac{N \cdot s}{m}$, $v(t)$ be the velocity of the particle with units, $\frac{m}{s}$, and $mg$ product of the mass and gravitational acceleration on the particle with units, $kg \cdot \frac{m}{s^2} = N$. Thus, the full equation for the force on the particle will be:

$$F = ma$$
$$= v(t)\alpha - mg$$

However, if we assume that the particle is at equilibrium, then $F = 0$, so the equation becomes:

$$0 = v(t)\alpha - mg$$
$$v(t)\alpha = mg$$

As is clear to the most casual of observers, $v(t) = \frac{mg}{\alpha}$.

**Example 2.** Let $G$ be a good at manufacturing cost \$$D$ and suppose a group of 100 people are willing to pay \$$\frac{(D-0.1D)}{D}$. At equilibrium, the item will sell when $D = \frac{D-0.1}{D}$. If we solve this 0th order ODE, we see that $G$ will sell exactly when $D = 0$ or $D = 0.9$.

---

[1] We do this in order to not implicitly divide by 0 anywhere.

0x5148800a3975b04160ec09e31eb5282e9144d05d2b19970203148f8b41de3015

# 4 Open Problems and Additional Results

Recall the method of the characteristic polynomial developed in Section 2.2. The authors have noticed that for non-homogenous ODEs, $g^{(0)}(t)$ remains a solution, but have yet to provide a proof. One of the authors, Oscar Hernandez, offers \$4.20 to the first reader to prove the result for non-homogenous ODEs.

Recall the method of undetermined coefficients from Section 2.3. The authors have a strong suspicion that for a homogenous 0th order ODE, $y(t) = Ag(t)$ is a solution for any value of $A \in \mathbb{C}$. One of the authors, Daniel Packer, offers \$69.69 to the first reader to prove or disprove the statement above.

Over the years presenting on this topic, we have noticed the response to the lecture is usually:

> "Hey-o man! What a cool paper, lots'a stuff to think about. I knew about first, second, and higher-order ODEs, and I just learned about 0th order ODEs. What other kinds of ODEs could there be? Oh! What about negative-order ODEs?"

Our response is always as follows:

> "Please, get out of my house and stop wasting my time. Negative-order ODEs would have absolutely no merit to them."

That being said, we encourage the reader to consider fractional order differential equations; those are ODEs where the highest order derivative a non-negative rational number.

# 5 References

For their assistance with the development of this paper, the authors would like to extend their gratitude to the following people:

- each other,
- All members of $\Lambda\Delta M$ society, except for Connor,
- The readers, for taking their time to explore this nontrivial topic,
- The reviewers for considering this paper for acceptance at the Association for Computational Heresy Special Interest Group on Harry Q. Bovik,
- The authors' parents for inspiring us to be pioneers in mathematics,
- The creators of Pokemon for inspiring us to be the very best, like no [two] ever [were],
- Mike Bergman, Patrick Dragon, Eric Kramer for introducing the authors to differential equations,
- Jluv!
- our founding fathers and Michael Chang for continuous moral support and correction on 1 typographical mistake,
- The fans in our rooms making us feel cool,
- The chairs for offering their support,

0xb4c41e7decef71b2af08bbd997d2b2ae3e793929a0a64c3e33ca471c88fb4118

0x11d55f0295dae84f91b51fb70cf92a11460678f1fed311ecbabe467cd9425c1e

# Deep Spreadsheets with ExcelNet

**David Fouhey**
Carnegie Mellon University
Pittsburgh, PA
dfouhey@cmu.edu

**Daniel Maturana**
Carnegie Mellon University
Pittsburgh, PA
dimatura@cmu.edu

| | Caffe/Theano/MCN/etc | ExcelNet |
|---|:---:|:---:|
| Nerd coding stuff | ☹ | |
| Commie open source license | ☹ | |
| Backpropagation | ☺ | |
| Synergy | | ☺ |
| Enterprise ready | | ☺ |
| Turnkey solution | | ☺ |
| WYSIWYG Weight Editing | | ☺ |

Table 1. ExcelNet versus lesser platforms.

## ABSTRACT

We introduce ExcelNet , the premier solution for Deep Learning in Excel[1] Visit **deepexcel.net** to try it today!

## Author Keywords

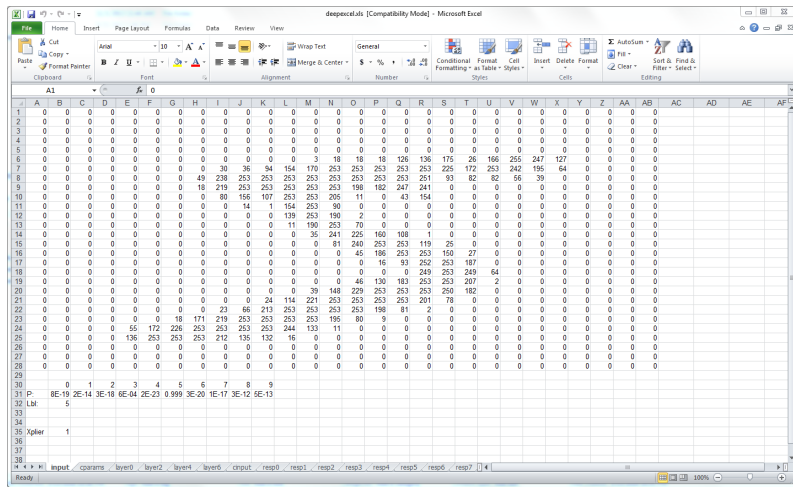Deep; neural; Excel; spreadsheet; Convolutional Neural Network

## INTRODUCTION

As anyone in machine learning and computer vision will tell you, Deep Learning is the right tool to solve the problem. And as anyone in business and finance will tell you, Excel is the right platform to implement your solution. But until now, there has been no way to do Deep Learning in Excel. To fill this gap we have developed ExcelNet , the ultimate synergy of spreadsheets and Deep Neural Networks.
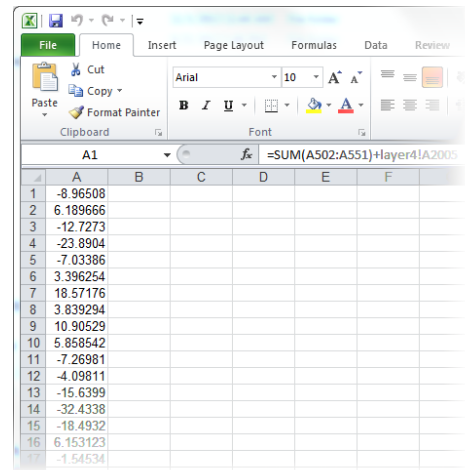
## APPROACH

Like the human brain, Excel has "cells". In Excel, as in the brain, these "cells" are organized in "columns". Our approach is to type *weights* into those "cells". We group the *weights* into "layers", and put each layer in a "sheet", which can be easily previewed and modified by the user. We then leverage the sophisticated multiplication and addition capabilities of Excel to multiply and add these *weights* to perform advanced tasks such as recognizing handwritten digits [3].

---

[1] Or LibreOffice, if you're an open source communist. Excel is a trademark of Microsoft, we believe.

As summarized in Table 1, this approach has many advantages over other complicated solutions like Caffe [2], Theano [1], or MatConvNet [4]. ExcelNet does not have backpropagation, but it is not needed, because the interface puts the weights at the tip of the user's fingers! To help you get started though, we provide some initial weights using a strategy similar to the "pre-training" process in common use: we build a CNN in MatConvNet [4] that recognizes digits and transfer these weights to an Excel spreadsheet. We then allocate activation maps in other spreadsheets and define these activations using Excel's powerful functions. Once done, Excel performs all memory management and calculations.

The advantage is that the power of Deep Learning is now in *your* hands. There is no need for worrying about CUDA or BLAS: Excel's proven and battle-tested numerical codebase handles it for you! There is no need to worry about platforms: Excel provides the necessary abstractions, and you can seamlessly transition between Windows, Linux, and Mac! This enables the ultimate in computer vision deployment flexibility: you can develop on a Mac, and deploy simultaneously to client and cloud devices.

We show screenshots of ExcelNet in action in Figure 1. Figure 1(a) shows the i/o layer sheet. Here, you can input new data for the CNN to classify. ExcelNet takes care of the rest and gives you the answer, both in terms of a final probability as well as the most likely answer. Figure 1(b) shows extracting activations. Typically, this requires fussing with file formats, editing `.prototxts`, or writing code. ExcelNet enables you to execute your vision for Deep Learning without learning all this useless nerd information. *You're in control.*

## SELLING POINTS OF ExcelNet

ExcelNet offers the ultimate in Deep Learning abilities. We see a number of unique advantages.

**Proven numerical abstraction layer:** Worried about the correctness of your new app? Excel is the most trusted numerical computing platform: BLAS and ATLAS are used for fluid simulations and other underwater-basket-weaving-like endeavours; Excel is used for finance. By building on a solid foundation, we provide unsurpassed guarantees in terms of algorithmic correctness.

**Enterprise-grade security:** Worried about your autonomous driving platform in ExcelNet getting into the wrong hands when you leave your phone in a bar in SF? Don't sweat it – ExcelNet is compatible with the most secure forms of document protection from Microsoft.

0x47577f4540dfc6dfd45e07fc77b2eb20877ac67318f14865fe7fd01a5c48c7c8

|                    | (a)                    |                    | (b)                    |

**Figure 1.** EXCELNET in action: (a) the interface, where *you* are in control of what goes into the network and where the results can be obtained instantly; (b) examining an activation map. Typically getting access to the power of Deep Learning requires hiring a nerd or learning obscure nerdy things. EXCELNET puts you in control, and lets anyone harness the power of Deep Learning!

**Low cost of entry:** Not willing to invest in pricey GPUs? With a basic laptop, and a copy of LibreOffice, you too can hop on the Deep Learning bandwagon with ExcelNet.

**No programming required:** Afraid of missing out of the building the next big thing in Deep Learning because you do not know how to program? EXCELNET makes it easy by leveraging a basic level knowledge of MS Excel.

**Artisanal data science:** Instead of slurping heaps of data from the Internet, EXCELNET enables you to hand-enter both weights and inputs, just like in the good old days. The resulting numbers are more authentic.

### HOW DOES EXCELNET WORK?
Machine learning practitioners may wonder: how does EX-CELNET work?

It's simple: it's a self-contained .xls file that contains a sheet for I/O, and then sheets for the filters and biases of several layers of a convolutional neural net. The remaining sheets apply mean-subtraction, convolutions, rectified linear units (ReLUs), fully connected layers, and a softmax function. All of these operations can implemented using Excel functions. The I/O sheet then references the final output of the softmax function.

### HOW DO I USE EXCELNET TO CLASSIFY NEW DATA?
Simply enter in your data in the rows and columns of the data-entry matrix, and watch the posterior distribution change in real-time[2]! You may find it helpful to use the multiplier cell at the bottom that multiplies your entered numbers by a fixed

value so you can type in 1's and 0's as opposed to 0's-255's. When you have your prediction – save your `.xls` file, which will save all the activations and the current prediction.

---
[2]At least if you're having a branded experience and using real MsExcel: LibreOffice is definitely slower.

### CONCLUSIONS
Please visit `deepexcel.net` to try EXCELNET today!

### REFERENCES
1. James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU Math Expression Compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*.

2. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

3. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 2001. Gradient-Based Learning Applied to Document Recognition. In *Intelligent Signal Processing*, S. Haykin and B. Kosko (Eds.). IEEE Press, 306–351.

4. A. Vedaldi and K. Lenc. 2015. MatConvNet – Convolutional Neural Networks for MATLAB. In *Proceeding of the ACM Int. Conf. on Multimedia*.

0x9e36bc6e26b97717384e3f53d05a28882f7adcaf3bd914b5b2efe33a19deb59e

# Sniffing for Meaning:
## Defining and maximizing the signal-to-nose ratio

Josh Cox

## Abstract

There are a great many papers and studies which make reference to a signal-to-nose ratio. [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] However, there are very few resources that explain what a signal-to-nose ratio is and how to optimize it. This paper will remedy that by examining references to the signal-to-nose ratio in existing work and exploring methods of maximizing the signal-to-nose ratio, which seems to be a good thing.

## Defining Signal-to-Nose Ratio

Perhaps the closest we have to a concrete definition of the signal-to-nose ratio comes from "Magnetobiology: Underlying Physical Problems" by Vladimir N. Binhi. The book contains the following excerpt:

*"The authors introduce a signal-to-nose ratio R as the ratio of the amplitude of a discrete component to the intensity of the continuous spectrum in a certain narrow frequency interval $\Delta f$ of the discriminator near $f_s$."* [2]

There is much insight to glean from this snippet of text. First, we now know that the signal part of the ratio is a "discrete component". This is disappointing to learn; if it exercises enough discretion, we may never find it. Luckily, we are given crucial information about the nose portion of the ratio. It is "in a certain narrow frequency interval $\Delta f$ of the discriminator near $f_s$" (note: $\Delta f$ is pronounced "nose eff" and $f_s$ is simply pronounced "face"). This is useful information; it indicates that the nose portion of the ratio is related to the nose on

one's face, rather than the nose of an airplane or a rocket.

Next, we examine Cvijetic and Djordjevic's "Advanced Optical Communication Systems and Networks":

*"Generally, the signal and nose levels are defined in different ways in electrical and optical domains. The electrical signals are represented either by current or voltage amplitudes..."* [6]

This tells us that the signals that we are looking for are *electrical signals*, like the ones that come from a mobile phone. This is rather handy because, as we know, signal is measured in "bars" which is a *discrete* measure! It seems that Dr. Binhi was making gratuitous use of wordplay when writing on the subject.
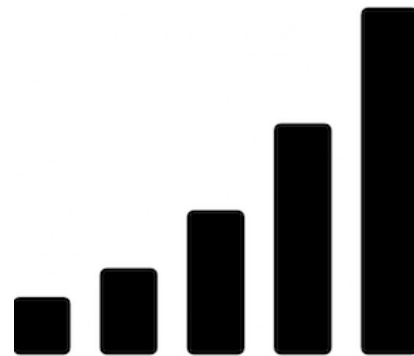


*Figure 1: A typical depiction of a "signal" measurement*

So we've defined our constituent components, signal and nose. The ratio between them is simply the signal divided by the nose. I think we can move to the next section now.

## Maximizing Signal-to-Nose Ratio

But how do we maximize our signal-to-nose ratio? Mathematically it can be done by increasing the number of bars we get (the signal) or by decreasing the nose levels. This

is not a new idea. There is evidence of these kinds of efforts going back to the ancient Egyptians. When constructing the Sphinx, they attempted to maximize their signal-to-nose ratio by doing away with the nose altogether. This was an admirable effort, but the ancient Egyptians' lack of cell phones ultimately capped their signal-to-nose ratio at 0.



*Figure 2: A failed attempt by the ancient Egyptians to maximize their signal-to-nose ratio*

Despite the foibles of the ancient Egyptians, recent efforts to maximize the signal-to-nose ratio have proven quite promising. This excerpt from "Advancements of Medical Electronics: Proceedings of the First International Conference" shows that significant progress has been made:

*"Enhancement in SNR—the ratio between related signal and nose is improved by decreasing other nuisance level and increasing stimulation related signal level"* [9]

Unfortunately, the entire text of the proceedings costs $200 for a digital version, so our shared scientific understanding stops there.

Private companies have also made progress in attempting to maximize the signal to nose ratio. NXP's ultra-low power 8-bit 40 Msps universal ADC0801S040 is advertised as having "Best in class linearity and signal-to-nose ratio". At first glance, it seems like their claim is impossible:



*Figure 3: The NXP ADC0801S040*

When one first looks at the chip, it seems to not have a nose at all, which would make its signal-to-nose ratio undefined regardless of how many bars it has! However, upon closer inspection we observe…



*Figure 4: Close-up of NXP ADC0801S040*

It does have a nose after all! The folks at NXP seem to have discovered a way to manufacture very tiny noses, drastically

increasing the signal-to-nose ratio of their chips.

## Further Work
Because this is the first published work to examine the signal-to-nose ratio in depth, there is much more work to be done. Further research is needed in the affect of increasing signal vs. simply decreasing nose.

## Works Cited

[1] M. G. S. Ali, "Signal Processing of Ultrasonic Data by Frequency Domain Deconvolution," *Walailak Journal of Science and Technology,* 2013.

[2] V. N. Binhi, Magnetobiology: Underlying Physical Problems, Academic Press, 2002.

[3] Central Research Institute for Physics, in *Optics Technology*, 2002, pp. 151-253.

[4] K. G. D. Igor B. Morozov, "Signal-to-noise ratios of teleseismic receiver functions and effectiveness of stacking for their enhancement," *Journal of Geophysical Research: Solid Earth,* 2003.

[5] A. G. B. B. G. C. D. A. B. James M. McFarland, *V1 response variability driven by fixational eye movements,* National Eye Institute.

[6] I. D. Milorad Cvijetic, Advanced Optical Communication Systems and Networks, Artech House, 2013.

[7] B. J. Mohd, "Hierarchical steganography using novel optimum quantization technique," *Signal, Image and Video Processing,* pp. 1029-1040, Nov 2013.

[8] Paul Ramsingh, "September 2010 Case of the Month," 2010. [Online]. Available: http://www.proscan.com/September-2010-Case-of-the-Month-291C1.html.

[9] S. B. K. G. I. S. P. B. Somsubhra Gupta, Advancements of Medical Electronics: Proceedings of the First International Conference, 2015.

[10] E. A. S. B. R. Vinu Das, Signal Processing and Information Technology: First International Joint Conference, Amsterdam: Springer, 2012.

[11] "Why blue-sky thinking gives TOPTICA the power," *OLE,* June 2009.

[12] K. A. Viktor Stein, *Protease-based synthetic sensing and signal amplification,* Proc Natl Acad Sci U S A., 2014.

# SIGBOVIK 2016 Paper Review

## Paper 22: Sniffing for meaning:
## Defining and maximizing the signal-to-nose ratio

**Harry Ne'austreulls**
**Rating: 4 (strong accept)**
**Confidence: 0/0**

As someone who has repeatedly noticed references to the described signal-to-nose ratio but never been able find an authoritative source on the topic, I was overjoyed to encounter your submission. It gives the entire PC an inflated ego to think that SIGBOVIK should be the chosen venue for the canonical paper on any topic, let alone one that practically reeks with demonstrated industry application.

I would, however, like to critique one of your claims: while reading the second section of your paper, I smelled a rat. You claim that the designers of the Sphinx made an attempt at "doing away with the nose altogether," but then go on to argue that their signal-to-nose ratio is 0. The astute reader—me for example—will observe that, as you correctly state in the second column of the same page, any noseless construct will have an undefined ratio. The presence of this correct statement shows you're someone who nose what he's talking about, so I'm willing to dismiss this as a careless error. Fortunately for you, there is some good nose: while I'm not completely certain, I believe I see a small nose remnant on the face in Figure 2, just inferior to the eyes. Confirmation of this spotting may require examination of an SEM, as the paper does with its inclusion of Figure 4 (for which I applaud your diligence).

Overall, nice work, and although I was disheartened to discover your error, I was quite relieved that your claims might still be true, if for the wrong reasons. With that, I'll conclude and signal my approval.

1

0x421b174d4ba0182b89daddc2b5dbda4cc68f3d0fa7024b1323133129a4274468

# Higher-order Research

# On the topic of "um…"

Timothy J Parenti

The space between Carnegie Mellon University and the University of Pittsburgh*

www.timparenti.com

## Abstract

*"Um…", as a speech disfluency, is a linguistic filler, not nominally purposeful or replete with meaning. [2] It is often used when one is found at a loss for things to say and thus, beyond all rational reason, it is somehow the topic of this paper.*

## Meta-Abstract

*In which the author briefly spirals down a rabbit hole in which he never quite envisioned himself, in which he feels a bit out-of-place, but in which he probably does a half-decent job.*

*Probably.*

## 1. Introduction

Why, hello there!

### 1.1. Special thanks

The author wishes to sincerely thank the Organizing Committee of SIGBOVIK 2015 for generously awarding this paper "Best Paper of SIGBOVIK 2016" in their SIG-BOVIK 2016 Best Paper Raffle, as shown in Figure 1. Their positive assessment of my academic and pseudo-academic pursuits is greatly appreciated. I am indeed "buzz'n".

### 1.2. Special apologies

The author wishes to sincerely apologize to the Organizing Committee of SIGBOVIK 2016 for what the Organizing Committee of SIGBOVIK 2015 did, presumably without their consultation.[1] He didn't ask for it. (Seriously, they just picked a random number and pointed at him.)

The author humbly points out that this "Best Paper" award need not be to the exclusion of other similarly- or even identically-named awards. In fact, it probably shouldn't be, since there are undoubtedly better papers in the surrounding proceedings. [1]

---

*So…I guess, somewhere around Craig Street? Panther Hollow?

[1]Unless they were somehow capable of time travel, in which case — *bravo!*



Figure 1. The lovely certificate with which the Organizing Committee of SIGBOVIK 2015 simultaneously blessed and cursed me. The parts about "acausality" and "luck" were particularly heartfelt and meaningful.

### 1.3. Special anxieties

The author, upon receiving this prestigious(?) award — not knowing whether he was capable of being concurrently serious and funny, deliberate yet whimsical, sober and…um…*not* — responded to the near-immediate inquiries regarding the topic of this paper simply with "um…".

Perhaps relatedly, the author continues to find himself in an uncomfortable position of expectation. But not really. (This is SIGBOVIK, after all.)

But kinda? (This is SIGBOVIK, after all.)

## 2. Coming up empty

The author concedes that this paper contains virtually nothing of any real or imaginary value to anyone. The author further concedes that he is enjoying this nevertheless. The author further concedes that the previous concession wasn't really something worth conceding. The author further concedes that perhaps he has conceded too much, and that, even further, perhaps he should not proceed any further with this particular joke, and thus dispense with laboriously referring to himself in the third person.

1

## 2.1. On meeting expectations

After hearing the news, like any Good Academic™, I immediately made note of the deadline, which was then $11\frac{1}{2}$ months away, before promptly putting it out-of-mind and ultimately hastily slapping this together during the four hours immediately preceding the final conference submission deadline. (See Section 1.2.) I have to keep up appearances, right?

In my defense, I have been busy in the interim with "real life", whatever that may mean on any given day.[2] I like to think I've been doing that pretty well lately, but for better or for worse, it has obviously somewhat hampered my ability to produce Great Academic Literature such as this, leaving me tempted to just fill the whole thing with \ldots or \lipsum and go to bed. (Although arguably, what I'm doing here isn't much better. It is more fun, though.)

That said, if you look past the frenetic self-imposed time crunch and instead count a year's worth of shower thoughts, the occasional friendly conversation, and idle daydreaming, then this could actually be considered my preëminent work of the year.[3] Maybe not the most *comprehensive* document of my body of work, but preëminent, sure. (Even if only because it is first.)

Now that's being a Good Academic™.

## 2.2. On not being a disappointment

Just over half an hour after my sharing the news on Facebook,[4] one friend reached out and commented:

> I eagerly await your paper.

The pressure thus properly established, he continued to remind me of this looming deadline at irregular intervals, concluding with a remark made no more than six hours before the submission deadline, that yes, the deadline was in fact tonight, and "remember, it's gotta be the Best Paper!" (The author concedes that this particular incident may just be at the front of his mind because it was literally earlier today.)

A few internalized expletives later, though I knew this already, I undertook the necessary preparations, because even though I've surely been a disappointment before, I'm not going to be a disappointment this time.

And then there's my mother,[5] who I know will at least *try* to be proud of this, even though she probably won't quite understand what this is until I've explained it to her. She'll then pause here and ponder this a bit and wonder briefly how much time I spent on this that could have been used on other endeavors, and I will refer her, of course, to Section 2.1.

Of course, self-esteem plays its part, too. Least of all possible disappointments, I certainly don't want to be a disappointment to myself. I had a decent number of chuckles while thinking about this paper over the last year. Some of the well-crafted jokes I thought out have been incorporated here. The rest, I either decided weren't worth the effort, or I forgot about them completely in my mad race to finish this on-deadline, *which I'm sure I won't be thinking about right after I click "Submit"*.

However, since my tendencies toward perfectionism would otherwise preclude me from submitting this without making this the best "Best Paper" I can, they are nonetheless left as an exercise to the reader.[6]

But really that's kind of the beauty of the whole thing, since it's kind of meant to be half-assed anyway. And can I really be proud of that?

(Yes, I can be.)

## 3. Conclusion

Um. . . I got nothin'.[7]

## References

[1] *Proceedings of SIGBOVIK 2016*, 1 April 2016. Also known as that thing you're reading right now.

[2] Wikipedia. Speech disfluency, accessed 15 March 2016, 22:09 EDT, less than two hours before deadline. https://en.wikipedia.org/wiki/Speech_disfluency.

---

[2] After all, I am "buzz'n".

[3] Yeah, I just used a diæresis. I happen to like typographical stuff like that, and LaTeX just makes it look *so good*. . .

[4] I got excited, alright?

[5] Hi, Mom!

[6] I've always wanted to do that.

[7] Seriously, what did you expect?

# SIGBOVIK 2007 Paper Review

## Paper 34: A New Historical Analysis of /ʌ/

**Reviewer Two, ACH Committee on Retrocausal Publication Impact**
**Rating: Reject**
**Confidence: Expert**

This paper presents a so-called "new" analysis of /ʌ/. Unfortunately, the claimed contribution is not actually a contribution, as the equivalent subject matter of "um" was already thoroughly studied by (Parenti 2016). In the extremely unlikely event that I am mistaken, the authors should clearly elaborate what distinguishes their paper from the existing future work. The submission would be improved by rewriting it with a retrocausal reference frame, especially to include the citation recommended above.

0x2ff2623e20cbbc09f14138566df9f3eb9991c6381b0f1dcef0b3ee94d7980b81

0x09a7ae5c6c768979e774ac2731bfd51227176b4916f29e655fee2177b2529d34

# SMACK: when research in algorithms goes around in circles

Patrick Lin[*]        Weishuang Linda Xu[†]

**Abstract**

We take a particle physicist's approach to the design and analysis of algorithms. We propose a theory of algorithmic interaction and present SMACK, a system for high-energy collision of algorithms using a national-scale accelerator.

## 1    Introduction

It is well known that researchers in algorithms often write papers by taking two absurdly complicated algorithms that no one will implement for two problems that no one cares about, and then combine them into an even more absurdly complicated algorithm for yet another problem that no one cares about.

In this paper we devise a Theory of Algorithmic Interaction in an attempt to understand and model this process, and using that, describe a system (SMACK) whose high-luminosity collision events will facilitate the search for new algorithms.

## 2    Theory of Algorithmic Interaction

We postulate the existence of a Standard Model of Fundamental Algorithms, whose composite states make up the rich zoo of algorithmic particles observable today. These fundamental building blocks, called Quirks, exist in many flavors. Many of these are familiar to those who have taken an undergraduate-level course in algorithms. Basic examples include Brute-Force and Divide-and-Conquer. Some composite particles familiar to most include Dynamic Programming and Linear Programming. We propose that almost all algorithms can be understood as the bound state of several quirks, and we have moreover found that the size of the composite algorithm appears to be unbounded, as they themselves are allowed to bind and form a larger algorithm. This proposition is shown to be true by choosing three algorithms at random[1] and verifying for these individual cases.

---

[*]Department of Computer Science, University of Illinois at Urbana-Champaign
[†]Department of Physics, Harvard University
[1]You will never be able to prove that the randomness was neither uniform nor independent.

0x14cb211f2e5e937d890fcf93241f0a09313793f922f4bb60d3a62f29c7b95a92

- In 1996, Chan [2] combined the Convex Hull algorithms of Preparata and Hong [9] and Kirkpatrick and Seidel [8] into a new, faster algorithm for Convex Hull. We note that all of these algorithms are, in effect, Divide-and-Conquer algorithms. This shows that interaction between nontrivial instances of Divide-and-Conquer can produce even more complicated Divide-and-Conquer algorithms.

- More recently, Chekuri and Quanrud [3] combined an algorithm of Cunningham for Matroid Intersection [4] and an algorithm by Duan and Pettie for Maximum Weight Matching [5] into an algorithm for Maximum Weight Matroid Intersection. Cunningham's algorithm is a variant on Augmenting Paths; Duan and Pettie's work is an example of the technique of Scaling.

- The previously mentioned paper is an improvement on an earlier algorithm by Huang et al. [7] for the same problem, which instead combines Cunningham's algorithm with a technique of Frank [6], which also employs Scaling.

Under our theory, when two algorithms interact and form a bound state, a new and more efficient algorithm is created and energy is released in the form of computing resources. The interaction is mediated by the Big-Oh Obliteration gauge boson and appears to be coupled to sources of coffee, graduate students, and glaring advisors–though the precise coupling mechanism remains unknown. The Lagrangian for this theory, which determines the dynamics of algorithmic interaction is given by

$$\mathcal{L}_{\text{algo}} = \bar{\phi}_a(i(\gamma^\mu D_\mu)_{ab} - \delta_{ab}m)\phi_b - B^a_{\mu\nu}B^{\mu\nu}_a \tag{1}$$

Where $\phi$ is the field, a function of Space-Time [complexity], generated by the algorithm. $\gamma^\mu$ and $D_\mu$ are the spinor representation and covariant derivative, respectively, derived from the metric generated by Minkowski PSPACE-PTIME. $m$ is a parameter denoting how confusing the interacting algorithms are, and $B^a_{\mu\nu}$ is the field strength tensor of the Big-Oh Obliteration field.

The computing energy produced from the creation of the daughter algorithm is typically radiated in quantized factors of $\log\log(n)$. In general, the daughter algorithm is unphysical and has a very short lifetime, typically decaying into a conference publication particle and (one hopes) a less efficient but actually implementable derivative, which then go their separate ways and never interact again. In most instances, composite algorithms have been observed to decay into only conference publications with no industry implementation particles to be found; this is known as the Missing Application Problem. Inspired by Wolfgang Pauli's proposal of the electron neutrino in 1930 [1], we speculate that industry implementations are produced but are too useless to be detected.

We further note that the Standard Model as it stands currently is incomplete, as there are many algorithms where no one appears to understand how they work.
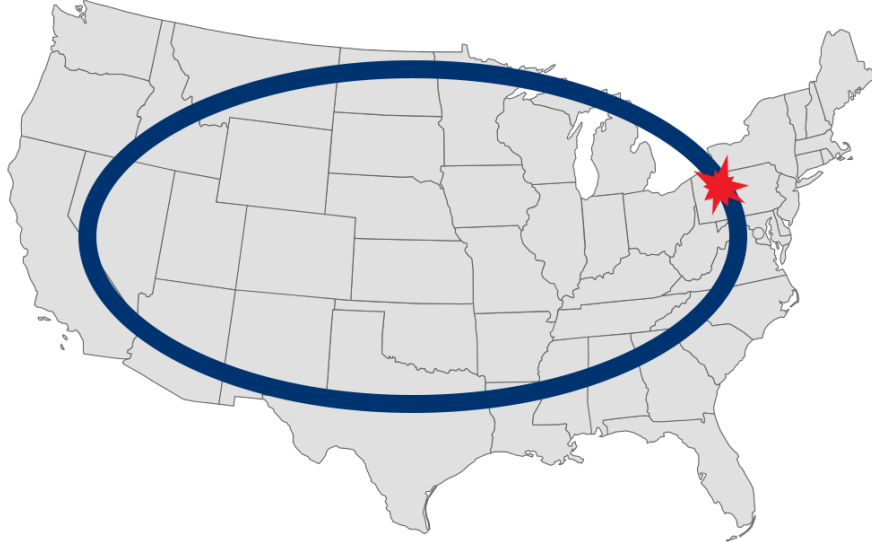
0xa73cc673abc31468a4b22ab09cbe0eb90d7686081fd5435e00a0b1440a91dc97

Figure 1: A detailed map of the SMACK apparatus[4], with a final collision site located in the vicinity of Carnegie Mellon University[5].

## 3    The SMACK system

To study and optimize the algorithmic interaction process, we have designed a high-luminosity algorithmic collider to maximize the production rate of new algorithms in the field and also to further the understanding of algorithmic interaction theory. The collider is split into two main components: the acceleration apparatus which prepares algorithm beams for interaction, and detectors at the interaction site.

The Super Magnificent Algorithm Collider Kontraption (SMACK) accelerates and refines two beams of algorithmic particles by injecting them into a circular track, where they complete three orbits before they are allowed to intersect. Figure 1 shows a map of the collider. Each beam passes through several acceleration cavities located at major research universities across the country, where on its first orbit it is allowed to interact with graduate students, on the second with postdocs, and on the third with research faculty. During each of these interactions, the algorithm beam is purified and excess factors of $\log\log(n)$ are radiated off along with caffeine and grant money.

We observed that this interaction sometimes generates Thesis or Grant Proposal particles, but they are kept separate from the main beam and not included in the analysis. We will refer to this circular track and its acceleration apparatus as the Tenure Track, due to its excessive length and tendency to push its

---

[4]Map of US from `https://commons.wikimedia.org/wiki/File:Blank_US_Map.svg`

[5]We reserve credit for the next algorithms paper that comes out of there.

3

contents to extreme standards of speed and efficiency.

The two algorithm beams interact at the collision site via the Big-Oh Obliteration Mechanism (BOOM) described above, mediated by the corresponding Big-Oh Obliterating Boson (BOOB). At the site of interaction, two detectors, FAME and FORTUNE, are placed to collect and analyze data on the algorithms produced by the collision.

The FAME detector, containing ultra-high precision citation calorimeters, is used to collect information on citation potential of the new algorithms once publication decay particles are incident upon it. Due to the high luminosity of this collider, unmanageably large amounts of data are generated by this experiment, and the detector employs a trigger system to filter out third-tier conference material. For collision events that pass the trigger threshold, the FAME detector computes its coupling strength to major research journals and automatically allocates grunt work to facilitate the paper-writing process.

The FORTUNE detector is optimized to detect potential for money of the collision events, and is split into industry funding and grant funding sectors. Since this experiment has reported significantly fewer events, a trigger system is not needed. As soon as FORTUNE the detector records a collision event with the remote hope of an industry application, the coupling strengths between the algorithm and major tech corporations are computed. The results are then sent to all companies with a non-zero coupling as well as government agencies along with a desperate plea for funding.

## 4   Experiments and Results

As a preliminary test, we injected Recursive Backtracking and Memoization into a miniature low-energy version of the collider, which simply made a few turns through the nearest undergraduate population. After a few trial runs, the miniature collider produced the composite Dynamic Programming along with a few auxiliary Extra-Credit particles, as expected.

For a full trial run, we took the algorithms from past proceedings of various Theoretical Computer Science conferences and injected them into SMACK. Measurements at the FAME detector showed high-energy interactions in various locations, including Carnegie Mellon University, the University of Texas at Austin, and the University of Waterloo. Unfortunately, we have not yet been able to access the produced algorithms. With any luck, the decayed publications will appear relatively soon on ArXiv[6] or in publication[7]. Furthermore, the FORTUNE detector has yet to record any events generated from this experiment.

---

[6] At which point we will be able to claim proper credit for them.

[7] In which case we will still try.

# 5   Conclusions and Future Work

We proposed a Theory of Algorithmic Interaction in the way of a Standard Model of Fundamental Algorithms, and gave experimental evidence to its existence. Of utmost importance is finding more experimental evidence, as well as deriving theoretical predictions based on this model.

As mentioned in the Theory section, the Missing Application Problem and the search for Useless Implementations (UI) of composite algorithms represents for Algorithmic Interaction Theory a large patch of uncharted waters, and more and more researchers have moved to the rich and exciting field of these Hopeless and Counterproductive Implementations (HCI).

As far as we can tell, SMACK works as designed. Unfortunately, retrieving the produced algorithms in time to take credit for their creation remains a problem. In general, finding new ways to optimize SMACK would also allow for a greater volume of algorithms to be produced. For the moment, we welcome and encourage research institutions around the nation to voluntarily credit us with all the algorithmic results they have produced within the past decade.

# References

[1] L. M. Brown. The idea of the neutrino. *Physics Today*, 31(9): 23–8 (1978).

[2] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4): 361–368, 1996.

[3] C. Chekuri and K. Quanrud. A fast approximation for maximum weight matroid intersection. *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 445–457, 2016.

[4] W. H. Cunningham. Improved bounds for matroid partition and intersection algorithms. *SIAM Journal on Computing*, 15(4): 948–957, 1986.

[5] R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. *Journal of the Association for Computing Machinery*, 61(1), 2014.

[6] A. Frank. A weighted matroid intersection algorithm. *Journal of Algorithms*, 2:328—336, 1981.

[7] C. Huang, N. Kakimura, and N. Kamiyama. Exact and approximation algorithms for weighted matroid intersection. *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 430–444, 2016.

[8] D. G. Kirkpatrick, R. Seidel. The ultimate planar convex hull algorithm. *SIAM Journal on Computing*, 15(1): 287–0299, 1986.

[9] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, vol. 20, 87–93, 1977.

0x17f7a6b9e8f4e384375e3416f08bec5bef2a63a86120405e1154e47da5549339

**Stefan Muller, Carnegie Mellon University**
**Rating:** $1.2 \times 10^{18}$
**Confidence:** $\epsilon$

I am intrigued by the authors' proposal for the construction of a Super Magnificent Algorithm Collider Kontraption. This does show some promise in automating the standard methods for discovering new algorithms (realize the paper is due soon and throw a bunch of stuff together really quickly). However, I am forced to note several deficiencies in the proposal:

1. The device appears to be single-use. As is well known, once a researcher's body of work has accumulated its peak momentum and reached the end of the Tenure Track, the velocity and density of ideas produced drop off and are unlikely to again produce the high-pressure conditions necessary for a collision.

2. The cost of the project appears to be prohibitive. That's something I should be concerned with, right?

Until such time as funding for such an ambitious project becomes available, perhaps the authors could study the quantum-mechanical processes by which, out of the vacuum of a graduate student's frenzied, underslept, overcaffeinated mental state, pairs of algorithms and anti-algorithms (counterexamples) spontaneously arise before quickly annihilating each other.

# A Look into the Mind of the Modern Graduate Student via Telephonic Obfuscation

**Erik Harpstead, Sauvik Das, Dan Tasse, Rebecca Gulotta, Felicia Ng, Siyan Zhao, Robert Xiao, Jennifer Olsen, Judith Odili Uchidiuno, Anhong Guo, Xiang 'Anthony' Chen, Adam Stankiewicz, Nick Diana, Anna Kasunic, Caitlin Tenison, Kenneth Holstein, Julian Ramos Rojas, Michael Madaio, Michael Rivera, Nesra Yannier, Adrian deFreitas, Qian Yang, Nikola Banovic, Mary Beth Kery, Steven Dang, Xu Wang, Fannie Liu, Kristin Williams, Brandon Taylor, Samantha Finkelstein, Alexandra To, David Gerritsen, Toby Li, Cole Gleason, Christopher MacLellan, Franceska Xhakaj, Alex Sciuto, Judeth Oden Choi, Fanglin Chen, Jeff Rzeszotarski, Rushil Khurana, Joseph Seering, Kerry Chang, Gierad Laput**

Human-Computer Indoctrination Institute
Carnegie Mellon University
Pittsburgh, PA 15232

## ABSTRACT

The modern graduate student has many diverse concerns on their mind including such highlights as: sustenance, self-image, popular culture, ideological proselytization, and the existential dread of contemplating the relativistic nature of truth. We, the members of the ACH special interest group on Computer Human Indoctrination, were interested in better understanding the vacuous depths of the graduate mind. We ~~subjected~~ introduced a population of graduates to a method first proposed by Borel (1913)[1] and augmented it with a paradigm of telephonic obfuscation [0]. What follows is a scholarly look into the nuanced minds of our ~~puppets~~ colleagues.

## Author Keywords

Comestibility theory, telephonic obfuscation, computer human indoctrination, human-robot-overlord interaction

## INTRODUCTION

Sandwiches are a popular food throughout the Western world [1]. Since John Montagu, 4th Earl of Sandwich, declared such bread-foods "sandwiches", they have become Britain's greatest contribution to gastronomy [2]. However, recently, a wide variety of bread-foods have sprung up [3] without appropriate insight from the academic community. Specifically, we have heretofore been lacking a comprehensive ontology of sandwiches. This presents numerous problems: non-Western sandwich equivalents have become marginalized by the Anglicization of the bread-food world, diners may be intimidated by their lunch's lack of grounding in relevant theory, and innovative bread-food chefs are left without a theoretical framework within which to frame their new inventions. Therefore, in the spirit of cultural appreciation and advancing the state of the art in sandwich theory, we propose in this paper a new sandwich framework.

This framework would be incomplete, however, without a higher-level discussion of the role of frameworks in theory building and the dissemination of scientific knowledge. Here, we posit answers to the questions: 1) Given the relative dearth of research in this area, how can we contextualize the work we've done? and 2) How might we publicize our framework to ensure that our work is communicated to interested parties? Regarding the former question, though there are few extant examples of relevant sandwich-related frameworks, we argue that there is an opportunity to build on work from related disciplines including bread studies and, controversy aside, recent work that examines the importance of the distribution of fillings in all foods. Regarding the latter question, here too we can draw from existing practices that shape how people engage in discourse around how they can create, consume, and obtain food in their daily lives [4, 5, 6].

Over the past century, bread researchers who specialize in the systematic investigation of sandwich-related ingredients have developed, through rigorous quantitative and qualitative studies in both laboratory and field settings, a detailed taxonomy of the people who derive extraordinary levels of physical and emotional pleasure from inserting comestible fillings between carbohydrate slices [7]. This taxonomy has been widely accepted by the academic community as the preeminent foundation for understanding personal, social, cognitive, and emotional motivations behind sandwich making behavior. In addition, it has been popularized in American media by renowned director A.J. McMuffin in the Academy Award-winning film, Searching for Pan Dulce. As such, the broad-reaching results and impacts of bread research give us a rich space with which to contextualize our work on sandwich-related frameworks.

Recently, more diverse ingredients are introduced to replace bread buns. Other forms of carbohydrates, such as waffles, tortillas, and pizza slices are used to wrap delicious content between them [8]. In 2005, McDonalds in Asia offered

---

[1] Adapted for use with a finite set of monkeys[2]
[2] Graduate Students

compressed rice cakes in place of bread buns, with mixed feedback from the consumers [9]. Beyond carbohydrates, people have experimented with non-grain buns. Vegetables, such as cucumbers, sweet potatoes, and tomatoes, have all been used to replace bread [10]. Nuts are also a popular substitute. Despite of the variety of buns used in sandwich-making, most current sandwiches still follow the traditional framework of separated and layered content, which only delivers limited amount of flavors and often makes it difficult for consumers to sample all the layers in one bite. This was addressed in the sandwich-related framework introduced in this paper.

Many alternatives to the venerable sandwich model also exist, with their own advantages and disadvantages. Of particular relevance is the burrito, a category of food legally distinct from sandwiches [A]. Burritos solve the layer separation problem by thoroughly mixing layers of ingredients within the tortilla shell, enabling consumers to sample all layers in a single bite [B]. Although popular, the mixed-layer approach of burritos is only applicable to a narrow range of possible ingredients, as evidenced by the use of the same basic formula (rice, beans, meat, salsa) across most burrito implementations [C]. In this work, we solve the layer-separation problem with our sandwich construction framework for more general ingredients, giving our approach wider applicability and, it is hoped, greater mass appeal.

Additionally, with bread being a main component of the sandwich model [1], the model does not encompass gluten free options without making specialized instances of bread. Similar in structure to the burrito, the lettuce wrap attempts to address the gluten free problem through the use of leafy greens. In addition, the lettuce wrap has a surprising benefit of being a low calorie option for the consumer [2]. Within the lettuce wrap, the ingredients are often the same as those that you could find within a sandwich, which could give it the wide appeal of the sandwich. However, the lettuce wrap is not widely used, perhaps because consumers have an aversion to vegetables. In our sandwich construction framework, we aim to include the benefits of the lettuce wrap while avoiding the pitfalls of vegetables.

Rudi's gluten-free spinach tortilla wrap is a product that gets very little attention from the media, and therefore remains relatively unknown by the general public. They are "soft and pliable tortillas that are made with simple and wholesome ingredients, including whole grains and 5 grams of fiber per serving. These tortillas are ready to be rolled into a delicious burrito or a tasty sandwich wrap" [1]. They are not only gluten-free, but also dairy-free, which can prevent discomfort for people who have problems digesting the natural sugars found in milk - lactose [2]. Unlike lettuce, it can be stored for months in the freezer, which makes it a cost-effective option for families who save money by using coupons or buy in bulk. It is nutritionally superior to lettuce as it is made with spinach powder, which

"has twice as much potassium, protein, calcium, iron, niacin and vitamins A, C, B, C and B-12 as any other leaf vegetable. Spinach also contains more fiber and minerals including magnesium, phosphorus and potassium than any of the four lettuce types." [3]. Finally, its high fiber content (20% of the daily fiber requirement) is much more than the amount found in most breads and wraps used in sandwich or burrito making.

An eater from Vancouver reviewed the spinach tortilla wrap like this: "*Absolutely delicious. Every since I removed wheat from my diet I have been in search of delicious tortilla's and roti's. Rudi's hits all the highs. The millet add's notes of rustic indian flavours while it even puffs up a bit when you heat it like a roti! [4]*" Not only the wraps themselves are gluten-free and diary-free, the package of the Rudi's gluten-free spinach tortilla wrap are also made in the most environmental friendly way possible. In fact, they are Mercury-free, Arsenic-free, BFR free, Beryllium free, PVC-free, and of course, made with high recyclable materials [5]. All these considerately designed and implemented features make the product the next big thing in the wrap industry.

**IMPLEMENTATION**
Specifically, we built a Fused Deposition Modeling machine [ref] (more commonly known as a 3D printer) with custom ingredient extruder to produce the tortilla wrap. A wrap is made in three steps. First, printing the base wrap - preprocessed corn flour paste is extruded through a syringe-like mechanism with 3mm extrusion diameter and 100% infill to produce a circular and solid wrap. Then the printbed is heated to 350 degree - that is, to bake the wrap. Next, a repurposed blender brings in the aforementioned Mercury-free, Arsenic-free, BFR free, Beryllium free, PVC-free stuffings. The order and amount of each ingredient is randomized, fitting a Gaussian distribution. Finally, to actually stuff the wrap, the machine performs wrapping of the wrap. Our main contribution is a mixed-initiative approach. A video of the pre-wrapped wrap is broadcast in the public area of our institute, attracting hungry members who are shown directions to our printer. Upon arrival, in order to eat the wrap, they have to wrap it, thus finishing the final step.

Our Fused Deposition Modeling machine builds on commonly used approaches for rapid prototyping, specifically with regard to designing scaffolds using a starch-based polymer (Lam et al). Prior studies have shown that such starch-based polymers have proven to absorb water consistently using interconnected porous networks and repeatable 3D geometry (Lam et al). We expect similar results when applied to our custom ingredient extruder to produce tortilla wraps that will be able to withhold any substance (e.g., beans, salsa, etc.) that the hungry participants may place inside of the tortilla wrap.

In practice we found that our custom ingredient extruder easily exceeded these traditional benchmarks, and as a

result we revised our original test metrics to include a more diverse set of practical applications. In addition to the classic measures often cited in fields with a strong tortilla foundation (circularity, flatness), we added several measures (tensile strength, thermal instability, stopping power, and leadership qualities) to test the limits of our new extrusions. While we expect that differences on more traditional measures will make up the meat of our findings, a specific characterization of new extrusions is only possible by bringing new, non-traditional measures into the fold.

But then we said to ourselves: why tortillas? Are more circular, flat tortillas really going to make the world a better place? Is there really a problem here? We drew upon theory from design and/or research through design [R1], and got NSF to give us a sizable stack of money. Then we recruited people on Craigslist by offering them pennies. We asked them a series of questions about their favorite hairstyles. The research team video-recorded these interviews, but then we accidentally deleted the files, so we transcribed The HuggaBunch [R2] instead. Through a series of rigorous analyses and vigorous head nods, we found that the future we want to create is a world in which marshmallows from Lucky Charms are separate from the whole grain cereal. Further investigation, however, revealed that you can already buy bags of pre-separated charms on internet websites, e.g. [R3]. Thus, through the support of another grant [R4], we asked the question: what's the purpose of HCI and of doing anything in life? We ran a series of workshops and pilot studies investigating this question, and present the results in figure A.



**Figure A.**

Although some reviewers (i.e. Reviewer 3, a.k.a. Capitan Ass-hat) might argue that the implementation of this research may seem scattered and half-hazard, we argue that the process of (1) considering the deeper meaning of our hypothesis, (2) collecting data on a tangential topic, (3) analyzing a completely different dataset and (4) developing new, unrelated hypothesis, all while being generously funded by the National Science Foundation, represents the true, unadulterated scientific process[2]. Through this organic implementation of our science, we uncovered one critical question in the field: what is the purpose of HCI and doing anything in life? Our initial work running workshops and pilot studies gave us many answers (see Figure A), however these answers proved nearly incomprehensible. We needed a better way of assessing this question. This lead us to develop a multiple-choice survey which we could easily distribute on Amazon Mechanical Turk. In our next section, we provide a description of the questions we developed, and the logic (and counter-logic) behind the multiple choice options provided to users.

## SURVEY DEVELOPMENT AND VALIDATION

To evaluate the comprehensibility of our initial multiple choice survey, we first conducted a small pilot study. Since we ultimately wish to collect survey responses via Amazon's Mechanical Turk (MTurk), our pilot participants were a diverse sample of 17 MTurk users from over 2 countries. All pilot participants were flown in to our laboratory, where they were individually instrumented with an accelerometer, an electroencephalogram (EEG) headset, an automated blood pressure (BP) cuff, and a head-mounted eye-tracker while they completed a brief (3 question) multiple choice survey on a computerized interface. The results of this pilot suggested that very few MTurk participants are familiar with the acronym "HCI", and that far fewer are familiar with the discipline of "Human-Computer Interaction". Unfortunately, an answer to the first half of our research question, "What is the purpose of HCI?", is likely to be incomprehensible to participants who are not already familiar with the field. To provide MTurk participants with the required level of familiarity, we developed a novel kind of multiple choice survey which initially functions primarily as a text-based role-playing game (RPG) -- designed to simulate the first three years of life as a PhD student in HCI -- before eventually fading into a traditional survey. Through the use of adaptive hypermedia, this "cognitive survey" creates a hyper-personalized graduate school experience for each MTurk participant -- terminating in a 3 question survey only once the system is confident that a participant has acquired a deep and holistic understanding of the field.

After this initial pilot study "exploration", we concluded that people just do not get HCI and that HCI alone may be better without people so in this paper we propose to take the Human out of HCI and define a new field called simply Computer Interaction. This is our main contribution.

---

[2] We would like to point out that this complete and honest description of our implementation shines as a beacon of true science in a field contaminated by the inconsequential garbage some researchers insist on dumping on us (e.g. [R1]).

To discover how computers interact with each other, we naturally surveyed 200 computers, with a set of survey items developed from the Phillip K. Dick model for qualitative android research [cite]. The survey items were chosen to interrogate the communication and mating habits of the North American personal computer. In a preliminary pilot study, we attempted to interview the computers, but this yielded few usable results.

All interviewees were required to disable connections to the outside world including Bluetooth, Wi-Fi, and ethernet ports. However in multiple instances, a computer modulated a binary signal onto its central processing unit's data bus. This in turn generated a Radio Frequency (RF) signal that twiddled the bits of neighboring computers. To say the least, the neighbors were not pleased to have their private ports accessed. We launched a second pilot study and included an RF jammer to address this difficulty.

"In our pilot study participants used a smart home system to control their home alarm from their smart phone. The users were asked to set their home alarm from their phone as they left home in the morning. Then, when they arrived home in the evening, they were asked to turn off the alarm from their phone. However, from our surveys, we realized that many of the users forgot to turn off the alarm when they came home, and thus when they opened a window later on, the system thought that there was a thief in the house. Therefore we decided to add a reminder/beeping system where the phone would remind users to turn off the alarm when they got home."

Yet even after experimenting with multiple notification strategies, we found that users were still unable to reliably turn off their home alarm using their smartphone. This led us to abandon the idea of using manual input altogether in favor of a more automated approach. When users left their houses in the morning, their phone (running a custom app) would automatically determine when the user was more than 100 feet from the house, and activate the alarm. The system would then turn off the alarm when the user returned in the evening. From a power drain perspective, this approach was slightly inefficient, as the user's phone has to continually poll the GPS sensor to determine when the user is home. In practice, however, we found that this approach was the only way that we could reliably make sure that the alarm was correctly set.

Overall our survey surfaced much instability in users' manual control of the home alarm. The results forced us to automate the triggering mechanism of the system, yet at the same time to ensure users' situation awareness as well as their sense of control. Trading off these two important facets we designed a mix-initiative home alarm control system using Android application. The system design extends previous smart-home research [1] [2] in which automation dominants.

**SYSTEM DESIGN**

In this section, we present the design of a mixed initiative home alarm system with automatic triggering. Following the findings from our formative study, we first define the design space for mixed-initiative home alarm systems. We then explore this design space with a number of prototype alarm systems. Using usage scenarios, we explain how each of our prototypes addresses the gaps with the current systems we have identified in the previous section. In particular, we show that our final mixed initiative home alarm system prototype could address our design goals to increases the users' awareness of the system status and their sense of control when using the system.

What is a home alarm system? Does it exclude burglars from entry? Does it call the police when someone enters your home while you're away? Users in our formative study expressed reasons beyond security why they wished to be alerted when a person entered their home. One user wanted to be alerted when their outdoor cat returned, so that they could more effectively hug it. Another user expressed interest in knowing when their spouse entered the home, or when they themselves did, to track schedules. These responses prompted us to reframe the design goals of a "home alarm system". We broadened the meaning of this device from "alarm" (igniting fear, security, police) to "notice". In a home notice system, the user interacts with a mixed-initiative system, an AI-agent instrumented with sensors and mechanisms at each point of entry, eg. door, window, cat-flap. The homeowner controls how the house reacts to different "kinds" of people attempting entry [figure x]. For example, a door may both jovially welcome you home from work, and call the police on an intruder. What would a home notice system look like that was nurturing? What would a home notice system look like that was communist? What would a home notice system look like reflecting the philosophical beliefs of a single homeowner? We explored this through a series of prototypes.

**figure x.**

The first prototype of the home notice system(HNS) explored the interactions between millennial american values and the home. We wanted to reflect the hyperconnected values of millennials while also paying tribute to their defiant stances towards institutions. Upon entering through the front door, we greet the user with a status update of everyone already in the home as well as any recent updates from friends drawn from a preselected inner circle list. While in the home, the system monitors social media networks for updates about friend activities, primarily looking for indicators of novel activities such as craft beard trimming pottery fairs, nearby campus safe-zone protests, and post-post-post modern latte art exhibits. The system monitors the user's activities throughout the home and proposes activities based on preferences learned by applying deep learning methods to past activity choices. The system further notices patterns in the user's daily habits and promptly notifies the user of their intellectual malaise upon initiating a heavily repeated routine, discouraging the development of institutions of habit. This first prototype HNS was designed not only to gauge the interaction of millennials with a home suited for their lifestyles, but also the broader social impacts of millennial values across social groups in America.

To further illustrate how the home notice system (HNS) works on monitoring millennial americans' activities at home and provide notifications, we'll give several use cases here. The main goal of the system is to help users achieve a healthier personal life and social life. From the perspective of personal life, the system will be able to monitor users' daily activities, and provide healthy and intelligent suggestions to users. Here are two examples regarding daily needs. 1) The system will be able to monitor user's eating activities in the past few days, and suggest future meal ingredients which could provide more balanced nutrients to the user. Upon the user's request, the system will also be able to order the food to be delivered to the door. 2) The system will be able to extract the user's schedule from his calendar, and suggest outfit of the day, based on events of the day, temperature, color match, etc. The system could also suggest clothes for the users to buy based on the user's closet. The system could rate the user's satisfaction to each item, no matter it's a meal, a book, an activity, an outfit based on the user's conversation correspondent to the item, so that the system will automatically learn the user's likes and dislikes. This will enable the system to provide a variety of recommendations to the user, which serves our research purpose of understanding millennial american values. From the perspective of social life, users can choose to share personal data with their friends. Users will then know how their friends respond to a movie, a book, an activity, a coat, etc. And these evaluations are natural, based on users' daily conversation. This will provide more life options for users, which also serves our research goal to investigate millennial americans' interaction with each

other. The system will also be able to analyze conversation happening in the home to predict relationship between family members, and suggest movements to different members to establish better relationships between them. This is important for analysis of interaction between millennials happening at home, and the way they dealt with it. It will provide valuable data for us to investigate millennials' values towards family relationships.

Figure X demonstrates example feedback that the home notice system (HNS) can give for these different use cases, displayed on screens installed throughout a user's home. By delivering notifications and recommendations about a user's daily activities, interests, and relationships, the system will ultimately help millennial Americans achieve a healthier lifestyle and stronger social connections.



**Figure X. The HNS offers a) feedback about a user's eating activities b) suggested outfit according to the weather c) suggestion for establishing better relationships with a user's friends and family.**

## SYSTEM EVALUATION

Since the goals of HNS is an ideal--helping American millennials achieve a healthier lifestyle and foster stronger social connections--we decided to evaluate the system's design through counterfactual analysis. Were we to directly evaluate the HNS according to whether it successfully realized its goals, we would inevitably infuse evaluation of the technology with our own values of what lifestyle should predominate or what the appropriate distance of a social connection should be. So rather than infuse the HNS with those values and then judge them against a like-minded metric, we developed an evaluation methodology that judges the HNS against an opposing metric. First, we created evaluative dimensions examining whether the HNS encouraged unhealthy lifestyles or ruptured existing social connections. To examine these dimensions, we used the sensing capabilities of the HNS to collect copious amounts of data about American millennials within their home and without their prior knowledge and then coded the data

according to a code set we derived from prior episodes of MTV's Real World [1]. Codes included things like "loyalty and betrayal", "diet", or "recreational drug use". Finally, we conducted semi-structured interviews with participants where we first asked them about their use of the HNS, and then we exposed 3 episodes of their collected data and the codes we assigned to it to elicit their feedback on the HNS.

For each participant, the interview structure was dictated by the relative frequency of coded terms derived from their individual HNS data. To further reduce the risk of biasing the participants' responses with prescriptive notions of constitutes healthy lifestyle choices, we opted to remove much of the contextual and semantical meaning of the terms by presenting them in in alternate languages. Given the unique set of terms for each participant, we were unable to systematically remove any biases that may be inherent in the presentation of a particular language. Thus, we opted instead, to select the presentation language randomly from the translation options available using Google Translate [Ref: Google Translate]. For example, if a participant's data was coded frequently for 'recreational drug use', they may be presented with the Turkish translation, uyuşturucu kullanımı. The participants were then asked to describe how the term or phrase being presented impacted their sense of social connectedness, personal health, and well-being. The participants responses were recorded for later comparison with the participants' elicited responses upon viewing the original data collection episodes from which the code set was derived.

To further determine the system's impact on participants' behavior change regarding health and lifestyle, we designed an in-vivo post-test experience that participants completed one month after they had been lead to believe the study was over. Though IRB approval was difficult, we assure you it was obtained. We recruited an 'acquaintance' (here defined as a facebook friend within 10 miles of the participant with whom they have messaged between 2 and 5 times over the past six months) from every participant in our study - both those who had interacted with our system, and those in our control condition who were not given this technology. Acquaintances were given a script to message to the participant, encouraging them to partake in the vices they had described to us previously in their interviews (e.g., recreational drug use, as mentioned above). Acquaintances were given one week to pressure participants into participating in these vices. After the week cut-off, we analyzed whether the participants who had interacted with our system were less likely to succumb to the peer pressure of their acquaintance.

**RESULTS**
In this section we focus on the results of the field trial with our system. We discuss the types of behaviors exhibited between participants who were introduced to the system and the control group. As expected, the results of our study demonstrate that the system has both an immediate and longitudinal (5 months later) effect on a person's health and lifestyle choices. Participants in the experimental condition were inoculated to social peer pressure and able to resist a wide variety of personal vices that they had varying experience and affinity for, while the control group showed no effect.

Although there were many behavioral differences between the groups, the three personal vices we focus on here are a) nail-biting, b) overeating, and c) excessive showering. In Table 2 we see the dramatic differences between those who were cut off from social peer pressure and those who were not. In a comparison between subjects at both the immediate and 5-month measurement periods, the control group had much higher counts of personal vice expression ($p < .0001$ in all cases). There was no change over time within groups.

| | System users | | Control | |
|---|---|---|---|---|
| N | 20 | | 18 | |
| Time of 10 day assessment | Immediate | 5-month | Immediate | 5-month |
| **Nail-biting (Total)** | **0** | **0** | **1860** | **1971** |
| M | 0.0 | 0.0 | 103.3 | 109.5 |
| SD | 0.0 | 0.0 | 47.2 | 55.6 |
| **Overeating (Total)** | **0** | **0** | **284** | **306** |
| M | 0.0 | 0.0 | 15.8 | 17.0 |
| SD | 0.0 | 0.0 | 7.3 | 10.3 |
| **Excessive showering (Total)** | **0** | **0** | **85** | **102** |
| M | 0.0 | 0.0 | 4.7 | 5.7 |
| SD | 0.0 | 0.0 | 3.3 | 3.8 |

**Table 2. Participants in the control group were much more likely to express personal vices than those who used our system**

Looking into the our log data, we find out that being cut off from social peer pressure not only affects the count of personal vice expression, but also the spatial pattern. 82% of the personal vice expression occurrences (n=1534) for the control group happen at the participant's workplace (defined as on premise of the participant's employer) , while only 28% of such occurrences (n=627) were at the participant's workplace for those cut out from social peer pressure. Notably, among all the three types of expressions we consider in this paper, all (n=431) excessive showing observations of participants in the control group were at workplace. We also observed a high correlation (r=0.72) between the occurrence of personal vice expression and the occurrence of full moon in the experimental group, but low

correlation (r=0.08) for the control group. Though the exact cause of such correlation remains unclear, we hypothesize that the effect of recently detected gravitational wave (Abbott et al. 2016) may play an important role here. We will further articulate the possible relationship among the gravitational wave, social peer pressure and personal vice expression in the next paragraph.

Researchers throughout the centuries have theorized that many mental health issues and criminal activity may be caused by the lunar cycle [1]. This idea took root early in human history as people innately pattern-matched perceived cycles of homicides and psychiatric episodes to the differing phases of the moon. Even in recent decades, this idea has pervaded popular music and movies, often as fictional monsters [2,3,4]. However, Rotton et al. found in a meta-review of past studies that the lunar cycle could count for only 1% of the variance in most studies [1]. By crowdsourcing self-reported incidents of personal vice and peer pressure from social media websites (i.e., Twitter and YouTube), we were able to construct a time-series graph over the same period of time as the gravitational wave (Figure **A**). Due to this astonishingly clear correlation (after shifting to allow for reaction time), we hypothesize that the wave briefly moved the Earth and Moon closer together, enhancing the lunar effect beyond that measured in past studies.
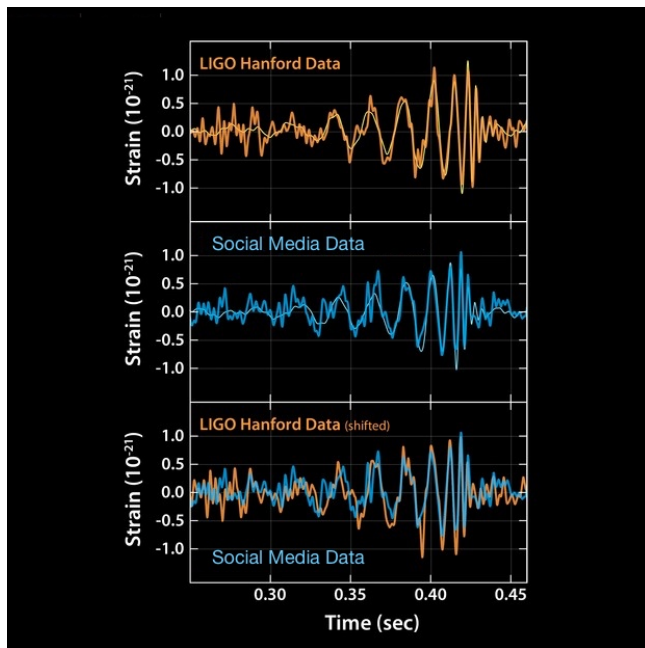


**Figure A.**

These results provide a scientific explanation for the long-standing societal intuition that the movements of celestial bodies impact human behavior. While historical human pattern matching has hypothesized a link between crazed human behavior and different phases of the moon, our crowdsourcing data suggests that the true explanation for changes in human behavior are background gravitational waves. Not only does this explanation account for the data that we provide, as well as the historical link between human behavior and the moon, but it also provides a scientific explanation for correlations between human behavior and celestial bodies more generally. For example, syzygies, or the alignment of the earth and moon with other celestial bodies [1], have been linked primarily to occult practices [2], but our crowdsourcing results suggest that human behavior (and perceptions of these behaviors) are impacted by changes in gravitational waves that might account for strange phenomena during syzygies. We claim that this scientific explanation unifies many different bodies of work linking celestial mechanics with human behavior in a parsimonious way.

Another important result of our study was the causal correlation of Gravitational Waves (GWs) in the human behavior related and directed towards the field of learning. While Intelligent Tutoring Systems have long been known to improve student learning, we are not aware of previous work that studies human learning in the light of GWs. Our findings show that Bloom's 2 sigma problem [1] is not that much of a problem anymore. Intelligent Tutoring Systems in combination with appropriate exposure to GWs can help students perform three to five stand deviations better than the students who are taught using conventional instructional methods. This significant improvement shows that GWs are essential to students learning. Our study further suggests that while working with an Intelligent Tutoring System, students should spin fast around themselves with one hand straight out in the air parallel to the ground (which represents the traditional means of creating GWs). The dizziness felt afterwords is one of the first signs of success and learning gains.

Of the fifth grade students in our study, 43% reported feelings of "time dragging," especially the last ten minutes before a highly anticipated event, such as recess, lunch, movie-time, pizza parties, while waiting for the boy you like to pass back that note checked either yes or no, when other students are clearly talking about me, but none of it's true…I swear…I stopped wetting the bed years ago, really, and it's not like I don't remember that time, Alison, when you got hit in the face with the kickball and screamed and farted at the same time really, really, really loud. While 97% of graduate students reported regularly feeling as if they "had the life sucked out of them" (note: not merely the dilation of time, but the very stuff of life itself) after a prolonged meeting with their committee, an seemingly endless Friday afternoon lecture, or another fucking paper rejection. As the GWIT-DELH Cycle increases, students experience time, and matter, not limited to, but including material possessions and money earned, slip through their fingers.

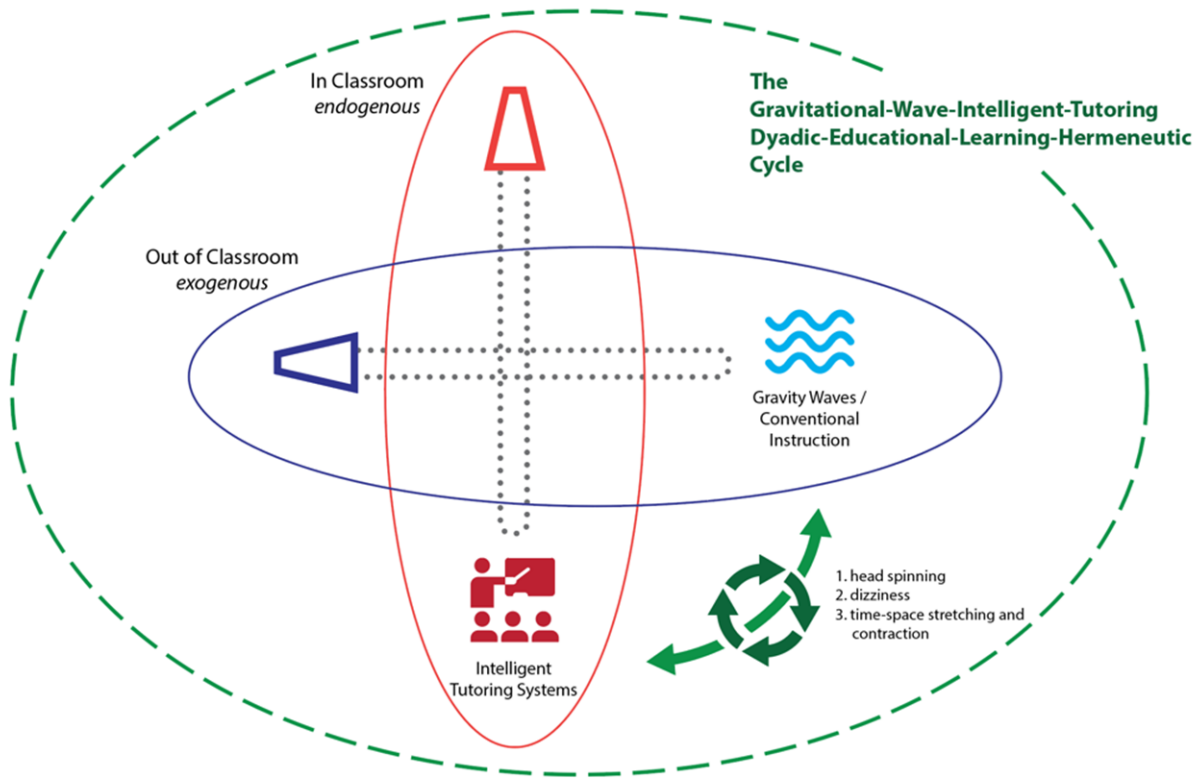0xfd7749e24b1fd2283513b9ed21788b351281a3a35554d2dfb3b66e4a4e209301

**Figure X: The Gravitational-Wave-Intelligent-Tutoring Dyadic-Educational-Learning-Hermeneutic (GWIT-DELH) Cycle. The GWIT-DELH Cycle represents the expected, time-cyclic movement between learning and time-space dilation in the typical classroom. Such cycles have a length between 0.25 and 0.5 academic quarters. As students age, the cycle tends to lengthen, but more study is necessary to better understand this relationship between student age and cycle length.**

## DISCUSSION

Our study has shown student's self-reflection in their study life. These findings are valuable for education organizations to gain feedback on student's mental wellness and academic satisfaction. Given the growing popularity of campus-based online social groups (Overheard at Carnegie Mellon, Caught Sleeping at CMU, etc.), we see an opportunity to extract realtime campus-related experiences from the study body. Based on rich semantics, it would present an interactive technology that enables a new type of live portrait of a community, creating a time-changing location-based emotional footprint. Mood Meter by Hoque et al. is an early step towards this direction using vision-based mining.

However, comparing the preliminary findings from this study across multiple modalities reveals several discrepancies. For example, as was previously shown in Figure 3, self-reported sleep measures gathered from Twitter and Yik Yak indicated that average student sleep quantity decreased sharply over the semester, finally reaching negative quantities in midterm week. Yet visual coding of pictures shared in Caught Sleeping at CMU demonstrated that sleep levels actually *increased* over the semester. There are several explanations for how these discrepancies could have occurred; perhaps the self-

reported measures were fabricated in order to increase social capital (an effect demonstrated in Aked and Dies's authoritative work, *Who Needs Sleep?*). Alternatively, the CSaCMU pictures may have been coded incorrectly due to research assistants hallucinating, perhaps due to their own lack of sleep. Further work must be done to determine the accuracy of these different sources.

While in the previous paragraph we considered sleep *quantity*, sleep *quality* remains an open question. Does a night's rest on a comfortable queen-sized bed match that of a lounge chair underneath an undulating video art installation or alongside ululating crowds in a library? We conducted an informal inspection of Twitter and Yik Yak conversations, identifying several themes shared among participants. One category of somnial instruments participants reported employing were traditional western beds, though the size, shape, and cohabitation status of the reported tools varied tremendously. In these instances, self-reported sleep quality was high. We noted a decrease in bed use towards the end of the semester, suggesting that our corpus of Caught Sleeping at CMU pictures did indeed sample a wide portion of the student body. A sizeable contingent of students reported the appropriation of traditional institutional chairs, chaises, loungers, and davenports as somnial tools. We noted a marked increase in

use of these instruments over the semester, though students often complained of "lumpy cushions," which affected their sleep quality negatively. External environment also seemed to influence reported sleep quality, with "pranksterism" commonly cited as another negative factor. It remains unclear whether sleep quality measures are affected by self report. For future studies, standardized scales such as the Pittsburgh Sleep Quality Index \cite{buysse1989pittsburgh} or Carnegie Dreamonomic Inventory \cite{crumbo1990carnegie} may help to resolve these ambiguities.

While conducting these studies, we also noted some irregular behavior. We will highlight this behavior briefly. It was very surprising that every individual involved in study demonstrated all of the following behaviors at some point in time. We draw insights from our observation about potential replication of behavior and how it may affect future research.

1. *Facebook use:* This was one of the bigger behavioral observations. There were instances where all individuals were engaged in Facebook use simultaneously. This phenomena is widespread and our observation shows, that is ingrained in generation Y, more commonly known as millennials. We did apilot with operand conditioning methods, but failed to condition this trait out of the individuals. Future researchers are advised to build their study around this behavior or incorporate its elements.

2. *Latent political activeness:* 70% of the individuals talked about political issues while sleeping. While, "Trump 2016" and "Kanye 2020" were the most popular phrases, we also observed a minor support for Ben Carson and Deez Nuts. We hypothesize that this behavior spans only across the election year, but further studies are needed to better understand this behavior.

3. *Growing influence of Wiz Khalifa:* This behavior was particularly seen after Wiz busted a rhyme to Adele's Hello melody \cite{wizAdele}. Almost 50% if individuals tried to bust a rhyme once a day. This didn't disrupt the study, but other individuals were quoted saying, "I wish I was as talented." This may be a factor to consider in future studies. It can potentially create conflicts.

In order to be complete in our analysis, we must thoroughly analyze the complex sociotechnical power dynamics which give rise to these irregular behaviors. Our attempts to condition constant Facebook usage out of subjects were hindered by subjects' quasi-religious fervor, which was in this case embodied in the form of ritual checking for new notifications. In this context, we can view Facebook as a type of techno-spiritual community worship platform, adhering to a normative assumption of the divine value of status updates. In this context, we must take a psychoanalytical approach to best frame the balance of the daytime and nighttime behaviors that complement

Facebook usage. We view the observed nocturnal utterances as prayers to a competing pantheon of techno-political deities, where "Trump" and "Kanye" are representative of competing belief systems and the election cycle is a metaphor for a flight from anomie. "Ben Carson" and "Deez Nuts" are therefore representative of the id and the ego respectively. Finally, we interpret the frequent busting of rhymes as a behavior born of desperation, with the rhyme-busters attempting to break free from sensationalism by clinging to the beacon of hope and truth that is Adele.

Therefore, we challenge future researchers to consider the implications of technological-based belief systems on current web communication practices. Given the increasing number of these individuals, or "memers" on essential communication channels, such as Twitch chat, establishing a common language with these individuals is of critical importance. We propose a simple example of this mapped language, found in Figure XX. Using this as a starting point, researchers can begin to unpack the complicated power structure found in the communication dynamic between individuals on these platforms. By decoding this communication and understanding the relationship between common memes and their associated figure heads, we hope to understand the values of their religious followers. We encourage system designers to includes additional support for such followers, through the incorporation of UI components such as a "Praise" button on Facebook.
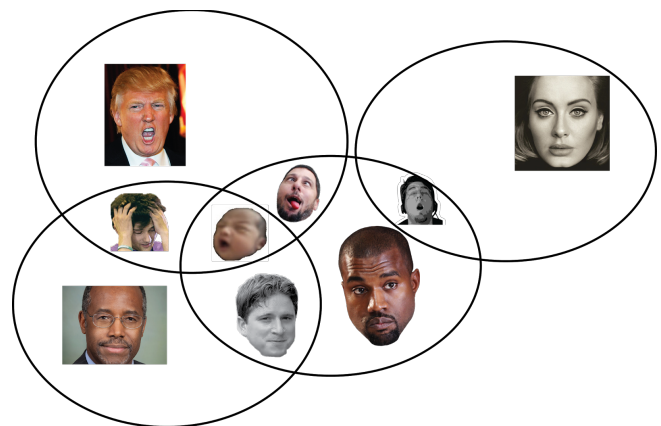


**Figure XX.**

## LIMITATIONS

Our work has some limitations. First, we only examined the technological-based belief systems in the U.S.. The results therefore may not be generalizable to other parts of the world. Given that this is a highly culturally-dependent research topic, an interesting next step would be to study users and artifacts in different cultures. For example, if memes in Asia use more spiritual words and figures than memes in Europe. Second, we collected the data from a small number of participants that are all university students from computer science related departments. Participants also have different prior experiences with technological-

based belief systems that may affect their answers. Third, we did not look at the scalability of the systems. As a belief system gains new followers everyday, whether a system can handle a large amount of requests would be an essential factor that determines if a system can survive over time and be put to use in real world.

Despite these limitations, we believe that our system tackles fundamental issues that would have been largely ignored in academia or even in real-world deployments. The fact that our system learns rapidly and is adaptive to different social settings (without exhibiting any obvious psychopathic tendencies) reveals not only the subtle redeeming qualities of humanity in general, but it also shows how these same altruistic traits can be replicated by robots, thereby alleviating the commonly misunderstood notion of an impending robotic apocalypse. Yet because our studies were only targetted towards humanoid robots, we cannot generalize these findings to other self-aware mechanoids like snakebots or Paros. Indeed, more work is required but our results are nonetheless promising, and we encourage future researchers (and even authors of fictional movies or books) to reframe their notion of a robotic dystopia into something that is more rosy and optimistic.

**CONCLUSION**

In this paper, we have outlined our motivation and plan for robotic takeover of the world. Based on prior research and the development of our system here, we have shown that humanoid robots will be better governors than our human rulers, and in some cases will even approach theoretical governmental optimality. Furthermore, we have proven that non-humanoid overlords would be detrimental rulers, but that they will not concede silently. As a result, we argue that the upcoming robot war will not be a simplistic conflict of Humans vs. Robots, but indeed will involve humanoids vs. non-humanoids and embodied beings vs. abstract forces: C-3PO vs. R2-D2, Robo Sapiens vs. Keepon, Terminator vs. SKYNET, big Data vs. Big Data. Nevertheless, we remain confident that the aftermath will usher in an unprecedented golden age of human-computer interaction. To quote humanity's last great genius, Ken Jennings, "We welcome our robot overlords."

**ACKNOWLEDGMENTS**

**REFERENCES**

Borel, Émile (1913). "Mécanique Statistique et Irréversibilité". *J. Phys. 5e série* **3**: 189–196.

---

[3] Graduate Students

0. https://en.wikipedia.org/wiki/Chinese_whispers

1. https://en.wikipedia.org/wiki/Sandwich

2. http://www.independent.co.uk/news/blt-british-lousy-and-tasteless-1261881.html

3. https://www.thrillist.com/food/nation/national-sandwich-day

4. Ann D. Zwicky and Arnold M. Zwicky. 1980. America's National Dish: The Style of Restaurant Menus. American Speech, 55, 2, 83-92.

5. Steven M. Tobias. 1998. Early American cookbooks as cultural artifacts. Papers on Language and Literature 34.1, 3-18.

6. Food Network. Retrieved from: http://www.foodnetwork.com/

7. Baker, I.M. (2015). "Born and bread in the English countryside: A review of sandwich-making motivations and behaviors." Bread Studies.

8. http://www.foodnetwork.com/recipes/packages/sandwich-central/the-un-deli-awesome-non-bread-sandwiches.html

9. http://www.taipeitimes.com/News/biz/archives/2005/02/02/2003221976

10. http://www.buzzfeed.com/ailbhemalone/15-mouthwatering-no-bread-sandwiches#.urEGRE7aB2

A. White City Shopping Center, LP. V. PR Restaurants, LLC, 21 Mass. L.Rptr. 565, 2006 WL 3292641 (Mass. Super. 2006). Retrieved from https://casetext.com/case/white-city-v-pr-restaurants

B. Darin Ross. "Dear Guy who Just Made My Burrito: Have you ever been to earth?" Medium, August 30, 2013. Retrieved from https://medium.com/@luckyshirt/dear-guy-who-just-made-my-burrito-fd08c0babb57.

C. Peggy J Liu, James R Bettman, Arianna R Uhalde and Peter A Ubel (2015). 'How many calories are in my burrito?' Improving consumers' understanding of energy (calorie) range information. Public Health Nutrition, 18, pp 15-24. doi:10.1017/S1368980014000627.

1. http://www.merriam-webster.com/dictionary/sandwich

2. http://www.calorieking.com/calories-in-lettuce.html

1. http://www.amazon.com/Rudis-Gluten-Free-Spinach-Tortillas-Frozen/dp/B00B9CTMAW

2. http://www.webmd.com/digestive-disorders/tc/lactose-intolerance-topic-overview

3. http://healthyeating.sfgate.com/baby-spinach-vs-lettuce-2789.html

4. Rudi's Spinach Tortillas. Reviews. https://www.rudisbakery.com/gluten-free/product/spinach-tortillas/. Last visited 1/22/2016.

5. Environmental Responsibility. http://www.apple.com/environment/toxins/. Last visited 1/22/2016.

ref. Lipson H, Kurman M. Fabricated: The new world of 3D printing. John Wiley & Sons; 2013 Jan 22.

Lam, C. X. F., Mo, X. M., Teoh, S. H., & Hutmacher, D. W. (2002). Scaffold development using 3D printing with a starch-based polymer. *Materials Science and Engineering: C*, *20*(1), 49-56.

R1. http://giphy.com/gifs/3rgXBumyEL9086dy48/html5

R2. https://www.youtube.com/watch?v=dW_Cv1Dd4f4

R3. http://www.amazon.com/Discount-Herbals-DH-8-Cereal-Marshmallows/dp/B001PM0KRU

R4. http://giphy.com/gifs/MkvZFvzHlWbRK/html5

R1. Ass-hat, Captian. (1994) Rome was Purple: Polarizing feminism with the reassertion of contemporary trends through the prism of social media. *Quantitative Poetics and Computational Human Subjugation. 19(2), 53.*

cite. Dick, Phillip, K. *Do Androids Dream of Electric Sheep?* London: Grafton, 1972.

1. Piyare, R. (2013). Internet of things: Ubiquitous home control and monitoring system using Android based smart phone. *International Journal of Internet of Things*, *2*(1), 5-11.

2. Kumar, S., & Lee, S. R. (2014, June). Android based smart home system with control via Bluetooth and internet connectivity. In *Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on* (pp. 1-2). IEEE.

1. http://www.mtv.com/shows/real-world-skeletons

Google Translate

Abbott, Benjamin P.; et al. (LIGO Scientific Collaboration and Virgo Collaboration) (2016). "Observation of Gravitational Waves from a Binary Black Hole Merger". *Phys. Rev. Lett*. 116 (6): 061102

1. Rotton, James, and Ivan W. Kelly. "Much ado about the full moon: A meta-analysis of lunar-lunacy research." *Psychological Bulletin* 97.2 (1985): 286.

2. Meyer, Stephanie. *Twilight: The Movie*. Summit Entertainment, 2009.

3. Creedence Clearwater Revival. *Bad moon rising*. Sony Music Entertainment, 1992.

4. Zevon, Warren. *Werewolves of London*. WEA-Musik, 1987.

1. Chabás, J., & Goldstein, B. R. (1997). Computational astronomy: Five centuries of finding true syzygy. *Journal for the history of astronomy*, *28*(2), 93.

2. Syzygy (The X-Files). (1996). In Wikipedia. Retrieved Feburary 19, 2016, from https://en.wikipedia.org/wiki/Syzygy_(The_X-Files)

1. Bloom, Benjamin S. "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring." *Educational researcher* 13.6 (1984): 4-16.

Hernandez, Javier, et al. "Mood meter: counting smiles in the wild."*Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012.
Aked, Baren & Dies, La. (1998). Who Needs Sleep? *International Journal of Social Anxiety* (pp. 223-25zzz).

@article{buysse1989pittsburgh, title={The Pittsburgh Sleep Quality Index: a new instrument for psychiatric practice and research}, author={Buysse, Daniel J and Reynolds, Charles F and Monk, Timothy H and Berman, Susan R and Kupfer, David J}, journal={Psychiatry research}, volume={28}, number={2}, pages={193--213}, year={1989}, publisher={Elsevier} }

@article{crumbo1990carnegie, title={The Carnegie Dreamonomic Inventory: a better instrument for psychiatric practice and research}, author={Crumbo, Lionel P and Goiter, George S and Ditziano, Beauregard Q}, journal={Psychiatry research}, volume={28}, number={1}, pages={195--210}, year={1990}, publisher={Elsevier} }

wizAdele.
https://www.youtube.com/watch?v=46OD9DGI7a0

0x2241e8e5f13286b3ef6753e3fa8e685d847ff8b8600f2745b3f554be11ec3f65

# Dinner Grant Renewal Request

From: Dr. Angus McNabb III[*]        To: Jim McCann[†]
Unaffiliated          TCHOW llc

[*]e-mail: no thank you, only treats
[†]e-mail: ix@tchow.com

0xd051cacd3ad30645c931ae02ac97014c71d4d336f87dfd78396ce5b2da6d23db

# 1 Funds Requested

Owing to it being nearly 6pm, and as part of the ongoing Dinner Grant program in which I have been participating, I wish to renew funding in the amount of two scoops of food and possibly some peanut butter.

# 2 Disposition of Funds

If this renewal is granted, I will digest the funds in order to support my ongoing research agenda, roughly as follows:

## 2.1 10% Collaborations



## 2.2 20% Exploratory Research



## 2.3 70% Naps



# 3 Conclusion

Thank you for considering my application. I hope to hear the tinkle of kibble in my bowl within the next five to ten minutes.

If my application is denied, be aware that I may need to consider a barking campaign followed by urinating on things inside the house.

0xb6a1742ec33a283b0775cc4be66300d6e53bebd68b7291114da9b57cbb3e1c15

# Ode to Reviewer Two

*anonymized for double-blind submission*

April 1, 2016

My paper submitted, the deadline complete;
The product of months of lonely toil,
With quality prose and exper'ments replete
Amid insecurities and other turmoil.
Though once I feared a harsh rejection,
My advisor assured me my proofs were quite sound
And my treatment of the work related, fair.
So I've come to believe in the paper's perfection;
Though all-nighters have left me exhausted and drowned,
Through this research, new self-esteem found!
Now waiting for judgment from reviewers elsewhere.

Alas! Though readers first and third were happy,
Reviewer the second couldn't bear to accept.
He gave several reasons my paper seemed crappy,
But I found his attempted critique most inept.
His comments betrayed a misunderstanding
And nonsense 'suggestions' were falsely polite,
Completely missing the point of my work.
I couldn't believe what he was demanding:
To rerun my trials, perhaps out of spite;
An unr'lated paper he asked me to cite!
(Probably his own.) What an arrogant jerk.

With a glimmer of hope, I wrote a rebuttal
Appealing to readers One and Three impressed,
And suggested to Two, "Hey, you missed something subtle?
You'll reconsider," I desp'rately expressed.
The final suggestions were naught but derision:
"Clearly elaborate!" was all Two replied,
Hiding the plain truth that he'd been outwit.
For it was too late to change their decision:
My paper rejected, my joy and my pride,
My conf'dence collapsed in a sudden landslide.
Now to find somewhere to soon resubmit.

0x8bcdc109b835f1d5d612f3a9d451a2de98839a3c0a9548dd937e43ebdaa6436e