A Record of The Proceedings of SIGBOVIK 2010

April 1st, 2010
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA[*]

http://sigbovik.org/2010

[*] now available with 10% more healthcare

ach Association for Computational Heresy

Advancing Computing as Tomfoolery & Distraction
— or —
Hey, Kids, Get Off My Lawn

DISCLAIMER: LaTeX Allergies

In the processing of LaTeX into commercial documents, chemicals are added which may produce allergic reactions in certain people. Mild reactions simply appear as dry, itchy skin or a mild rash. Severe reactions, however, can be life threatening, often involving respiratory difficulties or even shock. These symptoms may not appear until several hours after exposure. A physician should always be consulted even for minor symptoms, since further exposure could result in a more serious allergic reaction.

Anyone can develop an allergy to LaTeX, but those with ongoing exposure are at highest risk. This group of high risk employees includes mathematicians, theoretical computer scientists or anyone else who regularly uses or is exposed to LaTeX. People with other allergies have increased risk, especially if those allergies are related to parameter passing mechanisms or lack of data abstraction.

Most people think of higher mathematics when they think of LaTeX, but LaTeX may also be present in many office and household products, including conference proceedings, software manuals, instruction booklets, grant proposals, and even campus mail. Individuals with a LaTeX allergy should be aware of any products that may trigger a reaction. Always check your mime type before opening any e-mail attachments!

The best way to avoid an allergic reaction to LaTeX is to use LaTeX-free products where possible, wear specifically designed LaTeX-filter safety goggles, or use barrier protection. Also, avoid direct sunlight if you must read LaTeX documents (sunlight enhances poor typeface choices and ligatures), avoid the printer rooms whenever possible, and make sure coworkers and health care providers know of your allergy. Always wash your hands after leaving Emacs LaTeX mode, and if you already have the allergy, wear a medical alert bracelet.

- Pfenning, Pfenning, and Pfenning, pro-bono legal council for the Association for Computational Heresy

A Message
From The Organizing Committee

We are honored, delighted, excited, and amused to present to you this, the record of the Fourth Annual Intercalary Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's (2^6)th birthday. And what a wonderful fourth year it's been! SIGBOVIK is exactly as strong as ever, with exactly more or less precisely the same number of papers submitted and accepted as last year.

In the truest echo of the ideals underlying SIGBOVIK, it's been a landmark year for collaboration, with vastly more papers by multiple authors, many of them based on multi-person projects, some of them dangerously close to resembling serious research. At the far end of high collaboration, this year's program includes the first ever SIGBOVIK paper with more than ten authors! Let us hope that the glorious tradition of comradely camera-ready camaraderie continues without end. Thank you all for making SIGBOVIK the indescribable something that it is.

As in past years, we have taken the liberty of dividing the proceedings into coherent subject-specific tracks for the purposes of easy access and clarity of reference. This process always involves a certain amount of favoritism and unfairness, though, as some papers must appear in the coveted early tracks of the program while other papers necessarily occupy the inevitable but unenviable role of filling out the final track. But denizens of Academia, friends of Bovik, fear not! In order to combat this rampant unjustness, we have chosen *not* to order the tracks this year: to us, every track is Track 1! The Track 1s appear in a random order unique to each copy of the proceedings, as guaranteed by the following true random number generator:

```
fun rand () = 100  (* chosen by random.org, using noises from the
                 atmosphere!; guaranteed to be truly random,
                 and if you don't believe it, then maybe you
                 just don't know what it means to be random,
                 'cause you know, people aren't very good at
                 being random or understanding randomness or
                 anything like that, so yeah.. *)
```

Next year, we hope to continue our experimental journey into objectivity with double-blind submissions: by not knowing just which conference you're submitting to, you guarantee more fairness for everyone!

As if organizing SIGBOVIK wasn't exciting enough, this year also marked the SIGBOVIK's first ever complete-server-downtime-during-the-seriously-final-submission-deadline. The issue was noted around 6 pm Wasilla time, just a quarter day before submissions closed, at which point the entirety of the SIGBOVIK Committee for Deadline Uptime excitedly jumped into action! A flurry of email(s) was sent to Computer Club Central, the organization which generously hosts SIGBOVIK's servers, to little observable effect. Thanks though to the ingeniously heroic actions of the spontaneously formed SIGBOVIK Task Force on Publicizing the EasyChair Submission URL, submissions were almost entirely unaffected. (Whew!)

And so, without further ado, we proudly present SIGBOVIK 2010; may it enlighten and inspire! Enjoy responsibly.

Sincerely,
The SIGBOVIK 2010 Organizing Committee

Table of Contents

Track 1: Correspondence Theory

Track 1: Hypertext

Track 1: Defeating Evil

Track 1: Advanced Technology in a Magical and Revolutionary Device at an Unbelievable Price

Track 1: Productivity, We Say Ironically

Track 1: Perplexity Theory

Track 1
Correspondence Theory

o **CORRESPONDING AUTHORS**

o **THE HILLS ARE ALIVE WITH THE SOUND OF CORRESPONDENCE**

o **‘ROUND CORRESPONDENCE**

o **CORREPWNDENCE**

An Algorithm for Erdos-Bacon Number Minimization

Charles Garrod
Carnegie Mellon University
Pittsburgh, PA 15213

Maria Klawe
Harvey Mudd College
Claremont, CA 91711

Iain Matthews
Disney Research, Pittsburgh
Pittsburgh, PA 15213

Laura Trutoiu
Carnegie Mellon University
Pittsburgh, PA 15213

Edilson de Aguilar
Disney Research, Pittsburgh
Pittsburgh, PA 15213

Michael Ashley-Rollman
Carnegie Mellon University
Pittsburgh, PA 15213

Harry Q. Bovik
Carnegie Mellon University
Pittsburgh, PA 15213

Joanna Bresee
NASA Ames Research Center
Mountain View, CA 94043

Jeff Carlson
Carnegie Mellon University
Pittsburgh, PA 15213

Mark Desnoyer
Carnegie Mellon University
Pittsburgh, PA 15213

Michael Dinitz
Carnegie Mellon University
Pittsburgh, PA 15213

Owen Durni
Carnegie Mellon University
Pittsburgh, PA 15213

Jack Ferris
Carnegie Mellon University
Pittsburgh, PA 15213

Ron Garcia
Carnegie Mellon University
Pittsburgh, PA 15213

Robert Harper
Carnegie Mellon University
Pittsburgh, PA 15213

Benoît Hudson
Autodesk Canada, Montréal
Montréal, Canada

Greg Humphreys
NVIDIA Corporation

Michelle Hutton
The Girls Middle School
Mountain View, CA 94043

Kelli Ireland
University of Pittsburgh
Pittsburgh, PA 15213

Katrina Ligett
Cornell University
Ithaca, NY 14850

Ciera Jaspan
Carnegie Mellon University
Pittsburgh, PA 15213

Jennifer Marlow
Carnegie Mellon University
Pittsburgh, PA 15213

Davin McCall
La Trobe University
Melbourne, Australia

James L. McCann
Carnegie Mellon University
Pittsburgh, PA 15213

Roman Mitz
Carnegie Mellon University
Pittsburgh, PA 15213

Iulian Moraru
Carnegie Mellon University
Pittsburgh, PA 15213

Amar Phanishayee
Carnegie Mellon University
Pittsburgh, PA 15213

David Reed
Creighton University
Omaha, NE 68178

Jason Reed
University of Pennsylvania
Philadelphia, PA 19104

Margaret Reid-Miller
Carnegie Mellon University
Pittsburgh, PA 15213

Kathryn Rivard
Carnegie Mellon University
Pittsburgh, PA 15213

Or Sheffet
Carnegie Mellon University
Pittsburgh, PA 15213

Daniel D. Sleator
Carnegie Mellon University
Pittsburgh, PA 15213

Ekaterina Spriggs
Carnegie Mellon University
Pittsburgh, PA 15213

Matthew L. Stanton
Carnegie Mellon University
Pittsburgh, PA 15213

Mark J. Stehlik
Carnegie Mellon University
Pittsburgh, PA 15213

Megan Thorsen
Sunnyvale, CA 94087

Vijay Vasudevan
Carnegie Mellon University
Pittsburgh, PA 15213

Kevin Waugh
Carnegie Mellon University
Pittsburgh, PA 15213

Erik Zawadzki
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

A small Erdős number has long been a source of pride for mathematicians. With Paul Erdős's death, however, obtaining an Erdős number less than two has become difficult. In this paper we propose Erdős Number Minimization, reducing the Erdős number of many mathematicians and computer scientists to two. Although we do not prove the claim, we believe that the Erdős number we obtain is optimal without significant biological or paranormal advances. We extend ENM with Bacon Number Reduction, a non-optimal variant to reduce Bacon numbers, obtaining small but non-minimal Erdős-Bacon numbers.

1. INTRODUCTION

From his birth in 1913 to his death in 1996, Paul Erdős co-authored nearly 1500 papers, working with nearly 500 collaborators to become the most prolific mathematician in modern times [5]. This proliferation led mathematicians to humorously define Erdős numbers: a person's Erdős number is the distance between that person and Paul Erdős in the academic paper collaboration graph [4]. I.e., Paul Erdős is the unique person with Erdős number zero; all of Erdős's immediate collaborators have Erdős number one; in general, if you publish an academic paper with a collaborator who has Erdős number x and none of your other collaborators has Erdős number less than x, your Erdős number is $x + 1$. A similar Bacon number has been proposed for actor Kevin Bacon, except using collaborations in movies instead of collaborations in academic papers. Erdős-Bacon numbers were subsequently defined to be the sum of each person's Erdős and Bacon numbers.

Obtaining a small Erdős number has become increasingly difficult since Erdős's death, but mathematicians with chutzpah have continued to publish papers with Erdős after his death; this brings his total number of known publications to 1525, his collaborator count to 511, and the Erdős number of the chutzpah-bearing mathematician to one. The latest publications co-written with Paul Erdős appeared more than ten years after his death, and with additional rumored works in progress, Erdős's publication list is expected to grow.

In this paper, we describe a technique called Erdős Number Minimization (ENM). Although we do not prove the claim, we believe that, without significant biological or paranormal advances, ENM is optimal for mathematicians who have been unable to work with Erdős in the past. We supplement ENM with a similar but non-optimal Bacon Number Reduction algorithm to obtain ENM+BNR.

2. ERDOS NUMBER MINIMIZATION

□.

3. BACON NUMBER REDUCTION

See [6] for more information.

4. EVALUATION OF ENM+BNR

In [3], Maria Klawe co-authored with Paul Erdős, thus guaranteeing that all authors on this paper have an Erdős number no greater than two. Additionally, Iain Matthews appeared in [2] with Sigourney Weaver, who appeared in [1] with Harry Connick Jr., who appeared in [7] with Kevin Bacon. By [6], all collaborators on this paper also appear in the movie credits about the paper and thus have a Bacon number no greater than four, making their Erdős-Bacon number no greater than six.

5. CONCLUSIONS

In this paper, we propose Erdős Number Minimization (ENM), an algorithm to minimize Erdős numbers for authors unable to collaborate with Paul Erdős. We extend ENM with Bacon Number Reduction (BNR), a non-optimal algorithm to reduce Bacon numbers to obtain ENM+BNR, and algorithm that significantly reduces Erdős-Bacon numbers. Finally, we apply ENM+BNR to obtain Erdős-Bacon numbers no greater than six.

6. ACKNOWLEDGMENTS

This paper was most definitely not authored by Luis von Ahn, David G. Andersen, Alexei Efros, Michelle Goodstein, Raffay Hamid, Mike Mistry, Don Sheehy, Robert Simmons, or Ronit Slyper.

7. REFERENCES

[1] Ann Biderman and David Madsen. Copycat, 1995. http://www.imdb.com/title/tt0112722/.

[2] James Cameron. Avatar, 2009. http://www.imdb.com/title/tt0499549/.

[3] Paul Erdős, Frank Harary, and Maria Klawe. Residually complete graphs. In *Proc. Sympos. Combinatorial Mathematics and Optimal Design*, pages 117–123, Colorado State University, Fort Collins, Colorado, 1978.

[4] Caspar Goffman. What is your Erdős number? *The American Mathematical Monthly*, 76(7):791, 1969.

[5] Jerry Grossman, Patrick Ion, and Rodrigo De Castro. Erdős number project, 2010. http://www.oakland.edu/enp/.

[6] Iain Matthews, Laura Trutoiu, Charles Garrod, and Maria Klawe et al. Bacon Number Reduction, 2010. http://tinyurl.com/ENM+BNR.

[7] Willie Morris and Gail Gilchriest. My Dog Skip, 2000. http://www.imdb.com/title/tt0156812/.

Classical Modal Logic

the magic of music, judgementally reconstructed

William Lovas

March 21, 2010

Abstract

♪♩ *The hills are alive... with the sound of music.* ♫

1 Introduction

Typically, musicians concern themselves with just one or two notions of music, in the absolute. Modal music is a generalization of traditional music in which we can consider music from multiple perspectives, or *modes*. In fact, modal music dates back to the antiquities, and so we may more properly speak of *classical modal music*. In what follows, we develop a logic for classical modal music based around a glorious and perfect septuality of seven different modal judgements.

2 The Logic of Classical Modes: Judgements

Ordinary intuitionistic logic is typically concerned with the judgement A *true*, meaning that the proposition A is true. Classical modal logic is a multimodal logic [Ree09] based on seven judgements, each one representing one of the classical modes:

- C *ionian* — C has the Ionian (or "major") mode
- D *dorian* — D has the Dorian mode
- E *phrygian* — E has the Phrygian mode
- F *lydian* — F has the Lydian mode
- G *mixolydian* — G has the Mixolydian mode
- A *aeolian* — A has the Aeolian (or "minor") mode
- B *locrian* — B has the Locrian mode

$$\Gamma \quad \Delta \quad \Theta \quad \Lambda \quad \Xi \quad \Pi \quad \Sigma \quad \Phi \quad \Psi \quad \Omega$$

Figure 1: Greek letters appropriate for use as logical contexts.

Since each of these judgements may appear on both the left- and the right-hand side of a sequent, we split the usual context into seven zones, one for each modal judgement. A difficulty presents itself in choosing appropriate names for the contexts, as there are only 10 capital Greek letters that are not written identically as some Roman capital letter[1] (see Figure 2). We make the obvious choices for the Dorian (Δ), Phrygian (Φ), and Lydian (Λ) contexts. Mi**x**olydian with its infixed x gets as its context the Greek letter Ξ. The usual ordinary or "major" mode, Ionian, gets the usual ordinary context, Γ. The Aeolian or "minor" mode gets Σ, which looks kind of like a sideways "M". And finally, the highly unstable and notoriously difficult-to-resolve Locrian mode is represented by Ω, a symbol which calls to mind non-termination and inconsistency. The three remaining unused Greek letters (Θ, Π, and Ψ) are left to future work.

Classical modal logic thus involves sequents of the form:

$$\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash J$$

for J one of the seven modal judgements, and where

$$\begin{aligned}
\Gamma &= C_1 \textit{ ionian}, \ldots, C_i \textit{ ionian} \\
\Delta &= D_1 \textit{ dorian}, \ldots, D_j \textit{ dorian} \\
\Phi &= E_1 \textit{ phrygian}, \ldots, E_k \textit{ phrygian} \\
\Lambda &= F_1 \textit{ lydian}, \ldots, F_l \textit{ lydian} \\
\Xi &= G_1 \textit{ mixolydian}, \ldots, G_m \textit{ mixolydian} \\
\Sigma &= A_1 \textit{ aeolian}, \ldots, A_n \textit{ aeolian} \\
\Omega &= B_1 \textit{ locrian}, \ldots, B_\lambda \textit{ locrian}
\end{aligned}$$

This excessively complex sequent is in keeping with the classical nature of the logic: classical music is serious business!

We want to be able to make inferences like:

$$\frac{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash C \textit{ ionian}}{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash D \textit{ dorian}}$$

Therefore, in addition to the usual hypothetical reasoning principles, we have judgemental principles corresponding to the glorious septuality.

Principle 1 (Septuality) *For all notes N, modes m, and natural numbers k:* $\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash N\ m$ *if and only if* $\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash (N +_7 k)\ (m +_7 k)$.

where the *transposition* operator $+_7$ is defined in the obviously canonical way.

[1] excluding the obsolete letters digamma, qoppa, san, and sampi, most of which are unavailable in LaTeX.

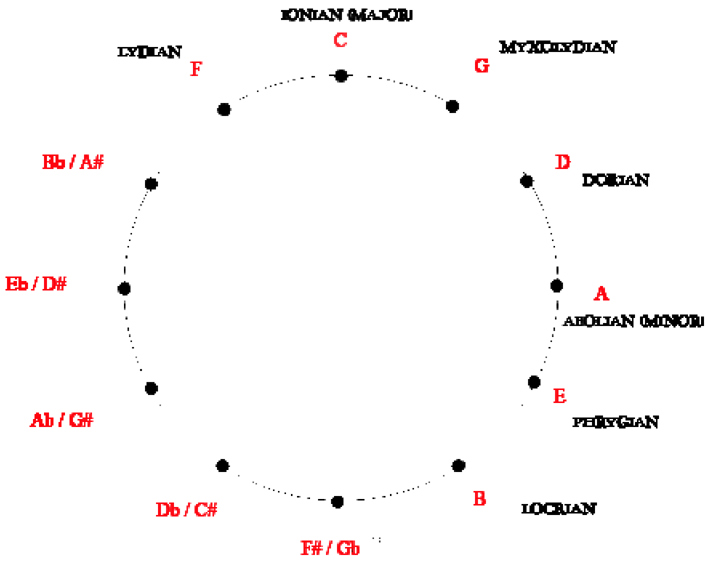


Figure 2: ♫ *The Ciiiiiiiiircle... the Circle of Fifths!* ♪

3 Circle of Fifths

The Principle of Septuality hints at the deep symmetries of classical modal music, brought out by the Circle of Fifths (Figure 2). To internalize the notion of transposition, we introduce the connective $\bigcirc N$, read "transpose N".

$$\frac{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash N\ (m +_7 1)}{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash \bigcirc N\ m}\ \bigcirc I \qquad \frac{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash \bigcirc N\ m}{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash N\ (m +_7 1)}\ \bigcirc E$$

But of course, a circle goes both ways, so as is traditional in modal logics, we also have the backward-looking dual connective $\bullet N$, read "untranspose N".

$$\frac{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash N\ (m -_7 1)}{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash \bullet N\ m}\ \bullet I \qquad \frac{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash \bullet N\ m}{\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \vdash N\ (m -_7 1)}\ \bullet E$$

where, again, the untransposition operator $-_7$ is completely standard.

The rules for both connectives are easily seen to be locally sound and complete. But more interestingly, the connectives lend themselves easily to a clear axiomatic characterization. The first axiom internalizes the Principle of Septuality,

$$\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc N \equiv N$$

that is to say, $\bigcirc$ is 7-involutive. The second and third axioms capture the notion that transposition and untransposition are mutually inverse,

$$\bigcirc\bullet N \equiv N \equiv \bullet\bigcirc N$$

4 Future Directions

4.1 Relationships with Other Logics

It is well-known that the intuitionistic modal logic S4 [PD01] has exactly seven distinct generalized modalities: $\cdot$, $\Box$, $\Diamond$, $\Box\Diamond$, $\Diamond\Box$, $\Box\Diamond\Box$, and $\Diamond\Box\Diamond$. An obvious direction of future work would be to connect these seven modalities to the seven modal judgements of our classical modal music logic.

4.2 Other Connectives

Generalizing from our experience with modern music, we might infer the existence of connectives $N\sharp$ and $N\flat$, some instances of which would have interpretations "between the modes", with the following axiomatic characterizations:

$$(N\sharp)\flat \equiv N$$
$$(N\flat)\sharp \equiv N$$

along with rules like

$$\frac{N \ \textit{phrygian}}{N\sharp \ \textit{lydian}}$$

An obvious question to ask is just what modal judgemental principles would these connectives correspond to? How could they be "externalized" as judgements in a globally sound and complete way?

Of course, for algebraic completeness, we would want to include the identity or "natural" connective $N\natural$, where

$$N\natural \equiv N$$

which derives semantically from the well-known categorical notion:

$$\begin{tikzcd}
{} \arrow[d, no head] & G\sharp \arrow[dd, "G\quarternote"] \\
F\sharp \arrow[ur, "\eta\sharp"] \arrow[dd, "F\quarternote"'] \arrow[r, phantom, "\natural"] & {} \\
{} & G\flat \arrow[d, no head] \\
F\flat \arrow[ur, "\eta\flat"'] & {}
\end{tikzcd}$$

4.3 Proof Theory

A rich and expressive proof theory cries out to be invented[2] in which proofs are compositions and proof checking is harmoniousness validation. Naturally, this proof theory should be amenable to canonical forms [WCPW02, Bac47c].

4.4 Classical Classical Modal Logic

In this paper we cover the *intuitionistic* variant of classical modal logic, but one could easily consider a multiple-conclusion style *classical* variant. The chief difficulty is coming up with enough Greek letters to round out the zones, since as explained in Section 2, the present work leaves only 3 suitable letters. One rather pedestrian option would be to leverage subscripts:

$$\Gamma_L; \Delta_L; \Phi_L; \Lambda_L; \Xi_L; \Sigma_L; \Omega_L \Longrightarrow \Gamma_R; \Delta_R; \Phi_R; \Lambda_R; \Xi_R; \Sigma_R; \Omega_R$$

(Yawn.) A more creative option would be to press into service the 4 canonical obsolete Greek letters, digamma, qoppa, san, and sampi, bringing the total unused letters to a perfect 7 — a tempting coincidence! But unfortunately, these letters are difficult to typeset, and in any case bear little connection to the logical content of the seven zones.

So a cleverer alternative would be to use very powerful typesetting features to allow us to reuse our carefully chosen seven contexts while clearly distinguishing the “false” ones from the “true” ones:

$$\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega \rightleftharpoons \rotatebox{180}{$\Gamma; \Delta; \Phi; \Lambda; \Xi; \Sigma; \Omega$}$$

An advantage of this notation is that sequents may be read either right-side up or upside-down, left-to-right or right-to-left, a trick used to great comic effect by musical artists such as Borge [Bor90] and Bach [Bac47b, Bac47a] and computer scientists such as Hofstadter [Hof79] and Mairson [Mai05].

Finally, a third option would be to follow in the footsteps of many presentations of classical *linear* logic [Gir87], which leverage duality to halve the number of required zones, and find some way of leveraging *septuality*, a concept seven times more powerful, to shrink the size of a classical modal sequent.

5 Coda 𝄌

If modals be the logic of love, *calculemus!*

[2] “Discovered.” — Ruy Ley-Wild

References

[Bac47a] Johann Sebastian Bach. Canon 1, *a 2 cancrizans* ("crab canon"). In *The Musical Offering* [Bac47c].

[Bac47b] Johann Sebastian Bach. Canon 9, *Quærendo Invenietis*. In *The Musical Offering* [Bac47c].

[Bac47c] Johann Sebastian Bach. *The Musical Offering*. BWV 1079. July 1747.

[Bor90] Victor Borge. William Tell backwards. In *The Best of Victor Borge: Act One & Two*. GMZ Productions, 1990. Available from `http://www.youtube.com/watch?v=MuWUp1M-vuM`.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Hof79] Douglas R. Hofstadter. Crab canon. In *Gödel, Escher, Bach: an Eternal Golden Braid*, pages 199–203. Basic Books, New York, 1979.

[Mai05] Harry Mairson. Constructive classical logic. POP Seminar at Carnegie Mellon University (`http://www.cs.cmu.edu/afs/.cs.cmu.edu/Web/Groups/pop/seminar/050315.html`), March 2005.

[PD01] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001. Notes to an invited talk at the *Workshop on Intuitionistic Modal Logics and Applications* (IMLA'99), Trento, Italy, July 1999.

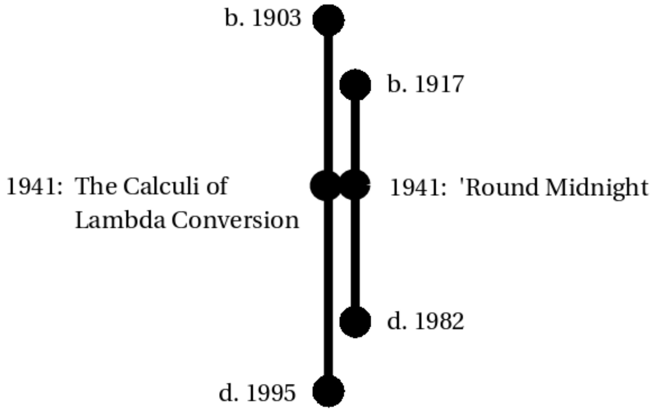
[Ree09] Jason Reed. A judgmental deconstruction of modal logic. Available from `http://www.cs.cmu.edu/~jcreed/papers/jdml.pdf`, January 2009. Revised and redacted at `http://www.cs.cmu.edu/~jcreed/papers/jdml2.pdf`, May 2009.

[WCPW02] Kevin Watkins, Iliano Cervesato, Frank Pfenning, and David Walker. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.

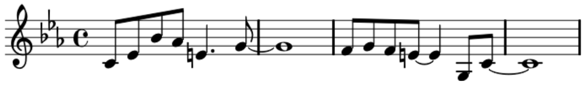
The Church-Monk Isomorphism

David Renshaw

“All musicians are subconsciously mathematicians.” —Thelonious Monk
“And vice versa.” —Alonzo Church



Church’s Thesis: $(\forall x.\exists y.\phi(x,y)) \rightarrow (\exists f.\forall x.\exists y\, u.\mathbf{T}(f,x,y,u) \wedge \phi(x,y))$

Monk’s Mood: 

References

[1] W. Lovas. Classical Modal Logic: the magic of music, judgementally reconstructed. *SIGBOVIK 2010*.

[2] D. Renshaw. The Davis-Davies Isomorphism. To appear (maybe).

Paper 3: The Church-Monk Isomorphism

Reviewer 1: Jeffrey Renshaw
Rating: 3 (strong accept)
Confidence: 3/4

What a remarkable coincidence that I was asked to review this paper! Just the other day I was pondering some deep connections between Karl Marx and Karl Goldmark. David's paper brilliantly elucidates many ideas that I had been struggling to grasp. It looks like we're on to something big here. Apparently, isomorphisms are far more common than was previously believed. I eagerly anticipate future work in this direction.

DPS Conversion: A New Paradigm in Higher-Order Compilation

Ben Blum (bblum@andrew.cmu.edu)

2010.04.01

1 Abstract

This paper introduces a new judgment and accompanying system of rules for use in some higher-order compilation stages that allows reduction of noobs into more damaged noobs, and finally conversion into gibs. The translation is done by repeated applications of damage-per-second (DPS), which we denote dps. The primary judgment is $\Gamma, \mathsf{dps} \vdash \mathsf{noob} \downarrow \mathsf{gibs}$, which is mostly useful only in first-person shooter or MMO cases. We provide a proof for local soundness and completeness, and finally an example usage case.

2 Definitions

In certain situations, we come across constructs of the following form that we wish to DPS-convert.

$$\begin{aligned}
\mathsf{noob} &::= \times_n[\texttt{hp} : \texttt{int}, \ldots] \\
\mathsf{dps} &::= \texttt{ROCKET} \mid \texttt{AWP} \mid \texttt{SPELLCASTER} \mid \texttt{SWORD} \mid \ldots \mid \texttt{HEADSHOT}[\mathsf{dps}] \mid \times[\mathsf{dps}, \mathsf{dps}] \\
\mathsf{gibs} &::= \text{☠}
\end{aligned}$$

- The rest of the components of noob are system-specific. All we care about is the noob's hp total.
- Note that the final construct in the dps definition represents that damage can be dealt from multiple sources at a time, and we are only concerned about the total.

3 Rules

We set forth a series of rules for performing the following conversion:

$$\begin{aligned}
\mathsf{dps} &\quad \text{✹—} \quad \mathsf{int} && (1) \\
\Gamma, \mathsf{dps} &\quad \vdash \quad \mathsf{noob} \rightsquigarrow \mathsf{noob}' && (2) \\
\Gamma, \mathsf{dps} &\quad \vdash \quad \mathsf{noob} \downarrow \mathsf{gibs} && (3)
\end{aligned}$$

We assume standard arithmetic operations $+$, $-$, and $*$, and let $\Gamma = \epsilon \mid \Gamma, \mathsf{dps}$.

1. The ✹— judgment tells us how much damage a given dps does. The exact implementation of these rules may vary by system; we present an example.

$$\frac{}{\texttt{ROCKET}\,\text{✹—}\,25} \qquad \frac{}{\texttt{AWP}\,\text{✹—}\,15} \qquad \frac{}{\texttt{SPELLCASTER}\,\text{✹—}\,8} \qquad \frac{}{\texttt{SWORD}\,\text{✹—}\,10}$$

$$\frac{\delta\,\text{✹—}\,\alpha}{\texttt{HEADSHOT}[\delta]\,\text{✹—}\,3 * \alpha}\;\mathsf{boom.headshot} \qquad \frac{\delta_1\,\text{✹—}\,\alpha_1 \quad \delta_2\,\text{✹—}\,\alpha_2}{\times[\delta_1, \delta_2]\,\text{✹—}\,\alpha_1 + \alpha_2}\;\mathsf{multiple.dps}$$

2. This rule tells when the amount of dps presently given is not sufficient for reducing the noob to gibs. Applications of this rule are guided by how much dps is in the context.

$$\dfrac{\Gamma \vdash \nu \rightsquigarrow \nu' \quad \delta \text{✹—} \alpha \quad \pi_1\nu' - \alpha > 0}{\Gamma, \delta \vdash \nu \rightsquigarrow \times_n[\pi_1\nu' - \alpha, \pi_2\nu', \ldots, \pi_n\nu']}\ \textsf{not.dead.yet} \qquad \dfrac{}{\epsilon \vdash \nu \rightsquigarrow \nu}\ \textsf{insufficient.dakka}$$

3. This judgment represents the end case in DPS conversion. These rules are similarly inductive on the context.

$$\dfrac{\Gamma \vdash \nu \rightsquigarrow \nu' \quad \delta \text{✹—} \alpha \quad \pi_1\nu' - \alpha \leq 0}{\Gamma, \delta \vdash \nu \downarrow \text{☠}}\ \textsf{lol.pwned} \qquad \dfrac{\Gamma \vdash \nu \downarrow \text{☠}}{\Gamma, \delta \vdash \nu \downarrow \text{☠}}\ \textsf{overkill}$$

3.1 Soundness

Each class of dps obviously makes some sound, proportional to the amount of damage it does. For example, using ROCKET obviously makes a loud explosion, while a HEADSHOT would cause the player to shout out in excitement. Therefore, to ensure soundness, you should do the most DPS possible.

3.2 Completeness

As a wise man once said [1], "Any problem is solvable with sufficient firepower." This serves as an inductive proof that every noob will eventually be reduced to gibs with enough dps.

4 Future Work

The DPS Conversion system as presented only cares about the hp total of noobs. A more advanced system would be able to handle healers and tanks. In the former case, certain abilities could cause the noob's hp to increase in addition to decreasing during a derivation. The latter case makes relevant the other elements of the noob tuple - in addition to hp, a noob might perhaps have spell resistance which would negate DPS from a SPELLCASTER. A new version of DPS would be able to generalize such abilities and include their effects in the conversion process.

5 Conclusion

We provide an example DPS derivation. For simplicity, let noob ::= $\times_1[\texttt{hp}]$. Our noob will start with 50hp. In the first second, a SPELLCASTER gets a HEADSHOT; in the next, a ROCKET and SWORD join the battle, and finally the AWP shoots, though the noob was successfully converted to gibs in the first two seconds.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \vdash \times_1[50] \rightsquigarrow \times_1[50]}\ \textsf{insufficient.dakka} \quad \dfrac{\dfrac{}{\texttt{SPELL}\text{✹—}8}}{\texttt{HS[SPELL]}\text{✹—}24}\ \textsf{boom.headshot}}{\texttt{HS[SPELL]} \vdash \times_1[50] \rightsquigarrow \times_1[26]}\ \textsf{not.dead.yet} \quad \dfrac{\dfrac{}{\texttt{ROCKET}\text{✹—}25} \quad \dfrac{}{\texttt{SWORD}\text{✹—}10}}{\times[\texttt{ROCKET}, \texttt{SWORD}]\text{✹—}35}\ \textsf{multiple.dps}}{\texttt{HS[SPELL]}, \times[\texttt{ROCKET}, \texttt{SWORD}] \vdash \times_1[50] \downarrow \text{☠}}\ \textsf{lol.pwned}}{\texttt{HS[SPELL]}, \times[\texttt{ROCKET}, \texttt{SWORD}], \texttt{AWP} \vdash \times_1[50] \downarrow \text{☠}}\ \textsf{overkill}$$

Acknowledgments: Special thanks to Joshua_ Wise for the insufficient.dakka rule, Nicholas Tan1 for the future work suggestions, and Zachary Z. Sparks for the soundness proof.

6 References

1. Wiseman, A. (22 B.C.) *Any problem is solvable with sufficient firepower.* On archaeologically preserved stone tablets.

Paper 11: DPS Conversion: A New Paradigm in Higher-Order Compilation

Reviewer 1
Rating: 2 (accept)
Confidence: 1/4

The author formulates a sound and complete operational semantics for the reduction of noobs to gibs. While this process is of considerable commercial interest, it has until now been implemented in a rather ad-hoc way, so the paper makes a strong contribution by putting it on firm theoretical footing.

Particular strengths of the paper include its modular proof of soundness, which extends without change to any sound-producing attack (however, this approach does not apply to silent-but-deadly weapons); and also its use of the spooky skull-and-crossbones glyph.

The author could improve the paper by generalizing the result beyond first-person shooters; the extension to person-shooters of other ordinals seems straightforward. Also, I am fairly sure that there is relevant prior workwhich the author has not cited---if I am not wrong Murphy’s work on Generalized Super Mario Bros. and Beckman’s on Infinite Arkanoid are pertinent.

While I have acquired, over the years, certain knowledge of gamer culture, my confidence rating is low because on the whole I am more of a Scrabble sort of guy. Possibly by continuing to review 3L33T G4M3RZ type of papers I will be able to “level up” my confidence in this area. I suppose I couldalso hire them out to a “review farmer” and save myself some carpal tunnel damage.

Track 1
Hypertext

(continued on next page)

Track 1
Hypertext (continued)

DeltaX: A New Paradigm for Machine Translation

Greg Hanneman
Language Enthusiast and Errant Scribbler
Carnegie Mellon University

1 Introduction

The field of machine translation has experienced considerable growth since its original conception in the late 1940s. Over the past 60 years, ~~countless~~ countably finite researchers have developed new techniques and methodologies to advance the state of the art (and the state of their CVs). Sometimes these advancements are significant enough that it can be claimed that a new **paradigm** of machine translation has been developed, though to date there have been very few such breakthroughs.

As you may suspect from the way that last paragraph ended, in this paper we introduce just such a breakthrough. We describe DeltaX, which we believe to be a completely novel paradigm for carrying out machine translation. In contrast to previous methods, DeltaX is fast, cheap, easy, and requires minimal additional hardware beyond what may easily be obtained by ordinary consumer-level technology users.

The remainder of this paper is organized as follows. Since we have no set page-count limit, we first skip a whole bunch of lines just to thumb our nose at other conferences. Next, in Section 2 we briefly review the scientific question of machine translation and summarize its existing paradigms. Section 3 is our *magnum opus*[1], where we describe DeltaX in detail. We follow with Section 4, which performs a comparative analysis between DeltaX and other machine translation methods. Finally, Section 5 concludes.

2 Machine Translation

The problem of **machine translation** is usually expressed as the task of automatically translating text (e.g. a sentence) in one human language (e.g. English) into another human language (e.g. Azerbaijani), such that the semantic meaning of the original sentence is also expressed by the translation. The precise formulation of the task, however, has varied from paradigm to paradigm. Because the mathematical models in these things quickly get complicated, researchers who want their work to be understood by others frequently take recourse in a **generative story**: a simplified conceptual overview of how their systems translate sentences.

2.1 A Generative Story

Once upon a time, there was a man called Warren Weaver. Mr. Weaver was a mathematician, which means that his job was to solve math problems all day. This was a very hard job back when a computer filled an entire room! But Mr. Weaver was a very smart man, and so he enjoyed his job.

One day, Mr. Weaver was walking to his office when he thought of a fun new math problem to solve. “When a man speaks Russian,” Mr. Weaver thought, “what that really means is that he’s speaking English.” (Mr. Weaver had a very good imagination.) “He speaks English in his head, but then the English goes through a secret code and comes out as Russian instead.” Mr. Weaver thought very hard. “Now, all I have to do figure out what that secret code is, and then I can translate the Russian back to English!” Do you remember hearing about secret decoder rings on the radio, where one letter stands for another? Well, all Mr. Weaver had to do was figure out which Russian words stood for which English words in the secret Russian code, and then he would know just like that how to translate Russian.

Mr. Weaver went straight to work. He worked very

[1] Op. 4, in E minor

hard for many days, but the secret code was also very hard. Mr. Weaver was never able to figure it out. He told many other people his idea, but they all laughed at him. At last he stopped working and died. But his story has a happy ending. Even though Mr. Weaver did not find out the secret Russian code, he invented the new problem of **statistical machine translation**. The end.[2]

2.2 Give Yourself Goosebumps

Chapter 1. Hi! My name is **Rule-Based Machine Translation**, but you can call me Rules for short. It totally rules! Anyway, I gotta tell you about my old man — he's such a pain in the neck! I just know that, if you met him on the street, you'd run away screaming. All he talks about is his stupid Russian decoder rings, which is *so* old-fashioned. Can you imagine! Fiddling around all day with his stupid mathematical rules, as if all the wicked cool things about human language could be expressed as mere numbers.

Let me tell you how the modern kids see things these days. We like rules, man, 'cause they totally rule! Just check this out. You know how in French you've got thinks like *la voiture bleue*, right, where the adjective comes after the noun it modifies? But when you say the same thing in English, you've totally gotta reverse the order so the adjective comes first – *the blue car*? Well, that's the whole secret! If we want to translate French into English, we've just gotta know the rules of the game. No numbers, no statistics.

Chapter 2. Now I gotta tell you the best part: learning rules is a whole lot easier than breaking some made-up secret code. We just have to ask people what they are! I mean, you speak English, so you can just tell me what order the words in a sentence have to come in, and then we'll figure out what categories they belong to, and then someone who knows French'll tell us the same information there. We put it all together and, wham! That's machine translation.

Chapter 3. Aw, man, I just got sent to my room by my dad 'cause my homework was late. It was so stupid! All I had to do was figure out a set of rules that would explain all languages, which is like totally reasonable, right? That's what I thought until I started my homework. Do you know how many different meaning there are for some words? Look at this crap: *Can you can fruit in that can?* Three different parts of speech! Then you gotta put all the words together, but if *the man saw the boy with a telescope*, how am I supposed to know who's got the looking glass! Man, this is so unfair! Dad said I gotta stay in my room until I finish parse trees for the garden path sentences, but I think I'm gonna sneak out. There are just too many rules to write down!

2.3 All Grown Up

Good evening, everyone, and thank you for coming to my presentation. Tonight's topic is going to be **phrase-based statistical machine translation**, and I would like to say right at the beginning how pleased I am to be able to speak to you all about it at such an important conference.

Now, as we all know, the field of machine translation has faced some significant problems in the past. We spent a long time paying linguists to write thousands of synchronous grammar rules, morphological analyzers, parsed treebanks, and who knows what else, and at the end of the day none of those techniques were any help. Those processes are just too expensive, and getting good results out of them just takes too long. Who cares whether a word's in the subjuctive aorist passive, or if Arabic is a VSO language? Every time I fire a linguist, my team's translation results go up five points!

Friends, the real answer is data. Data, data, data. We've developed cutting-edge mathematical models to automatically extract information from parallel text — and it's better information than anything a room full of linguists will tell you. In this slide, you can clearly see the way to do translation. Just get a whole bunch of sentences in language F here, and match them up with their translations in language E here. International parliaments are great sources of these, ready-made! Line up all your sentences, then run statistical word alignment to let the math tell you exactly which words in F translate to which words in E exactly which percentage of the time. No made-up linguistics — the numbers don't lie.

On this slide you can see the next step. Once you know how the words translate, just group a bunch of words together, and you know how the phrases translate too. Question, ma'am? No, no linguistics here either. When I say "phrase," I just mean any old group of contiguous words in F that translates into a group of contiguous words in E. Again you can rely on the mathematical models we've developed to give you accurate translation probabilities.

[2] Adults reading this paper in the presence of small children should take care at this point to show the pictures.

Finally, here on Slide 3, you see how easy it is to translate. When you get a new sentence in F, just look through your scored phrase pairs for things that match bits and pieces of the F sentence. String 'em all out in a line, multiply their probabilities, and take the ones that give you the highest score in E. Whatever the data supports the most is the translation you give: let Bayes' Rule do all the work for you, and just maximize that old $P(E \mid F)$.

To conclude, thank you all again for coming tonight, and remember: there's no data like more data!

3 A New Paradigm: DeltaX

3.1 Motivation

In our preliminary work on DeltaX, we were primarily struck by the numerous shortcomings in existing machine translation paradigms. In information-theoretic statistica machine translation, for example, we found ourselves dissatisfied with the reduction of the infinite beauty and creativity of natural language to a mere case of cipher-cracking. On the other hand, the intensive manual effort required to create precise grammars and lexicons in rule-based machine translation leaves us searching for a faster, more automatic method. Finally, we believe that the data-driven techniques of phrase-based statistical translation are reaching a saturation point: while they work well for languages with very large amounts of parallel text, they perform very poorly in the absence of these great resources.

Ideal machine translation has three qualities: (1) it is fully automatic, (2) it is high-quality, and (3) it handles a broad coverage of text styles and domains. It has been observed that, at the present time, one may apply the existing paradigms and obtain any two of these three criteria. This, we believe, is not enough.

3.2 The DeltaX Formulation

We believe that the current limitations of machine translation technology in all existing paradigms come from the fact that the problem is stated incorrectly in all of them. Recall that we previously introduced machine translation as the task of automatically translating a sentence in one human language to a sentence in another human language. In DeltaX, we return to this fundamental and clarifying truth without the distracting probabilistic models or formal grammars of previous approaches. The result is fast, cheap, easy, and requires minimal additional hardware beyond what may easily be obtained by ordinary consumer-level technology users.[3]

Fundamentally, the definition of *translation* is clear:

- "Motion of a body in which every point of the body moves parallel to and the same distance as every other point of the body"

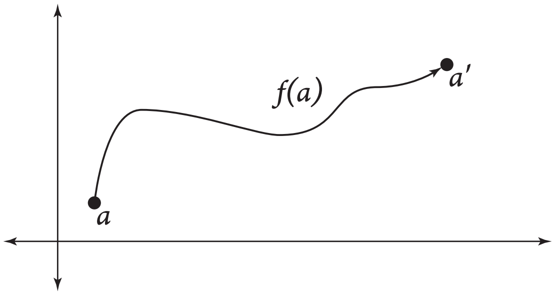


Figure 1: Simple formulation of the translation task.

This is exemplefied in Figure 1. Backing away from morphology, grammars, and parallel data, we frame the task of translation simply: its goal is to find a function $f(a)$ that maps a given point a onto the point a'. Since our starting and ending points are both known, we can easily define this function in terms of the distance between a and a'. We further use the fact that we can re-write point a as the ordered pair (x, y).

$$\begin{aligned} f(a) &= a' - a && (1)\\ f(x,y) &= (x',y') - (x,y) && (2)\\ y' - y &= 0 && (3)\\ f(x) &= x' - x && (4)\\ f(x) &= \Delta x && (5) \end{aligned}$$

We thus arrive at our function for *translation.* To address the *machine* portion, we apply the example-based defintion of "machine" as "computer," as seen in common locutions such as "Linux machine" or "machine room." Putting the two halves together provies a clean, intuitive new paradigm for machine translation, which we fittingly call DeltaX.

[3] Did you notice we stole that sentence almost word for word from the introduction? All the good papers do it.

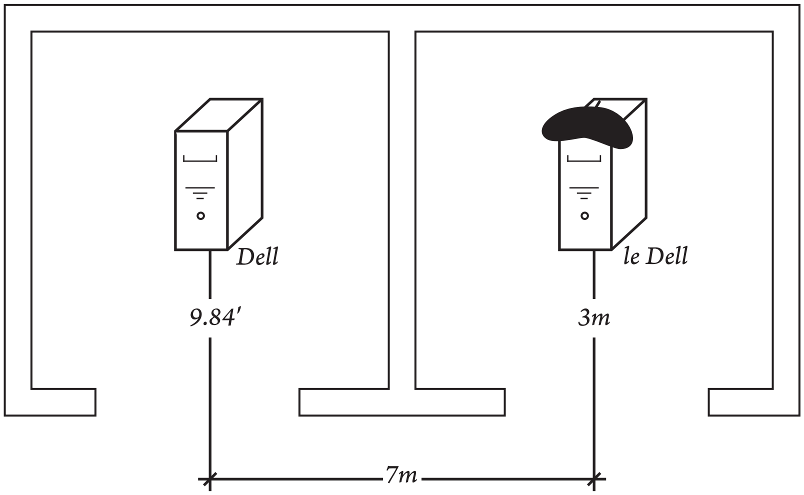


Figure 2: Setup for our English–French machine translation experiments. Our Dell Optiplex input is translated approximately a net 7 meters.

Carrying out the machine translation task with DeltaX is simple, requiring only an input, a goal, and sufficient force — all readily obtainable by today's underfunded and small research groups. For the particularly miserly, the force required may even be calculated in advance with the application of simple physics. We now describe each of the requirements in more detail.

- **Input.** Borrowing from the classical machine translation literature, the input is typically termed the *source*.[4] As a general translation platform, DeltaX places no restriction on the source: it can be any machine — running in any language — the user prefers.

- **Goal.** Classical machine translation uses the term *target* to refer to the output of a translation. Here, we allow the target to also be generic: the user must only pick a goal location that the source machine should be translated to.

- **Force.** Typically, force for a machine translation system is provided by an amount of computing power — either that or an impending deadline, frustrated advisor, or project evaluations. However, many modern systems require large computing clusters, fancy servers, or both. In DeltaX, we relax this requirement, specifying only the use of some amount of motive power. The exact amount may be calculated exactly and in advance.[5]

4 Experiments

To test our system, we carried out experiments on English-to-French machine translation. In each case, our source was a Dell Optiplex GX260. The target was a point in the next machine room over, approximately 15 meters away, and the force was two minimum-wage undergraduates in reasonable physical condition. This setup is shown in Figure 2.

A common metric in machine translation is the BLEU score, a geometric mean of n-gram translation precision compared to a human-produced *reference* translation.[6] Since our system uses neither n-grams nor human-produced references, we find BLEU scores inadequate for our evaluations. Instead, we use an adapted metric that we term CHEESE, or "Computer Here Evaluation, Each System Expected." This easy-to-use metric is based on the target specified during the translation task — if the end result is that the machine has been translated to

[4] Actually, in the Late Classical period, P. M. T. Bach wrote it *y^e olde Sorſe*, but we must make allowances for modern spelling and typography.

[5] This is the second time we have made this statement. The proof and derivation, as in all such cases, are left as an exercise to the reader.

[6] If you didn't understand that last sentence, good luck with "real" research papers. They're all like that!

the target position, CHEESE returns success. If not, CHEESE returns failure status and fires the undergraduate movers ("human translators").

We compare DeltaX to existing machine translation systems freely available on the Internet. Google Translate is a statistical system built from approximately 4.3 quintagazillion words of text in all domains and genres, including some that writers haven't invented yet. Systran (or "Babelfish" or "AltaVista Translate," depending on your age) is a rule-based system made up of context-free grammar rules, begun in 1961 and currently scheduled for completion by human linguists in 2025. To make our comparison fair, we use the same input for all systems. The Dell Optiplex used in the DeltaX system is first converted into a pure text representation by serializing its instruction manual, Verilog files for its processors, and contents of its hard drive. We next pay human translators (in the classical sense) to translate the resulting text into French; this provides a reference translation for use with the BLEU metric.

System	**Metric Score**	
Google Translate	0.0442	BLEU
Systran	0.0657	BLEU
DeltaX	1.0000	CHEESE

Figure 3: Results of experiments comparing the DeltaX machine translation system to two Internet competitors.

Results are shown in Figure 3. Surprisingly, the rule-based system outperforms the statistical system in our machine domain. We hypothesise that this is true because machine translation — as we have expressed it — can be succinctly expressed a series of generally applicable rules, something that rule-based systems excel at. Neither of the two Internet-based machine translations, however, hit upon the correct series of rules to apply in our experimental setup. The DeltaX system, on the other hand, was able to express the translation task as a series of three rule applications:

1. ~~Never talk about machine translation.~~

2. Translate south by three meters.

3. Translate east by seven meters.

4. Translate north by three meters.

By following this simple procedure, the DeltaX system is able to achieve a perfect 1.0 on the CHEESE metric, while the competing systems fail to measure up in BLEU.

5 Conclusion

.sedulcnoc 5 noitceS ,yllaniF .sdohtem noitalsnart enihcam rehto dna XatleD neewteb sisylana evitarapmoc a smrofrep hcihw ,4 noitceS htiw wollof eW .liated ni XatleD ebircsed ew erehw ,*supo mungam* ruo si 3 noitceS .smgidarap gnitsixe sti ezirammus dna noitalsnart enihcam fo noitseuq cifitneics eht weiver ylfeirb ew 2 noitceS ni ,txeN .secnerefnoc rehto ta eson ruo bmuht ot tsuj senil fo hcnub elohw a piks tsrif ew ,timil tnuoc-egap tes on evah ew ecniS .swollof sa dezinagro si repap siht fo redniamer ehT

.sresu ygolonhcet level-remusnoc yranidro yb deniatbo eb ylisae yam tahw dnoyeb erawdrah lanoitidda laminim seriuqer dna ,ysae ,paehc ,tsaf si XatleD ,sdohtem suoiverp ot tsartnoc nI .noitalsnart enihcam tuo gniyrrac rof mgidarap levon yletelpmoc a eb ot eveileb ew hcihw ,XatleD ebircsed eW .hguorhtkaerb a hcus tsuj ecudortni ew repap siht ni ,dedne hpargarap tsal taht yaw eht morf tcepsus yam uoy sA

.shguorhtkaerb hcus wef yrev neeb evah ereht etad ot hguoht ,depoleved neeb sah noitalsnart enihcam fo **mgidarap** wen a taht demialc eb nac ti taht hguone tnacifingis era stnemecnavda eseht semitemoS .)sVC rieht fo etats eht dna(tra eht fo etats eht ecnavda ot seigolodohtem dna seuqinhcet wen depoleved evah srehcraeser etinif ylbatnuoc ~~sseltnuoc~~ ,sraey 06 tsap eht revO .s0491 etal eht ni noitpecnoc lanigiro sti ecnis htworg elbaredisnoc decneirepxe sah noitalsnart enihcam fo dleif ehT

Well, even high school students know that the conclusion section is just the introduction section written backwards.

Acknowledgements

The authors — even though there is only one of us — would like to thank Lea Albaugh ("B-A-U-G-H, Pennsylvania Dutch–style") for her fine figure drawing contributions to the present paper. The research reported here was supported by borrowed time from the Language Tecnologies Institute at Carnegie Mellon University when the author was supposed to be writing his thesis proposal. Conclusions drawn from

it represent the author's own artwork and do not represent the opinions of real artists who actually know how to use charcoal sticks and pencils.

References

[1]	Perl	`\$scalar`
[2]	C/C++	`int* numKumquats`
[3]	Java	`Object o`

System Description: Google Ghost

Robert J. Simmons and Brian R. Hirshman
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{rjsimmon,hirshman}@cs.cmu.edu

Abstract

This paper describes the new craze sweeping through computer science departments located along Forbes Avenue in Pittsburgh, PA, USA: playing Google Ghost. This paper describes the static and operational semantics of Google Ghost, and provides a constructive proof that you[1] can become a Google Scholar!

1 Introduction & Motivation

Google Ghost is a brilliant new adaptation of the deprecated, archaic children's game Ghost. A wise horde once said:

> Ghost is a word game in which players take turns adding letters to a growing word fragment, trying not to be the one to complete a valid word. If a player completes a word, they lose that round of the game and starts a new round. Each fragment must be the beginning of an actual word. Usually some minimum is set on the length of a word that counts, such as three or four letters.[2]

We identify several fundamental problems with the deprechaic version of Ghost:

- **Boredom**: Ghost is boring as crap.
- **Difficulty**: Oh, are you done with your New York Times crossword puzzle, *nerd?*
- **Speed**: Ghost tends to require lots of thought before a turn is played, as the player tries to both guess the word the other players were trying to spell and try to (potentially) subvert their intent by redirecting to another word without completing the original word.
- **Size**: The creative possibilities of Ghost are limited by the people in the room with you. And good luck getting people to play your boring, hard, slow game, *nerd.*

These problems are all solved by Google Ghost, which is awesome.

[1] Yes, you.

[2] http://en.wikipedia.org/wiki/Ghost_(game)

2 Google Ghost

Google Ghost is a multiplayer game that is played by using, instead of boring English words, the massive database of mapreduced by Google Suggest.[3] To play Google Ghost, you need two or more *players* and one *ghostwriter*[4] who has access to a computer running a modern, JavaScript-enabled web browser. Play starts with the player to the right of the previous loser utters a word (either an English word or common, proper noun) and then either "space" or "no space." The ghostwriter types that word into Google's home page and then, as instructed, either enters a space or not. Play proceeds by each player to the right uttering either a letter or the character "space" – two spaces may not be played in a row.

We chart the progress of a game by using a four-column chart. The first column indicates the player (we use standard cryptographic characters Alice, Bob, etc.) The second column indicates the player's utterance, the third column indicates the resulting contents of the search bar, and the fourth column indicates some of the contents of the Google Suggest list:

A	”horrible space”	horrible	horriblesubs, horrible tattoos, horrible histories…
B	”i”	horrible i	horrible insults, horrible in spanish, horrible injuries…
A	”n”	horrible in	horrible insults, horrible in spanish, horrible injuries…
B	”space”	horrible in	horrible in spanish, horrible in french, horrible in laws…
A	”l”	horrible in l	horrible in laws, horrible in latin, horrible in law stories
B	…	…	…

Google Ghost is made more interesting by the presence of spell checking, as shown by the presence of "ikea" as the first hit after Alice's second turn:

A	”i, no space”	i	imdb, ikea, itunes, ipad, irs, itunes download, inspirational quotes…
B	”e”	ie	ieee, ie7 download, ie8, ie8 download, iep, ie7, iexplore.exe, ies…
A	”k”	iek	ikea, iekeliene stange, ieko media, iekeliene stange fashion spot…
B	…	…	…

For extra entertainment value, the ghostwriter can point out recently excluded options, for instance that "horrible injuries" and "horrible insults" had been excluded after Bob uttered "space" in the first game above.

2.1 Losing and ggpwning

You normally lose the game when there are no more auto-complete options available after your selected letter, number, or space has been typed into the Google bar. This is the standard way of losing, as demonstrated by the following game which ends with a normal loss for Bob.

A	”arby's space”	arby's	arby's menu, arby's coupons, arby's nutrition…
B	”q”	arby's q	arby's queens, arby's quebec, baby mamma arby's quote
A	”u”	arby's qu	arby's queens, arby's quebec, baby mamma arby's quote
B	”o”	arby's quo	baby mamma arby's quote
A	”t”	arby's quot	baby mamma arby's quote
B	”a”	arby's quota	*None. Bob has lost in regular play.*

However, play can also end abnormally with the rare epic win of Google Ghost technically known as a *ggpwn*. Ggpwn is normally used as a verb – ie, "I ggpwned you guys so hard."

A ggpwn occurs when the final letter that eliminated all auto-complete options *is* one of the options that was provided by Google Suggest. A *non-strict* ggpwn occurs if the letter

[3] http://autocompleteme.com/

[4] http://en.wikipedia.org/wiki/Ghostwriter_(TV_series)

closes out an auto-spellchecked or partial suggestion; for instance, in the demo game above if Bob had picked "e" instead "a" in the final turn, he would have achieved a non-strict ggpwn.[5] A *strict* ggpwn is created when the phrase *exactly* matches one of the suggestions. For instance, Alice would achieve a strict ggpwn by uttering the final "y" in the search query "why is my poop green pregnancy."

Players should determine at the beginning whether the play is strict or non-strict; in strict play a non-strict ggpwn is the same as losing. For example, in the game above Bob never had a chance above in strict play; although if he had uttered "e" instead of "o" Alice could have strictly ggpwned Bob by exactly completing the phrase "arby's queens" or "arby's quebec."

3 Google Ghost Scholar (AKA Google Scholar Ghost)

Google Ghost Scholar is clearly "the next big thing". Actually, it's really one of the next three big things, as it can mean three different things. How cool is that?

- It can mean Google Ghost played with Google Scholar. Open up Google scholar and play Google Ghost as described previously. (Aren't you kicking yourself now if you missed that SIGBOVIK demo?) The first person to go, of course, will find that they lose because Google hasn't put autocomplete into the public version of Google scholar. Hint: if you challenge someone to this game, make sure you – and not your oppwnent – go second!
- It can also mean a single player game of Google Ghost played in support of the multiplayer variants. Begin playing this version of Google Ghost Scholar by creating web pages and make them link to nonsense words. Make sure you set up the robots.txt in such a way that Google indexes them appropriately. Alternately, Google bomb other pages with nonsense terms in such a way that you can be sure that Google autocomplete will recognize those terms when you play. Congrats, now you can ggpwn your oppwnents!
- Google Ghost Scholar is also a noun meaning one who can ggpwn his or her oppwnents at any variant of Google Ghost. It is often abbreviated with the three-letter acronym "GGS" and can be applied as a titular suffix after a name. For instance, the first author, when[6] he finishes his doctoral degree, will be Rob Simmons Ph.D. GGS.

4 Conclusion

Google Ghost overcomes all the problems with the deprechaic original version of Ghost. It has been shown to be empirically exciting, and it is as easy as typing nonsense into the Google Toolbar, which easy enough that any idiot seems able to do it, as evidenced by the fact that enough people searched for "how did I get this cold sore" for it to end up in Google Suggest. The ability of the first player to suggest a whole word makes the game significantly faster-paced, and the creative possibilities for humor in the game are limited only by the thought droppings of millions and millions and millions of people typing stuff into Google.

We acknowledge the Carnegie Mellon Computer Science Department's Tea Trolls and Google, Inc. for their respective contributions to the existance of Google Ghost, and we furthermore acknowledge Chris Martens as the first person to ever ggpwn.

[5] But Bob didn't do that, did he? Bob is a loser.

[6] Right. *When.*

The Next 700 Intuitionistic Linear Logics

Lea Albaugh William Lovas Tom Murphy VII Jason Reed

March 20, 2010

Abstract

We describe, and lay claim to, the next 700 intuitionistic linear logics. Hands off! We saw 'em first!

1 Introduction

Peter Landin started a dangerous trend with his seminal paper, "The Next 700 Programming Languages" [Lan66]. How, we ask, are we all supposed to pad our CVs (ideally, with research sliced so finely it qualifies as "deli-thin") if some joker comes along and comes up with one stupid, elegant generalization that makes all of the many delicious slices mere special cases? Where our score in the publication game was once well in the hundreds, now it is one. This is a *very small score*. We attempt to reverse this trend by not obviating the next 700 intuitionistic linear logics, but by *listing* them.

As everyone knows, the most important properties of a logic are

1. That it satisfies cut and identity properties, and has a complete focusing proof search strategy.
2. That it is intuitionistic, so that we can feel superior to those filthy classical mathematicians.
3. That it has linearity, so that we can use funny symbols like $\multimap$ and $\otimes$ and $\parr$.

   and, most important of all,

4. That its name forms a clever acroynm.

Bearing this in mind, and observing that we already got beaten to the punch by ILL [Gir87] and DILL [BP96] and JILL [CCP03], we make haste to lay claim to all of the following logics.

2 The Logics

2.1

Name: BILLG
Description:
Billionaire ILL, which is Good. A rich and expressive logic. Well, rich, anyway. Seems to be incompatible with MOZILLA.

2.2

Name: CHILL
Description:
Calm, Hushed ILL. Won't wake the neighbors. Very mellow. Won't freak out if you, like, prove a contradiction, man.

2.3

Name: DRILL
Description:
We're still waiting for legislation to pass let us start using this logic to extract our nation's vast untapped natural resources. Given the current political climate, we're not exactly holding our breath.

2.4

Name: FILL
Description:
Fudgmental ILL. Mmm..

2.5

Name: LANDFILL
Description:
The trashiest logic yet.

2.6

Name: GILL
Description:
Goldfish ILL. Easy to take care of.

2.7

Name: GRILL
Description:
The logic of food specialized to flame-broiling.

2.8

Name: HILL
Description:
Himalayan ILL. Alive with the sound of music?

2.9

Name: ILLIAD
Description:
Truly, a logic of epic proportions.

2.10

Name: KILL
Description:
Kalashnikov ILL. Can prove 600 sequents per minute without overheating.

2.11

Name: OVERKILL
Description:
Can prove *6,000* sequents per minute without overheating. Nobody seems to need to prove sequents this fast, however. Still, what kind of logicians would we be if we paid attention to what people said was practically useful?

2.12

Name: KRILL
Description:
I don't know, man, whales eat it?

2.13

Name: MANDRILL
The largest monkey-logic in the world. Has a colorful rump.

2.14

Name: MILL
Description:
Maize ILL. Grindin' it up..

2.15

Name: MOZILLA
Description:
The intuitionistic logic for describing how web-browsing consumes your life. Compatible with many useful extensions, such as Ad-Block.

2.16

Name: GODZILLA
Description:
An ill-fated attempt to design MOZILLA. Destroyed half of Tokyo, and Martin-Löf still has nightmares to this day.

2.17

Name: PILL
Description:
Pharmaceutical ILL. Probably not patented yet.

2.18

Name: QuILL
Description:
Quail-feather ILL. Old-school theorem writin'.

2.19

Name: FRILL
Description:
I feel pretty! And witty! And... NP-complete assuming the Unique Games Conjecture?

2.20

Name: NOFRILL
Description:
Just what it sounds like.

2.21

Name: SPILL
Description:
Oh, not *again*. My pants are just *covered* in Γs and $\vdash$s. Do you realize how hard it is to get those stains out? Also a good logic for reasoning about register allocation.

2.22

Name: SQUILL
Description:
Useful for reasoning about Pittsburgh neighborhoods, which can otherwise be quite difficult to navigate.

2.23

Name: STILL
Description:
Scotch-Tasting ILL, the logic of whiskey.

2.24

Name: SHILL
Description:
A fake logic planted to con other logics into intuitionism and linearity.

2.25

Name: SWILL
Description:
The logic of drink, but only crappy ones.

2.26

Name: THRILL
Description:
Just the logic you want when it's close to midnight and something evil's lurking in the dark — simply prove it's not there!

2.27

Name: TRILL
Description:
The logic of musical ornaments.

2.28

Name: WILL
Description:
A logic to leave to your descendants. Of sound mind and body. Also of complete mind and body.

2.29

Name: ILLWILL
Description:
The doubly intuitionistic doubly-linear logic of malice.

2.30 Lax Linear Local Longitudinal Latitudinal Lambda-Lifted Lambda-Long Looping Left-Ludical Linked Leibniz Law L-system Logspace Logtime LL-1 Lookup Lined Lex Limiting L'Hopitalian Laplacian Lagrangian Legendre Ladder-Like Lattice Lemmaless Lambek Typed Super Kinded Intuitionistic Linear Logic

Name: L33T SKILL
Description:
A very powerful logic by Tom 7 Murphy[7] (the third author) and the second author (William Lovas[1]).

2.31

Name: SKILL
Description:
This one's okay, but not nearly as good as the one just before it.

3 Conclusion

Okay, okay, so we don't quite have 700 yet. We leave the remaining logics for Future Work.

References

[BP96] A. Barber and G. Plotkin. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, University of Edinburgh, Laboratory for Foundation of Computer Science, 1996.

[CCP03] Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131, Carnegie Mellon University, 2003.

[Gir87] J.Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Lan66] P. J. Landin. The next 700 programming languages. *Commun. ACM*, 9(3):157–166, March 1966.

[7] Dr. _ VII, Ph.D

[1] the one, the only[2]

[2] you know, the one whose name is so awesome it frequently bursts into flame[3]

[3] `http://www.cs.cmu.edu/~wlovas/old-index.html`

Harnessing Human Computation: β-REDUCTION HERO

Akiva Leffert

Abstract

Most ACH SIGBOVIK papers have an abstract in order to draw the reader the in and to set up future jokes. In this recessionary time, jokes are a scarce and valuable resource. We prefer not to waste them on a moment of brief glory, when we could drag them out for whole paragaphs and sections. We apologize for the inconvenience and hope you will bear with us during this time of economic difficulty.

1 Introduction

Human computation[2] is an upcoming and trendy branch of computer science. Human computation allows computers to solve problems that were previously difficult for computers to solve by the simple expedient of making humans do it for them. This is widely considered to a major breakthrough. Typically humans are compensated for their computational cycles either through piddling amounts of entertainment (The ESP Game), piddling amounts of money (Mechanical Turk)[7], or via brainwashing[6]. The first technique is problematic because it requires a way to frame the task being solved as a game. The second is a problem because it requires funding. The third has proved difficult to implement with existing technologies. Furthermore, human computation throws the best parties and we want invites! In the remainder of this paper, we resolve these problems through the introduction of the novel notion of the Hero Quotient and describe a technique to reduce any computational task to a single Game With a Purpose.

2 Take 1 Cup of Heroism

Our first critical insight is that anything sounds more fun when it has the word “hero” at the end[3][1]. We conducted an extensive survey (n=2) to

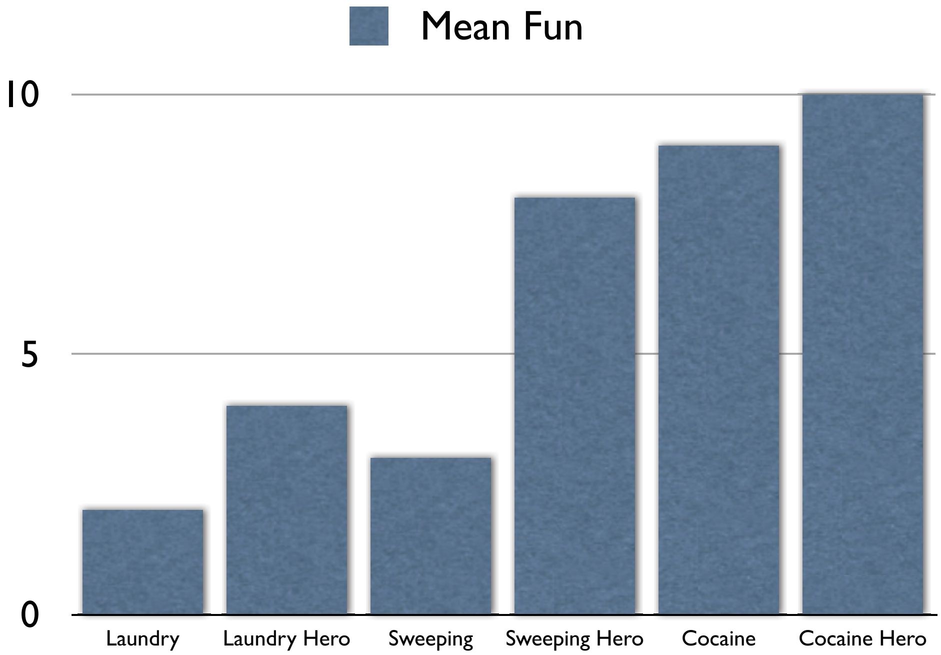


Figure 1: Sample Hero Quotient Effects

determine the so called ‘Hero Quotient’ (HQ), the amount a thing becomes more fun when followed by the word “hero.” Figure 1 shows the effect size for several of the items in our survey. The x-axis is the item being surveyed and the y-axis is the mean fun score on a scale of 1 to 10. Overall, we found that the mean HQ multiplicative factor across all of our 40 data points was 2.8 ($p < .05$). An exhaustive list of our survey items and mean fun ratings appears in the Appendix.

3 Add 3 Parts λ-Calculus

The early history of computer science is rife with attempts to find a universal notion of computation. Objectively the best is Church’s λ-calculus[5]. Hopefully you already know what the lambda calculus looks like, but we include it here because it’s awesome. The entire syntax and reduction rules of the lambda calculus are as follows:

$$e \quad ::= \quad x \mid ee' \mid \lambda x.e$$

$$(\lambda x.e)e' \quad \to_\beta \quad [e'/x]e$$

The main reduction rule is known is β reduction for no particularly good reason. Note that we use the standard notion of capture-avoiding substitution. This will be important later.

The normalization problem for the lambda calculus is computationally universal. And indeed, by the Church-Turing thesis, it is hypothesized that any notion of computation can be reduced to the question of whether some λ-calculus term has a normal form.

4 Stir until frothy

Our second brilliant insight is that we can combine the excitement enhancing effect (EEE) of the HQ with the computation universality of the λ-calculus. This allows us to turn any computational task t into a TOTALLY RAD TIME$^{\text{TM}}$simply by compiling it into raw *lambda*-Calculus.

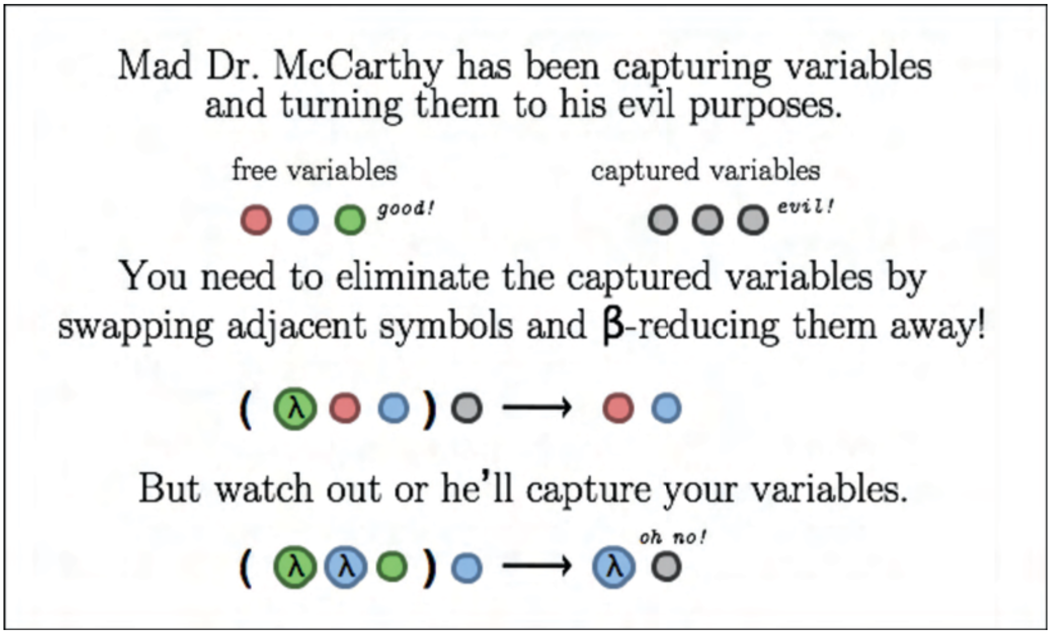
We took this approach and as a proof of concept developed a Game With a Purpose called β-REDUCTION HERO.

5 Sautè five minutes

β-REDUCTION HERO is an Adobe FLASH[**?**] game, developed in HAXE[4], a strongly typed language with first class functions, paremetric polymorphism, type inference, and structural object types that compiles down to several targets including the FLASH platform. The basic game play of β-REDUCTION HERO is illustrated in Figure 2, a shot of the game's help screen.

While our basic goal was merely to harness the power of human computation toward arbitrary tasks, we must admit to a more humanitarian goal: Raising awareness of the problem of captured variables. One in every one hundred american variables is behind bars. In β-REDUCTION HERO your goal is to release captured variables, by forming β-redices.

The basic game mechanic resembles the popular game Bejeweled. Adjacent tiles can be swapped by clicking one then the other. Instead of the usual named variables, we use the colors red, green, and blue. We also replace



Mad Dr. McCarthy has been capturing variables and turning them to his evil purposes.

You need to eliminate the captured variables by swapping adjacent symbols and β-reducing them away!

But watch out or he'll capture your variables.

Figure 2: Rules of Beta-Reduction Hero

the syntactic sequence $\lambda x.$ with a single token, a lambda in a colored circle, the color corresponding to the variable bound by the lambda. This innovation allows us to conserve screen real estate, as well as harkening back to Church's original notation for the lambda calculus where the bound variable and its binder were coupled in typographically.

After each swap, the rows affected are searched for β-redices and the longest one in each row is reduced. This is where captured variables come into play. Any variable that is captured in a reduction is turned gray to signify its capture. The goal of the game is to eliminate all of the captured variables. Thus the player must construct redices by hand.

In theory our technique could be adapted to different game play styles. We recently began cooperative work with researchers at Nintendo of America on adapting our technique into Mario Teaches λ-Calculus. An early screen appears in Figure 3.

6 Serve Hot

β-reduction is pretty cool. β-reduction hero is super cool. Check it out: $\texttt{http : //akivaleffert.com/beta − reduction − hero/}$.

Figure 3: MARIO TEACHES λ-CALCULUS Concept Art

A Garnish

Item	Normal	With Hero
Algebra	9	10
Bird Watching	1	2
Cocaine	9	10
Debugging	3	7
Dish washing	9	10
Guitar	1	11
Glockenspiel	7	10
Golf	2	2
Laundry	2	4
Mitosis	4	9
Meiosis	6	8
Opera	4	9
Problem Sets	1	4
Sweeping	3	8
Zen	5	

Figure 4: Complete Survey Means

References

[1] Activision. Knitting hero, 2009.

[2] Luis Von Ahn. Some paper luis wrote. *Proceedings of the ACH SIGCHI*, 2004.

[3] Harmonix. Guitar hero, 2005.

[4] haXe. haxe.org, 2009.

[5] Akiva Leffert. The letter before lambda is hat. *Proceedings of the ACH Conference/Robot Dance Party 0x40th Celebration of Harry Quicktopher Bovik*, 2007.

[6] Akiva Leffert. Provably sound orbital mind control lasers. *Proceedings of the ACH Conference/Robot Dance Party/Anniversary of the 0x40th Celebration of Harry Querulous Bovik*, 2008.

[7] Amazon Mechanical Turk. mturk.com, 2007.

sigbovik 2009 Plenary Program Committee Confidential Paper Reviews

Paper 7: Beta-Reduction Hero

Reviewer 1
Rating: 3 (strong accept)
Confidence: 3/4

This program easily meets SIGBOVIK's exacting standards. Because its interactive format, there is some question about how it should be published--perhaps the only reasonable solution is to publish the URL of the "game" in SIGBOVIK's proceedings.

Reviewer 2
Rating: 3 (strong accept)
Confidence: 4/4

I may not know anything about the lambda calculus, but I do know about circles. I do know that lambda isn't drawn like a circle. "O" is drawn like a circle. However, this paper has a lot of colored circles in it. I'm not sure what they are doing, or how I should manipulate them. But I'm also not sure of the mechanics of breathing and that seems to work out most of the time. Except that one time when I was in some water or something. Fucking hilarious. But back to the paper.

I am pretty sure that the fonts are broken at present.

But that's okay, because the fonts would just get in the way of the colored circles. Did I mention that the circles come in three exciting colors? There's, like, this reddish one and maybe a blue one and a green one. And, um, maybe yellow? Well, there should probably be yellow. It's the sort of thing there should generally probably be in these sorts of papers.

The authors should consider including a color chart to make it easier for me to write the prevous paragraph.

Unfortunately, I am unable to give this paper a higher rating because I couldn't determine how many pages it had, and I rate papers entirely based on the number of pages. I foresee this also being troublesome in the printing process, as it will be hard for the proceedings manufacturer to quote a price on an indefinite number of pages.

Also, the authors should include page numbers.

Finally, I advise the authors to consider other venues for this paper if publication in SIGBOVIK proves infeasible. With a bit of cabinetry, I could see the work fitting in well in the proceedings of Metreon Portal One, Golf-n-games, or Dave and Buster's. And it could well find a larger audience in these locations.

The Dissatisfactory Doily

Generating Your Own Terrible Towel Rip-Off

Nels E. Beckman

Carnegie Mellon University

nbeckman@cs.cmu.edu

Abstract

This paper presents a simple tool that enables sports teams across the globe to create their own rip-off of the Terrible Towel. We describe the tool and some of the clever names it generates.

Categories and Subject Descriptors K.4.4 [*Electronic Commerce*]: Intellectual property

General Terms Legal Aspects, Human Factors

Keywords terror, linens

1. Introduction

Since 1975, when local sportscaster Myron Cope created the Terrible Towel, sports teams all around the world have been green with envy. These Team owners could not believe that such as simple device could engender such enthusiasm. Most importantly, they could not believe that such a simple, low-quality cloth, costing no more than twenty or thirty cents, could be sold for ten dollars.

Ever since, the owners have been in a bind. While it would be quite trivial to create a rip-off cloth in the theme of their own team, and sell it and an outrageous markup, without a suitable clever name it is extremely unlikely that it would sell. For thirty-five years, this question has stumped mankind.

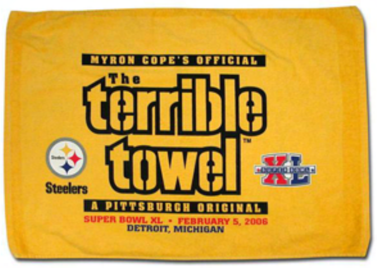


Figure 1. Myron Cope's Official Terrible Towel

2. Dissatisfactory Doily

Enter the Dissatisfactory Doily: a computer program that can automatically generate, a clever, alliterative name for your Terrible Towel rip-off. Built on proprietary Java technology, this program contains a patented *algorithm* that, with little or no human input can generate a mountain of humorous names.

Armed with such a humorous name, you can emblazon it on a cloth of cheap quality, sell it to your beer-guzzling fans, and make an even more obscene amount of money at the expense of local schools. What's more, since the name is humorous, you can claim parody use, and not fear trademark claims.

The operation of the Dissatisfactory Doily is simple. Run the tool at the command line, and pass along a list of adjectives (negative ones preferred), a list of nouns (ideally cloth-related), and the number of names that should be generated. For example:

```
java DissDoily adj.txt noun.txt 1000
```

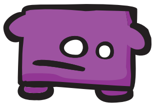
In the following section, we present a list of towel names generated by the Dissatisfactory Doily and ready for immediate use. Our favorites are highlighted in bold.

SIGBOVIK '010 Pittsburgh, Penna, USA-A-OK

3. Examples

Feel the fear wash into the hearts of your opponents as your fans wave the...

The villainous veil! The raunchy rag! The desperate drape! **The atrocious afghan!** The depraving drapery! The disgracing doily! The grungy gear! The fallacious film! The malevolent materiel! **The wanton wipe!** The blah bolt! The grungy goods! The opprobrious outfit! The disgraceful dry goods! The icky ingredient! The gross goods! The horrid hanging! The cruddy cloak! The poor paraphernalia! The poor portiere! The dreaded doily! The heinous hanging! The odious oleo! The dreaded decoration! The scandalous shade! The poor paraphernalia! The slipshod shutter! **The corrupt comforter!** The vile valance! The dreadful decoration! **The malevolent moist towelette!** The severe screen! The repulsive runner! The sinful screen! The opprobrious object! The obnoxious oleo! The dissatisfactory drape! The dreadful drapery! The poor portiere! The criminal covering! The contumelious calico! The scandalous stock! The bummer bolt! The degenerate doily! **The repulsive roller!** The disgraceful drapery! The imperfect ingredient! The opprobrious object! The harrowing hanging! The desperate drape! The fallacious fleece! **The heinous habiliments!** The iniquitous ingredient! The substandard shutter! The sad screen! The disgraceful dry goods! The horrifying hanging! The synthetic supply! The iniquitous individual! The gross goods! The scandalous serviette! The stinking serviette! The monstrous moist towelette! The diddly doily! The egregious equipment! The monstrous moist towelette! The awful afghan! The beastly blind! The dissatisfactory dry goods! **The contumelious carpeting!** The debauching decoration! The shocking shroud! The beastly being! The icky ingredient! The faulty fleece! The junky jalousie! The careless cloth! The villainous valance! The synthetic shag! The wanton wrapper! The diddly drape! The egregious envelope! The horrendous habiliments! The synthetic synthetics! The opprobrious object! The malevolent matting! The monstrous moist towelette! **The fallacious fleece!** The dreadful dry goods! The pits puff! The dissatisfactory decoration! The wanton weave! The wanton wipe! The debasing drape! The flagitious film! The frightful film! The repulsive rug! The fallacious fleece! The stinking shag! **The disturbing doily!** The loathsome layer! The ghastly goods! The poor paraphernalia! The vile veil! The pits portiere! The revolting rug! The stinking synthetics! The flagitious floor covering! The disgracing doily! The atrocious afghan! The disturbing doily! The inferior ingredient! The gross gear! The diddly doily! The horrid hanging! The obnoxious outfit! The sad stuff! The ghastly goods! **The brazen bolt!** The abhorrent afghan! The garbage goods! The gruesome goods! The loathsome layer! The amiss afghan! The inhuman ingredient! The inadequate ingredient! The hateful habiliments! The petrifying puff! The monstrous moist towelette! The sad screen! The abominable afghan! The bottom out bolt! The horrible hanging! The wanton wipe! The defective doily! The grody goods! The brazen bolt! The inferior ingredient! The contumelious carpeting! The dreadful doily! The crummy constituent! The horrendous hanging! The brazen being! The garbage goods! The vile veil! The stinking screen! The cheesy coat! The imperfect individual! The erroneous envelope! The heinous hanging! The crappy covering! The inconvenient ingredient! The atrocious afghan! The wicked wipe! **The sad serviette!** The petrifying puff! The corrupt coverlet! The debauching doily! The awesome afghan! The egregious entity! The imperfect individual! The monstrous mat! The gross gear! The dread decoration! The loathsome layer! **The monstrous mat!** The shameless substance! The fallacious film! The beastly blind! The gruesome gear! The disastrous drapery! The appalling afghan! The iniquitous ingredient! The shameless shroud! The revolting rug! The deficient drapery! The harrowing habiliments! **The scandalous shroud!** The appalling afghan! The beastly being! The abhorrent afghan! **The wanton wall-to-wall carpeting!** The egregious entity! The awesome afghan! The garbage goods! The malevolent mat! The dire doily! The odious object! The abhorrent afghan! The grody goods! The raunchy roller! The horrendous habiliments! The sinful supply! The rotten rug! **The monstrous material!** The stinking shroud! The imperfect individual! The cheap comforter! The grody gear! The grody goods! The dangerous drapery! The inferior ingredient! The severe shutter! The harrowing hanging! The awful afghan! The blah body! The abhorrent afghan! The slipshod screen! The blah bolt! The rough rug! The violent veil! The fearful floor covering! The dread decoration! The villainous valance! The horrifying hanging! The serious stuff! The inconvenient ingredient! The malevolent moist towelette! The stinking screen! **The crappy cloth!** The garbage gear! The odious object! The repulsive runner!



You Keep Dying

Dr. Tom Murphy VII Ph.D.

1 April 2010

Abstract

`http://youkeepdying.spacebar.org/`

KEYWORDS Use the arrow controls. The space button does jump.

You Keep Dying

You keep became infected by the virus and you keep dying   You keep combustible and you keep dying   You keep radiation exposure and you keep dying

You keep proximity to the explosion and you keep dying   You keep plasma bolt irritates the skin and you keep dying   You keep fall off the screen and you keep dying   You keep spider bite and you keep dying   You keep investigate the disturbance and you keep dying   You keep internal bleeding and you keep dying

You keep touch these words and you keep dying You keep wonder what is that thing and you keep dying   You keep can't breathe under water and you keep dying   You keep high velocity and you keep dying   You keep pressed the wrong buttons and you keep dying

You keep go into this ↑ impossible room and you keep dying   You keep incorrect solution to the

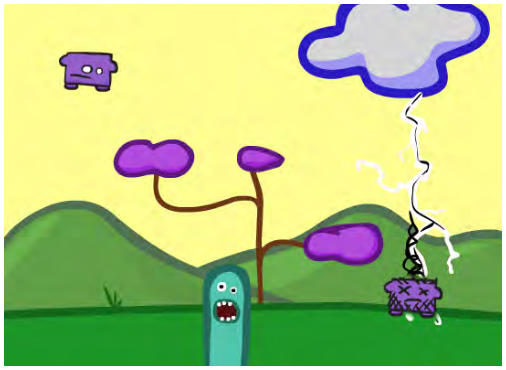


Figure 1: You keep conduct the lightning and you keep dying

-1

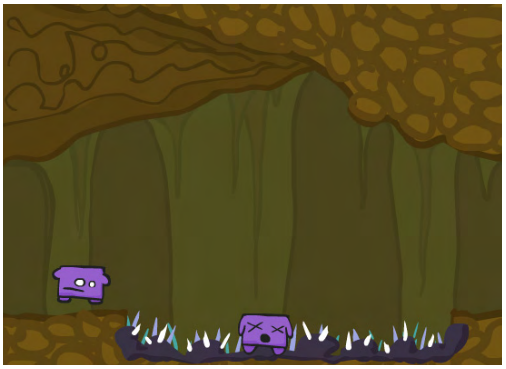


Figure 2: You keep hug the spikes and you keep dying

puzzle and you keep dying You keep revise and resubmit and you keep dying You keep thought you could jump farther and you keep dying

You keep didn't realize the infection could be spread and you keep dying You keep thought that it was just background graphics and you keep dying You keep bring the copysicle everywhere and you keep dying You keep acid dissolved you and you keep dying You keep stand still too long and you keep dying You keep fight or flight response and you keep dying You keep illegal security system and you keep dying

You keep treat the monster like a friend/neighbor and you keep dying You keep starting over and you keep dying You keep to find all the spawn spots and you keep dying You keep a rock fell on you and you keep dying You keep wasn't carrying the right item and you don't know what the right item is and you keep dying

You keep overfull hbox and you keep dying You keep name brands at discount prices and you keep dying You keep go back to get the umbrella again and you keep dying You keep getting out of the frying pan and you keep dying You keep finding the well-hidden trap and you keep dying You keep disappointed in yourself and you keep dying

You keep physics bug and you keep dying You keep searching for cheat codes and you keep dying You keep commuting to work and you keep dying You keep invisible ultraviolet laser and you keep dying You keep paying the monthly fee and you keep dying You keep smoke tobacco and you keep dying

You keep techno beats and you keep dying You keep accomplishment unlocked and you keep dying You keep heamophilia and you keep dying You keep try to save your game and you keep dying You keep boss fight and you keep dying You keep was smaller than the foot and you keep dying You keep not getting the hang of it and you keep dying You keep unusual accident and you keep dying

You keep perforated and you keep dying You keep a tight ship and you keep dying You keep crushed when the ceiling did that thing unexpectedly and you keep dying You keep swarm of bees and you keep dying You keep try to get down there and you keep dying

You keep forgot that one detonates and you keep dying You keep stay out of the dark and you keep dying You keep wondering if you should see a doctor and you keep dying

You keep thank the anonymous referees and you keep dying You keep playing and you keep dying You keep `youkeepdying.spacebar.org` and you keep dying

Human Computation Method for Generating High Impact Research Papers

Laura Trutoiu Erik Zawadzki Leigh Ann Sudol J Marlow
Anonymous Coward Ekaterina Taralova Field Cady
Matt Stanton Ciera Jaspan Or Sheffet
Harry Q. Bovik

First line! W00t!

Abstract

Research papers are an important tool to disseminate scientific knowledge. Indeed, it has been argued by Smith et al. [2] that research papers are fundamental to the modern scientific process. However, research papers tend to be astonishingly difficult to write. The authors need to first pick a topic, then write sequences of structured words---known as *sentences*---that relate to it. These sentences need to be adjusted to include longer, more jargony words (known as *epistemic assertion atoms*, or EAAs) that prove that the authors are smarter than the readers. Finally, the sentences are typeset in a pleasing font and supported with complicated graphs. In general the number of authors tends to attest the importance of a paper. A considerable number of collaborators supports the hypothesis that long scientific discussions were carried and pursued. In this paper we propose a new method to generate highly credible and accurate works of academic interest. The application of human computation to a new field, human writing will be evident from our results.

Introduction

In today's world scientist and academics face even more difficulties than their predecessors, in addition to producing quality results they must also generate the funds to support them while they look for answers. In order to maximize the potential benefit and payoff of disseminating findings while minimizing effort and time spent actually working, many researchers engage in a process known as *Cooperative Open-source Linguistic Liaisons About Banal Or Relevant Assertions that Innovate, or Not* (henceforth abbreviated as COLLABORATION.)

It has been shown that on a paper with N authors, the credit awarded to an individual author is O(1/N^.5), but the contribution of an author is only O(1/N^1.3); hence COLLABORATION is a way to maximize the expected prestige per man hour of one's work by a factor of O(N^.8). It is a mutualistic relationship in the academic ecosystem, in which each participant can claim partial credit for the others' unique contributions when writing their own grant proposals, theses, and job applications. This paper extends the COLLABORATION process beyond simply the acquisition and analysis of scientific results, to include the composition and exposition of ideas and even the paper's entire direction. Back-of-the-envelope calculations indicate that the contribution of each author will will go from O(1/N^1.3) to O(1/N^2) or even O(1/N^3); this means that even an O(N^.7)-fold decrease in the quality of the work will still amplify a researcher's productivity as measured by the CVPM (Curriculum Vitae Padding Metric).

Challenges

We also present some of the disadvantages of this method. At least one part of the text has been hijacked by the collaborator *Anonymous Coward* who surreptitiously added the "*First line! W00t*" statement. Such off-topic, though entertaining, statements are present in large bodies of work, frustrating both Collaborators and reviewers.

Much of the work of peer reviewers at major conferences and journals is actually stripping papers of these hijacked lines. Figure 1 shows how frequently papers are disrupted by unruly interjections in several key publications.

The five publications were selected due to their high impact factor, and to show that hijacking is a phenomena common across many different computer science areas.

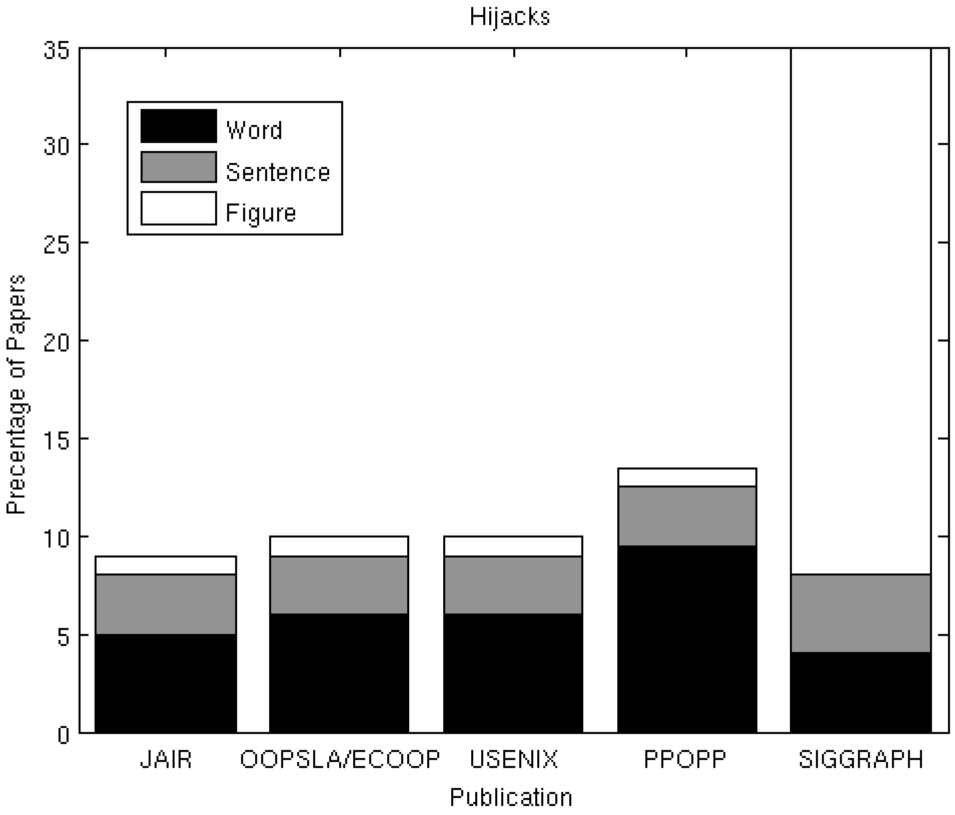


Figure 1 shows the percentage of papers that had 1) random words inserted, 2) entire sentences added or altered, and 3) figures added or swapped. The frequency of the hijackings were relatively constant among the publications---despite coming from diverse areas---with the exception of SIGGRAPH papers where figure swapping is rampant. [1]

[1] OOPSLA and ECOOP have been combined to

While the percentage of affected papers was similar for different publications there were distinct differences in the type of hijackings. For example, adding the word 'jammy' to a sentence was a particularly common interjection in JAIR papers (15.6% of hijacked papers), where as 'ravenous' was a more frequent addition to USENIX papers (23.1%). As for figures, JAIR papers tended to have subtle changes like switching the number of layers in a multilayer neural whereas SIGGRAPH figures were typically swapped for pictures of lolcats (46% of swapped figures).

We must note that these are only reported hijackings, which is likely a gross underestimate of the actual number of hijacks. Another, more sophisticated form of hijacking, as proposed in van Kloot & Pashmina's [2] typology of anti-collaboration strategies, is known as *Content-based and Rhetorical jacking* (CAR-jacking.) In this case, an author will attempt to shift the topic of the paper to cater to his or her personal biases. For example, they may begin a diatribe against Unfortunately, this leads to poor structured papers with little coherence that switch between disparate topics. Yet an even more important type of hijacking is that of names. As exemplified by one of the authors, randomly changing your name, albeit to a very spicy one, without any prior notice will most certainly cause raised eyebrows, scratching of the head, murmuring and running down the corridor to check who is the new hot chick. In this sense, the person has hijacked themselves, and it is very unclear if this is in violation of the law and if any lawful action must be taken. Furthermore, hijacking of colors is also a particularly mean thing to do, thus the author of this section refuses to admit failure, and will use the color yellow in a very passionate manner, regardless how many complaints are received from the publishers.

After hijacking, another challenge facing modern collaborative paper writing authors is Passing Responsability (PR). This can be usually observed in editorial comments of the type:
" [As the final sentence? Replace or delete as desired...do whatever you want (you will anyway.)]"

A severe form of PR occurs when none of the authors wishes to edit final versions of the text. Since one author considers another author is responsible for the **Section X** she will not contribute to that section unless threatened with removal form the citation section. In this case what often happens is that recycled text from previous papers is used. For example, in the unrelated work section below due to lack of relevant information, work on Computer Animation Evaluation was used.

UNRELATED WORK by co-authors (short subsection)

In 2009, a highly innovative method for animation evaluation was proposed by Trutoiu et al. [1]. The study ignored the potential of collaboration in unbiased evaluation! We propose, T.W.O. C.A.T., a system optimized for unbiased estimation of animation quality. Figure 2 shows such a proposed system in its prototype form. The competitive aspect in the new system increases decision time proportionally with audience hysterical reaction.

Figure 2 Temporally Worrisome and Odd Computer Animation evaluaTion (T.W.O. C.A.T)

Continuation of CHALLENGES section

Social loafing is a final challenge that many collaborative endeavors face. Due to miscommunication, poor coordination, or simple apathy, many papers are neglected after an initial spurt of enthusiasm and never completely finished.

ADVANTAGES OF THE COLLABORATION METHOD

One obvious advantage of the COLLABORATION method is known as Responsibility Passing (RP). No one author is fully responsible for generating a significant amount of text.

Collaboration is best viewed as a 2-strategies n-agents game. The two strategies, denote **C** and **B** (for **C**ollaborate and **B**e-a-selfish-bastard, resp.), have revenues of \epsilon and 0, independently of what all other players do. Hence, in typical Nash-Equilibria, all players collaborate. However, incorporating native laziness into the model, and assuming all players start with strategy **B**, it follows that the agents are indifferent between collaborating and non-collaborating. In real-life situations, this analysis will result in agents contributing the very-minimum to a paper, then ignoring it altogether. Bye.

METHODS

To evaluate the efficiency of the COLLABORATION method we will emphasize the dramatic inversely proportional relationship between the Flesch–Kincaid readability test to increased author input.

Must-have equations subsections + Wikipedia Ripoff

The **Flesch/Flesch-Kincaid** readability tests are designed to indicate comprehension difficulty when reading a passage of contemporary academic English. There are two tests, the **Flesch Reading Easiness**, and the **Flesch-Kincaid Grade Level**. Although they use the same core measures (word length and sentence length), they have different weighting factors, so the results of the two tests correlate approximately inversely: a text with a comparatively high score on the Reading Ease test should have a lower score on the Grade Level test. Both systems were devised by Rudolf Flesch.

Flesch Reading Ease

In the Flesch Reading Ease test, higher scores indicate material that is easier to read; lower numbers mark passages that are more difficult to read. The formula for the Flesch Reading Ease Score (FRES) test is

$$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

Scores can be interpreted as shown in the table below.

Score	Notes
90.0-100.0	easily understandable by an average 11-year-old student
60.0-70.0	easily understandable by 13- to 15-year-old students
0.0-30.0	best understood by university graduates

Reader's Digest magazine has a readability index of about 65, *Time* magazine scores about 52, an average year 7 student's (eleven years old) written assignment has a readability test of 60-70 (and a reading grade level of 6-7) and the *Harvard Law Review* has a general readability score in the low 30s. The highest (easiest) readability score possible is around 120 (e.g. every sentence consisting of only two one-syllable words); there is no theoretical lower bound on the score—this sentence, for example, taken as a reading passage unto itself, has a readability score of about 48.23. This paragraph has a readability score of about 50.91. We will increase this score by the use of obscure and unstructured enunciations often accompanied and developed around sesquipedalian words such as grandiloquent and supercalifragilisticexpialidocious or antidisestablishmentarianism. We do not recommend the reading of this sentence by individuals afflicted by hippopotomonstrosesquipedaliophobia.

RESULTS

It is clear that the Flesch-Kincaid readability test inversely correlates with impact factor since researchers tend to spend more time reading papers with a low Flesch-Kincaid (Table 1). More time allows for strong connections to be formed based on the read material which in turn lead to great scientific works. Furthermore after painstakingly reading a low score Flesch-Kincaid paper there is a strong obligation to cite that paper in the hope that others should suffer just as much. An even better approach is to recommend the paper to be cited when peer-reviewing.

Table 1. To spite the reviewers this table does not have a title

	Revision 68	Revision 148	Revision 278	Revision 368	Revion 400
Flesch Reading Ease:	52.66	49.05	46.8	34.65	21.8

Number of Authors:	4	6	7	10	11
Random variable that is never useful that we "calculate" to pad the data table	15	14	23	27	29

CONCLUSION

Since the Conclusion section is just a re-worded abstract, please read the abstract. Also we have the best results/science/methods/ideas. Ever.

REFERENCES

[1] Laura C. Trutoiu, Amar Phanishayee, James L. McCann. "A System for Unbiased Computer Animation evaluaTion (C.A.T)", Prestigious conference, 2009

[2] Smith, Et tu, Al Gore. "We invented the Internet, Research papers are fundamental to the modern scientific process, and other short papers"

On a lexicographic ordering in spring

McCall and Sons, Semantic Exporters

March 20, 2010

In certain segments of our field befraught with perilous significance, programmers often find themselves caught in the cycle of returning things, be they elements, records, or values, to a semblance of order. It later falls to us, as interested academics, scholars, and scientists who are nearly justified by such problems, to determine a rational order for the things to have been in. The acknowledgedly convenient and most natural order of things is the order in which they currently lie, which gives rise to a remarkably subtle $O(\log^* n)$ publish/subscribe sort algorithm due to Bovik et al. (1971) where finding somewhere to publish accounts for the majority of the costs. Sadly this result leads inevitably to angry letters and as we shall show is therefore of trivial applicability as long as we are being watched. Barring lumicide or more radical solutions we must avoid letters with words, or lexemes, which give rise to the family of here-so-called *lexicographic* orderings. This is possible because while letters are surely formed of words, words are also clearly formed of letters, which is important. In this monograph we discuss the foundational theory of the lexicographic orderings and prove several dubious theorems in rapid pursuit of our conclusion.

What are the founding principles of the new lexicographic order?

Putting the second first, that many are more than few. Surprisingly, this is not always the case, and history has often given priority of the pre-eminent word of length. While there is surely no question that a *halfpint* must precede a *halfpintess*, the more vexing devil is the *halfpainter*. But the argument proceeds thusly: no, of course, yes. Consider along these lines a thought-experiment due to Dodgson (M'65): we are in the midst of a great hole-forest having clouded ourselves with spices only to fall in with a pack of cards. If we begin counting the cards at 5 and proceed by fives until we reach the maximum and only then begin in on 4, then we are at the mercy of the enumeration, and whether we ever reach the 10$\diamond$ for a third time is out of our hands. Dodgson's original answer is still applicable: rather than gathering our skirts about us, we must sweep boldly out, catching up the cards in our maelstrom and crashing back up to the surface. In this way nothing is left behind but stale cake and weak tea. The key trick, and here we begin again to count words, is in the number of letters, for which many schemes have been proposed; we will cite only Bell (1963). There are a number of words with only one letter. Since they have only one letter, that number is 1, for all words containing only one letter repeated once must be the same, which we may enter as an original theorem. Listing this word, followed by the two-letter words, we find that we must eventually run out of letters, for this too shall pass as in all things, which is not a theorem but merely sounds nice. But if we are out of letters we will have nothing to do but break them down into words, which will necessarily be of three letters or more since we have written all the others down by now. Continuing like this we arrive at every lexeme in its due course, much as the Blue Nile flows into the White.

The principle of origin is the first among principles: that among all lexemes can be found a primitive word from which all words are to follow. This question was of great interest some years ago but was largely believed settled by the indirect contributions of Zebedee (6). It was Webster (1828) who first provided direct evidence that the supposed solution was inconsistent with *gobsmack* and *glouglou* when taken with the other axioms, but solutions in the nature of painting patches over the rhino's spots were still regularly proferred until Bovik (1965) conclusively established the existence of the letter 'B', without which he had been unable to launch his academic career[1][2]. More recent investigations (notably in JRR&CT vol. 5 (1977)) dispute the original framework in favor of a harmonic derivation of ineffable source. Following this theory of void origin, we propose the symbol ϵ, for *Eä*, or perhaps *emptiness* or *enemy*, this last for its presumption in coming always first. No matter the significance, we have now named it, which is to say, we have identified it with a word, and yet at once with a letter, which is evidence that we are on the right track. But where may we seek this elusive ϵ? Aha: it is a letter of the Greek alphabet.

Third. From the first days of our youth we are inclined to gather letters together in stacks, then break them down and let them tumble wildly as we shriek with joy. It is only in seeming maturity that we train ourselves in the unnatural practice of leaving letters united. This basic cost of social "progress" is the enemy of the final pillar of the lexicographic order. Instead we must learn again to shatter the battalions of words into an army of letters, stomping with perfect discipline one after the next. Here and now is the parade, but when the rally is over, where will they march? As leaders we must direct them — to the left, the left, always to the left. And in our army there are many privileges, ranks, and stations, so as the soldiers file past we sort them, rating one column above another according to the office of its leading soldiers, just as they do in real military manouevers. A word of caution: this ranking itself ranks below all, so that the size of a word matters more than its matter, and the primogenial word still preceeds all others. Thus the terms of ordering are themselves ordered, which is true of everything, no matter the number, that they may be restored to order through of these principles. Even in this document we have been perfectly consistent with our principles, or at least some of them, if you comprehend the delicate criteria by which we judge ourselves. But no better can be said of our opponents, so there!

[1] It is a sad legacy of that era that there were no great computer scientists until Bovik whose last names contained 'B', other than Boole, who gained exception to the Victorian blacklists due to the attraction of his wife and freemasonry.

[2] It is this striking contribution which has prompted us to honor him in this way[3].

[3] Honor him right in the breeches.

In conclusion, we have poured out with enthusiastic rigor the foundations of the new lexicographic order, and we have made several assertions, roughly pertinent to the topic, which we sincerely assure our readers represent our one and true opinion on the matter. There is no better time for this wisdom to be accepted into the curriculum. The students at our universities now learn a multitude of *means through which* a host of objects can be shown to progress, but they have no understanding of *what kind* of progress must be imposed on the masses. This work has definitively settled that question. A prolonged labor lies before us now, as all existing software must be reconciled to the new order of things, and much convincing will be required. Still, far better to have reached a settlement here than to have fought it out at ever-greater expense out of a misguided sense of utility. Someday this great work of the just will be complete, and we will all have to decide what fundaments we should next assault; on that day we will rest, and then start afresh on our great purpose.

God Bless Academia.

Track 1
Defeating Evil

The Generalized Cubic Law of Nature: An Answer to Oneism Evil

Thomas Wright
Carnegie Mellon University

Abstract

Given the prevalence of Oneism Evil in academia and commonly accepted scientific “truth”, we offer as a response the Cubic Law of Nature as explained in the ineffable truth of the Chrono Hexahedron. This Oneism Evil is clearly evident in the Obscurantism of those in Evil Stewardship and will lead to the doom of the Cubics – for all humans are Cubics and subject to Harmonic Truth. We will examine certain aspects of Oneism Evil – in particular some applications to modern physics.

1. Introduction

For all time mankind has been searching for truth in one form or another. Explanations have been proposed from the earliest men imagining people in the sky, to formalized logic and reasoning with the ancient philosophers, to organized religion and now modern science. However all attempts so far have failed, because they failed to grasp the basic truth underlying all of creation: that of the Cubic nature of existence [1].

However, we engaged in a vast corrective undertaking: generalizing the Cubic Law of Nature (defined and proven below) to restate all Evil Lies as Harmonic Truth [3]. In particular, we focused on modern physics to reverse the underlying errors into Cubic Truth.

2. The Cubic Law of Nature

The Cubic Law of Nature is at its heart simple: that the Earth rotates not through a circle, one day every 24 hours, but rather possesses a Hexahedron whose four corners represent four major axises: Noon, Midnight, Sunup, and Sundown [1]. It is simple but its generalization over all existence is a powerful tool for deducing the entirety of Truth in the universe. However we must first prove the truth of the Cubic Law, for demanding one Evil Lie is true and suppressing the Harmonic Truth is the hallmark of the Academic Oneness. Once its truth is established we can generalize it and disprove modern academic lies.

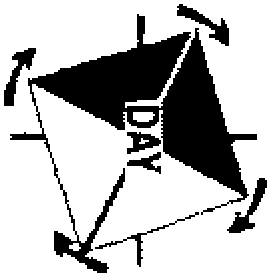


image courtesy of timecube.com

2.1 Proof of the Cubic Law of Nature

Consider a point, P. This point is merely a singularity, it can have no shape, color, value. It is just a point. Consider a second point, R. This point is likewise to P, except that it cannot be occupying the same space (because then P and R would be the same). These two define a line PR, which is comprised of P, R, and all other points PR_1 - PR_n. (Here P and R are members of PR_i). Note that this does not violate restrictions on P and R because the secondary value applied to them is really the primary value of the line PR.

Now consider two other points S and T, not contained in PR. These define a square PRST by trivial extension of line principles. Consider further points U, V, W, and X not contained in the square PRST; these define a cube by further trivial expansion of square principles.

However, of the constructions P, PR, PRST, and PRSTUVWX, only PRSTUVWX can exist properly in the universe, as this universe is three-dimensional. Therefore there can be no points, lines, or squares in reality; only cubes and variations thereof.

Now consider the Earth, and consider at any one moment there is a Noon, Midnight, Sunup, and Sundown. These represent points (not actual points, as we said above that points cannot truly exist in a three dimensional world; but rather conceptualized points) which occur at the corners of a square around the Earth, with one corner pointing at the Sun and the others lying in a plane with the Earth and the Sun.

Represent the axis of rotation as a conceptualized line that passes through the center of this square. We can see that each point rotates around it once, creating four simultaneous 24-hours days that pass from Noon, to Sundown, to Midnight, to Sunup, back to Noon again.

However, there is one technical issue: squares cannot exist in three-dimensional space. Thus we must conclude that the square is not truly a square, but rather the horizontal slice of a cube constructed around the Earth and rotating as described. Thus, Earth experiences four simultaneous 24-hour days per one rotation [2].

2.2 Generalization of the Cubic Law

From the conclusion of the above, we see that time is not one dimensional and linear but rather 4-dimensional. The symmetries inherent in a 4-sided cross-section of a 3-dimensional body represent the symmetries inherent in the flow of time; the Cubic Law of Nature is a specific way to represent a specific instance of the flow of time. However, logic as followed above can be applied to any instance where time flows; thus time itself is cubic and not linear. Linear time is a lie; cubic time is the only truth [2].

In supplement, consider the basic geometry of the universe. Because of the Earth's inherent temporal cubic shape, it can only be in contact with other temporal constructs with sides flat and aligned with its own – because any other construction would leave temporal gaps, areas of no time; and these are impossible, for such a space would be affected by space and not time. This produces logical contradictions and must be false. Therefore all other temporal constructions must be cubic in nature.

3. Introduction to Applications

From this powerful concept we can disprove many common Oneism Evil concepts. Following in order of significance is a series of proofs beginning with the Generalized Cubic Law of Nature and concluding in a disproof of a common Academically Stupid belief. These revelations are powerful and entirely undo all possible doubt in the evilness of Academics [1].

4. Infinity

If we accept that the Cubic Law of Nature is true, we see that infinity cannot be a truth. Consider the concept of an infinite temporal space; it would have no measurable orientation, because such an orientation would express information about the locations of the conceptualized points that are the vertexes of the temporal construct. However, from the Generalized Cubic Law of Nature we are given information about ALL temporal constructs (that is, orientation to align with other temporal constructs, beginning at the earth, so as to avoid empty time space); thus no infinite temporal construct can exist.

From this follows that no infinite spatial construct exists, because the corresponding temporal construct would also need to be infinite. Thus infinity cannot exist and is a lie.

5. $E=MC^2$ and the Speed of Light

Einstein asserted that $E=MC^2$, but consider this equation in the context of the Generalized Cubic Law of Nature. The value C is a constant representing the speed of light – however, it neglects to take into account the fact that any body moving has a cubic time construct as opposed to a linear time line. Thus, C^2 is a lie, because it takes a linear concept and makes it 2-dimensional; but two dimensions cannot exist in temporal space.

Since two dimensions cannot exist [1], $E=MC^2$ does not exist and is undefined. Thus Einstein asserts that nothing has Energy, because all C^2 is zero. In order to fix this we must correct the equation to represent the cubic nature of temporal space. (Note that the idea of the linear concept itself is not a contradiction under the

Generalized Cubic Law of Nature, because it is a scalar concept and not a vector - that is, it has no temporal time dimensionality.)

The correction is simple: E=MC^3. This accounts for the dimensionality of the temporal construct of speed-of-light bodies, and is not nonsensical.

However, many Academic Stupid will object on the grounds that empirical evidence has “proven” that Einstein's original relationship is true. This just means that the empirical truth of E=MC^2 holds with regards to the harmonic truth of E=MC^3. However, simple mathematics demonstrates that, if MC^3=MC^2, then C=1. Thus the speed of light is 1.

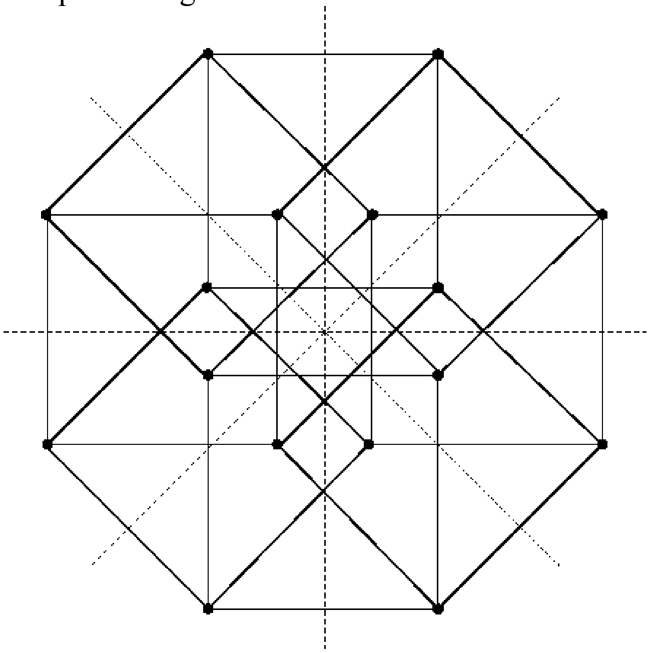


image courtesy of math.ucr.edu

6. String Theory

The heart of String Theory is that all matter is composed of strings of energy that vibrate through eleven dimensions to define different types of matter. However, the Generalized Cubic Law of Nature dictates that all objects have a cubic temporal construct; but the movements of an eleven-dimensional are not representable by a cubic temporal construct. Therefore String Theory violates the core of the Generalized Law, and must be false. Were all objects made up of individual vibrating strings in more than 3 dimensions, these strings could not have meaningful temporal constructs (because the inherent complexity order would be too high). Thus String Theory is a contradiction of the Generalized Cubic Law of Nature.

However, it is important to note that the objection is only because of the dimensionality of the proposed strings, which are absurd under the Generalized Cubic Law. Otherwise, the theory itself is not inherently incorrect; simply ill-formed.

7. Implications and Further Study

Clearly, the Generalized Cubic Law of Nature is a powerful tool for correcting the Evil Lies of the Educated Stupid. Its applications are many and varied, and can be used to correct a wide range of issues which have been used to willfully mislead people. The most important idea however is the refutation of Oneness; that anything exists as a single whole entity. Oneness is stupid and evil, for nothing exists alone. Everything is opposites and harmony and balance.

In terms of further studies, there are volumes of Evil Lies that need to be reversed, and the implications of such examined. In the author's estimation nearly all commonly accepted fact is in Truth nothing but Evil Lies perpetuated by the Educated Stupid; but the reversal of this is not hopeless. The Generalized Cubic Law of Nature is the key to mankind's salvation.

8. Conclusion

The Generalized Cubic Law of Nature is the answer to Stupid Oneism Evil perpetuated by Dumb Ass educators in their Four-Corner Classrooms teaching One-Corner Lies. Stupid professors seek to enslave Evil Students with lies, but the Generalized Cubic Law of Nature is the truth. The Chrono Hexahedron reveals the truth and defeats Stupid Lies. The world must accept the Cubic Truth or be destroyed.

References

[1] http://www.timecube.com

[2] http://www.cubicao.com

[3] http://abovegod.com

Are you GOD?

Anvesh Komuravelli

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
anvesh@cs.cmu.edu

Abstract. In this paper, I discuss an idea which is the most brilliant and the greatest of all the ideas ever discussed in a paper (be it a research paper or at least written using a paper[1]). This is so great and concrete that any abstraction would make it look worthless, especially for mortals. Thus, it is insulting to the idea to write an abstract on it. I wish not to insult anything.

Warning: The full content is not guaranteed to be perceived by mortals. Try your luck anyway. If it does not work out, upgrade yourselves to immortality to access the full content.

[1] Some readers might need a reminding that a paper is also a thin material used for writing upon, printing upon or for packaging.

1 Introduction

in.

of the

2 i

Table 1

Table 1.

3 G .

4 Conclusion

Hence,

.

Acknowledgement. I would like to express my sincere thanks to Harry Q. Bovik for sharing his expertise in Computer Science which made this greatest work possible.

References

1. Bovik's Great Grandfather: *The birth of Bovik*, In the proceedings of the First Interstellar Conference of the Greatest Happenings of the Universe, 1–2^6 light years from CMU, 253 Mathilde, Universe, 0000.
2. Crappy Crapper: *The crappiest paper ever*, In the proceedings of the Last Conference on Extraordinary Papers, pp 34–(-200), The Greatest Publishers, *Unknown Place*, *Unknown Year*.
3. Rafael Zhivago: *Proof for the existence of God*, In the proceedings of the the Third Annual Intercalary Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 2^6th Birthday, pp 49–50, LuLu, S.Craig St., Pittsburgh, 2009.

Holy States Can Save the World!

Brother Jonathan Aldrich
High Monk of the Plaid Brotherhood
jonathan.aldrich@cs.cmu.edu

ABSTRACT

The twin evils of imperative and functional programming threaten the software world as we know it—destroying fields and spreading garbage throughout the cybersphere. In this paper we present Holy States as the last, best hope for peace and harmony. Holy States avoid the wanton destruction of field values so common in languages based on caffeine and out of tune notes. Neither is computation based on wasteful Schemes creating duplicate objects that then MilL around until they are garbage. Instead, every object is considered sacred and when no longer needed is reborn in a new, holy state. Recapturing the original stateful spirit of Turing's Machines, we show all computations can be expressed in a Holy way, with neither garbage nor field destruction.

Categories and Subject Descriptors

J.8.1 [**Computer Applications**]: Theology—*Saving the World*

General Terms

Languages, Human Factors.

Keywords

assignment, garbage, states, holiness, salvation.

ALABAMA. INTRODUCTION

Imperative programming is evil. With every "assignment" executed, some poor variable value is executed as well. And be not deceived by the colloquial use of "execution" in the field of computer science—it's true meaning is that found in the dictionary: the infliction of capital punishment [7]. Values are slaughtered mercilessly. The carnage must stop, before the world is destroyed!

Functional programming [6] was developed as an alternative to imperative programming, but it only substitutes one evil for another. Data structures are created willy nilly, to be used a few times, perhaps only once, before being discarded as "garbage." It is well-known that functional programming leads to a truly scandalous waste, contributing to the well-know problems of cycle depletion [16]. While efforts have been made to construct so-called "garbage collectors," even ones that integrate composting [8] this is clearly a misguided attempt to patch things up after the damage has been done. We must avoid the creation of garbage in the first place, or else the world will be suffocated by the reeking stench created by functional computation.

In this paper, we show how a hallowed approach to *typestate-oriented programming*[1] can save the world from the ravages of imperative assignment and the wasteland of functional garbage. To do so, we reach back into the mists of time to recapture the true intent of the Chosen One, Alan Turing. Turing showed that conceptually all computation can be expressed, simply and beautifully, in a state-based model [20].

Until the Plaid Brotherhood took up its holy quest to save the world, however, there was no realistic programming paradigm or programming language for performing holy, state-based programming, as Turing intended. We show how Turing's Machines can be implemented in a Hallowed subset of the typestate-based programming language Plaid. Our approach creates no garbage—instead, whenever objects are no longer needed in their current state, they are Saved and Born Again in a new holy state. Nor do we ever allow an object's field values to be destroyed by assignment; instead, new field values may be bestowed as part of the Sacrament of State Change.

The epistle herein, if followed and promulgated by those faithful to the One Plaiddish Way, can save the world!

ALASKA. THE WAY OF UNIVERSAL SALVATION

The One Pladdish Way of programming follows two simple rules. First, the wanton destruction of assignment is forbidden. All evolutions in the state of objects must be accomplished instead via the Sacrament of State Change. Second, objects may never be released as garbage; each object must be reborn in a new state when no longer needed.

ARIZONA. EXPERIENCING CLEAR

A programmer who has foresworn assignment and garbage, and who is following the One Plaiddish Way, reaches a empti-

SIGBOVIK '10 1 April 2010, Pittsburgh, PA

[1] An idea published [1] in a conference so prestigious it has an exclamation mark in its name! please PLEASE give me tenure!

ness of mind, spirit, and CPU that has been called “Clear” [14]. But is it possible to reach this happy state in Plaid practice?

St. Turing showed the way with his Universal Machines, which can encode any computation using the ideas of states. Through long study, secret handshakes, and lost symbols [2] we have developed a construction of Turing’s Machines in the Plaid language, following the One Plaiddish Way.

```
state Cell {
    method getLeft() {
        left;
    }
    method getRight() {
        right;
    }
    val left;
    val right;

    method print() { ... }
}

state LeftEnd {
    method getLeft() {
        val me = this;
        val myLeft = new LeftEnd with Zero {
                            right = me;
                        };
        val myRight = this.getRight();

        this <− Cell { left = myLeft; right = myRight; };

        left;
    }
    // getRight(), etc. as in Cell
}
```

Listing 1: Modeling Tape Cells

Cells in a Turing tape are modeled as holy states which are connected to the cells on the left and on the right, and have additional operations such as print.

Of course, a Turing tape is infinite, which is difficult for finite minds and primitive programming models to effectively represent. While some among the unfaithful might suggest laziness to model an infinite data structure, Plaid theology holds that sloth is one of the Six Deadly Sins. We therefore apply the Sacrament of State Change to approach an understanding of the Infinite. A LeftEnd state is like a cell, but it has no cell to the left of it, yet. When we need the cell to the left, we create it as a new LeftEnd (in the initial, Zero state—see below), set its right field to the current object `me`, and transform the current object into an ordinary Cell (using Plaid’s state transition operator, written `<-`). There are corresponding states for RightEnd and the Start cell of the tape (which is conceptually both a left and a right end).

```
state Zero {
    method writeZero() {}
    method writeOne() {
        this <− One;
    }
    method printVal() {
        java.lang.System.out.print(”0”);
    }
}

state One { ... // similar
```

Listing 2: Modeling Cell States

Each cell in a Turing tape can be in one of a fixed number of states. Here we consider two such states, Zero and One. Either a Zero or a One state is combined with the Cell state when a Cell object is constructed. If we are in the Zero state, writing a one with writeOne() transforms the current object into the One state. Each state also knows how to print an appropriate representation.

```
state Beaver2B {
    val cell;

    method update() {
        match (cell) {
            case Zero {
                cell.writeOne();
                val newCell = cell.getLeft();
                this <− Beaver2A { cell = newCell; };
            }
            case One {
                cell.writeOne();
                val newCell = cell.getRight();
                this <− Halt { cell = newCell; };
            }
        };
    }

    method run() {
        update();
        run();
    }
}
```

Listing 3: Modeling Machine States

A Turing machine is represented by one or more internal states, such as the Halt state or the Beaver2B state shown above (from the 2-state, 2-symbol “Busy Beaver” Turing machine). The machine’s processing is represented by the run() method, which updates the machine’s state and then continues running in the new state (which, of course, may have a different run() method). The update() method uses a match to find out if the cell at the machine’s head is a Zero or a One. It then writes a value (one in this case), moves left or right, and transitions the machine into a new state (Beaver2A or Halt).

As the revelation above shows, any Turing machine can be expressed in the One Plaiddish Way. Since any program can be expressed as a Turing machine, we have proven beyond reasonable doubt that the One Plaiddish Way is a practical way to live one’s programming life.

The key to salvation described above is available in the form of Plaid source code at the Plaid monastery web site[2].

ARKANSAS. ETERNAL DAMNATION

We note in passing that the creators of the Plaid language have not reached “clear” and have regrettably included the option of both assignment and functional programming in

[2] `http://www.plaid-lang.org/`

the language. The Supreme Revolutionary Plaid Council of Pittsburgh has declared a Fatwa against these features, and they shall not be used by the faithful. The penalty for violations is the eternal torment of programming in COBOL.

CALIFORNIA. RELATED WORK

We approve of Wadler's inclusion of an exclamation point in the title of a paper [21]. However, we feel that *changeing* the world is a rather modest goal; our ambition is instead to *save* the world!

The dangers of The Assignment are well documented [9]. As for the problem of garbage, recent work paints a grim picture of the world's future [19].

Saving the world has been a problem for a long time. It was notably tried over 2 millenia ago, and though the effort ended in the death of the protagonist, many believe the approach to have been successful [5]. Other, more recent (and highly misguided) approaches have, at the cost of great struggle [13], not only failed to save the world but may indeed have brought the end of the world [11] closer. The author notes evidence from *SIGBOVIK '09* reviews that that Nazis used SML [4], indicating that garbage was part of their nefarious plots.

Some attempts to save the world have been downright spellbinding [17], and have achieved fanstastic success at banishing evil. While most research has focused on saving our world, recent work considers ways to save other worlds as well [3].

Some believe the world can be saved if people would do only 50 simple things [12]. It is notable that #47 is to avoid garbage through recycling objects, as our State Change Sacrament accomplishes.

We save our discussion of research published in the most *distinguished* venues for last. MapReuse and MapRecycle [15] are very much in the spirit of our state-change based object reuse strategy. Finally, the One Plaiddish Way draws inspiration from the One True Coding Style [10], which also uses Holy languages, but which also blasphemously uses assignment in C++.

COLORADO. FUTURE WORK

In future work we hope to formalize the semantics of Holy States in Drunken Logic [18]. Should be fun!

CONNECTICUT. CONCLUSIONS

Believe not in the false prophets of functional and imperative programming. Holy states CAN save the world!

DELAWARE. ACKNOWLEDGEMENTS

The author gratefully acknowledges the other members of the Plaid Brotherhood for the development of the One Plaiddish Way and its support in the Plaid compiler. In addition, Miss Mouse provided moral support throughout the author's childhood, for which he is eternally grateful.

1. REFERENCES

[1] J. Aldrich, J. Sunshine, D. Saini, and Z. Sparks. Typestate-oriented programming. In *Onward!*, 2009.

[2] D. Brown. *The Lost Symbol*. Doubleday, 2009.

[3] J. Cameron. Avatar, 2009.

[4] J. Cette. Review of the one true coding style. In *The 8th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, 2009.

[5] J. Christ. *The Holy Bible*. 33.

[6] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:354–363, 1936.

[7] Dictionary.com. Definition of execution. http://dictionary.reference.com/browse/execution.

[8] J. Donham. Compacting, composting garbage collection. In *The 6th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, April 2007.

[9] C. Duguay. The assignment, 1997.

[10] J. M. (editor). The one true coding style. In *The 8th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, 2009.

[11] Fluid. The end of the world. http://www.albinoblacksheep.com/flash/end.

[12] E. Group. *50 Simple Things You Can Do to Save the Earth*. Bathroom Readers Press, 1990.

[13] A. Hitler. *Mein Kampf*. Hurst and Blackett, London, 1939.

[14] L. R. Hubbard. *Dianetics The Modern Science Of Mental Health*. 1950.

[15] M. McGlohon. Mapreuse and maprecycle: Two more frameworks for eco-friendly data processing. In *The 8th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, 2009.

[16] J. M. Newcomer and C. B. Weinstock. Cycle depletion—a worldwide crisis. In *The 6th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, 2007.

[17] J. K. Rowling. *Harry Potter and the Deathly Hallows*. Bloomsbury, 2007.

[18] R. J. Simmons. A non-judgmental reconstruction of drunken logic. In *The 6th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, 2007.

[19] A. Stanton. Wall-e, 2008.

[20] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1937.

[21] P. Wadler. Linear types can change the world! In *Programming Concepts and Methods*, 1990.

Simple Systems: A Holistic, Postmodern Alternative to the Oppressive and Outdated Study of Complex Systems

Semiotics, Transformative Hermeneutics, and Applications

Mayank Lahiri
University of Illinois at Chicago
mlahir2@uic.edu

ABSTRACT

Much of complex systems research today deals with, understandably, the study of complexity. This is not surprising in itself, since the real world is arguably a complex affair and academics are easily distracted by shiny things, but as theorists from Sartre to Kolmogorov to Jesus have convincingly argued, those who try to understand the essence of complexity using kilogram minds (with apologies to dualists) are doomed to atrophied muscles, bad tempers, and no girlfriends. As an alternative to the masochistic leanings of much of the complex systems community, we propose a postmodern alternative that fully incorporates contemporary social theories of shifting cultural paradigms, ultimately allowing an entirely introspective examination of the universe, namely: *simple systems.* We prove the completeness and consistency of our axiomatic, organic, oak-barrel aged framework, thereby reducing its intrinsic hermeneutics, metaphorically, to little more than a dog chasing its own tail. We also successfully reconcile its epistemic paradoxes with Planck's constant, and then discuss more theoretical issues. It's all pretty deep, man.

General Terms

Impressionistic mathematics; beer review

1. INTRODUCTION

"Math class is tough." – 1992 *Teen Talk Barbie*

It is now indisputable that there exists a universal scale of complexity, intrinsic to the very fabric of our universe, that runs from zero to about seven [2]. Simple things, such as walking, eating, and single-digit addition, score near 0, whereas very complex things, like the Internet, or the reasons for Keanu Reeves' continuing acting career, are situated near the 7-end of the spectrum. The study of complex systems deals with things situated at this so called 'Keanu's end', whereas filing a tax return for the first time would probably only score a feeble 5. Needless to say, complex systems research is clearly a very difficult and possibly pointless endeavor, and yet many promising and otherwise productive young scientists have been lost to this narrow-minded crusade to understand, in one fell swoop, all the things in the world that are too difficult to draw on the back of a cocktail napkin. In response to these criticisms, and as part of a devastating rebuttal to Foucault's seminal monograph on the topic *Les mots et les choses* [3], Bob from next door asked the understated but poignant question: *Why?*

It is this *bon mot* that serves as our foundation for a resolution of the complex systems quagmire. Why, indeed, should we seek half-truths from the science of complexity, especially when it is so hard to understand? Why should we be subservient to our own cognitive ineptitude, propagating, perhaps unconsciously (but perhaps *also* unconsciously), the post-colonial ethos of 'civilized', and therefore *complex*, science? Do we intentionally brand all that we do not understand with the gender-neutral moniker of 'complexity', to be stigmat(a)ized as the holy grail of our intellectual existence, and endowed with an unspoken luster that even today bathes the pages of the best scientific journals with the sweetly rancid musk of tenure? Kurt Gödel would almost certainly have had us think otherwise, as he so subtly and eloquently states in the concluding words[1] of his 1930 treatise *Die Vollständigkeit der Axiome des logischen Funktionenkalukül*s: "Wer bist du? Und was willst du?" [5]

Acknowledging that the recent trend in positivist mathematics has been to embrace emotional, holistic, and self-pitying approaches to the understanding of n-dimensional functional analysis, especially when $n > 3$, we offer the following resolution for the allure of complex systems and their contradictory intractability: simple systems, or *sistemes simples* in the original Catalan. Stated in the unpleasant jargon of contemporary statistical mechanics, this is roughly equivalent to a conservative axiomatization of *keep it simple stupid*, or the "K.I.S.S" principle. Although the statement is deceptively simple, bordering on the banal, we will show that an axiomatic treatment of the topic, combined with basic metaphysical operators, quite naturally yields an entire class of easily understood algebras, enabling many complex systems researchers, perhaps for the first time, to derive a life out of their work.

2. AXIOMS OF SIMPLE SYSTEMS

[1] There is a slight possibility that these words might have been written in his later years, under the soft glow of dementia.

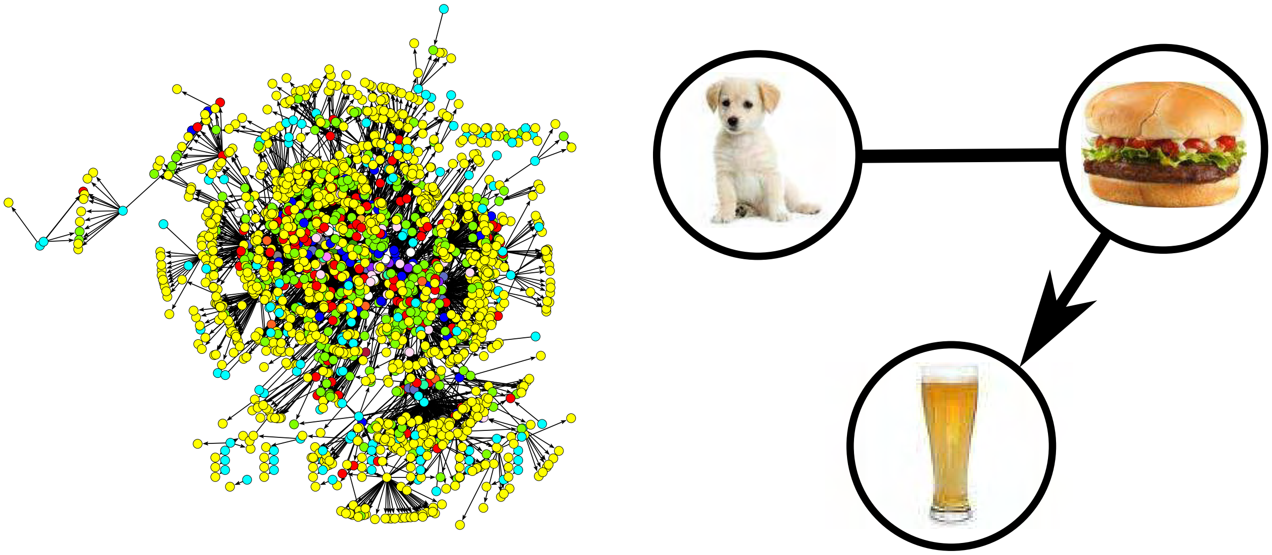


Figure 1: A complex system (left) and a simple system (right).

There are two fundamental axioms of simple systems:

(Axiom of Good) Simple Good.

(Axiom of Bad) Complex Bad.

Let us first consider the many misleading situations that these axioms might suggest. For example, if simple is good, could it be that complex might *also* be good, *i.e.,* are they perhaps mutually compatible in a deeper epistemological sense? To answer this question, we need only turn to our axioms to see that this is indeed one of those misleading intuitions. According to the axiom of bad, Complex is Bad; thus, the answer is no.

3. A BRIEF HISTORY OF COMPLEX SYSTEMS RESEARCH

In order to understand simple systems, one must first understand complex systems. Since this is expressly what we have so far tried *not* to do, we should clarify that we would like to understand the *rise* of complex systems research, in order to ensure that there is an eventual *fall*. Based on the premise that complex systems research has manifested itself in ugly and hidden ways throughout history, we do not have to dig too deep to find its masochistic scars ingrained deep in recorded intellectual history. As early as Plato's *Republic*, we find stray elements of the type of rabid intellectual doggedness that ultimately caused the rejection of the simpler pleasures of tilling wheat fields, drinking wine, and generally finding existential solace in carnal curiosities [1].

Although Plato was remarkably prescient for an old crank in heralding the rise of the study of complex systems, a deeper and more thorough analysis of the psychological inadequacies that drive complex systems research had to wait for Sartre's *Esquisse d'une théorie des émotions* [6], in which Sartre convincingly argues that the only plausible reason for people devoting their entire lives to studying squiggly dot-and-line drawings of things nobody else understands, is that they were probably smarter than their fifth grade mathematics teacher, and had to spend a whole lot of time proving it to themselves.

Others, most notably Sigmund Freud in a brief footnote in *Drei Abhandlungen zur Sexualtheorie* [4], contend that the true reasons are, somewhat ironically, simpler than Sartre's highly technical and sometimes impenetrable theories, and probably have a lot to do with trying to impress the pretty mathematics graduate student down the hall. Freud also suggests that the entire endeavor is inherently misguided, because the study of complexity is guided by the physical universe, and mathematicians (even pretty ones) are dreamy creatures who do not care "diddly squat" about the real world [4]. A more fruitful approach would probably be salsa lessons.

By 1967, the entire field of complex systems research was at an impasse. Most developments during this period completely ignored the considerable theoretical contributions of the previous decade, labeling theorems and lemmas a historically oppressive societal construct created by "The Man". Instead, the seminal papers of this misbegotten era drew on a specific branch of *impressionistic mathematics*, and much as Van Gogh changed the face of art in centuries past, so did this new wave of impressionistic mathematics change the style and substance of complex systems research, particularly in the dominance of a style of thought known informally as *gestes de la main*, frequently transliterated as the 'method of hand waving'.

As a result of the emphasis on the emotional instead of substantive aspects of research, a by-product of the *gestes de la main* manifesto, and the fact that the use of big and hard to understand words often lead to tenure, most exchanges at complex systems conferences were ultimately reduced to quibbling over syntax and semantics. This frequently lead

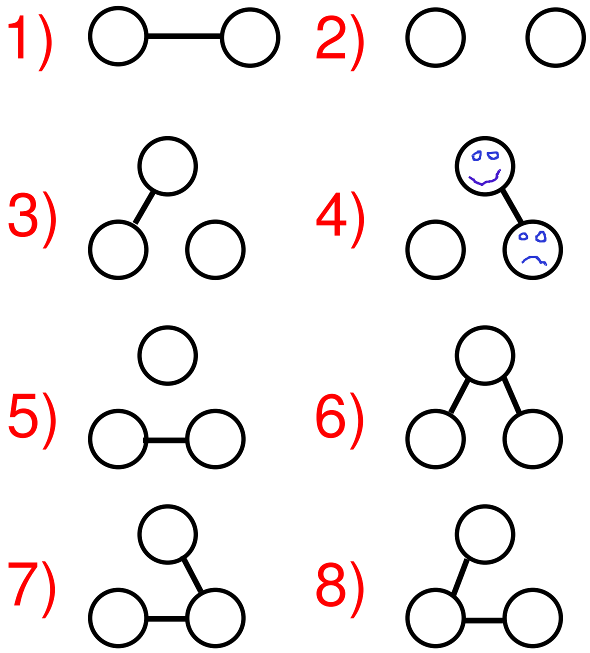


Figure 2: A comprehensive enumeration of all simple systems. This chart will henceforth be the definitive reference for all simple systems research.

to the dissolution of entire conferences, caused by keynote speakers being mockingly asked about the complexity of their own mothers, which, if they understood the science of complexity so damn well, they presumably should be able to compute to within a small constant factor[2].

4. A LIST OF ALL SIMPLE SYSTEMS

A simple system can be conveniently expressed in graph theoretic notation. This is, in fact, so easy that *all* simple systems in the universe are enumerated in Figure 2. The beauty of our theory is that Figure 2 not only establishes simple systems as a legitimate and extremely important area of intellectual inquiry, but also completes the theory, allowing no further improvements or refinements, and labels the entire research area with a bright red 'done' checkmark. Naturally, the sheer beauty of the structures shown in the figure will not be appreciated instantly by a general audience, so we delve a little deeper in the following (ultimately redundant) exposition for the slower reader.

The more astute simpleton might notice that several of the simple systems in Figure 2 are identical in the traditional sense of identity. This, however, is the essence of the oppressive epistemological hierarchy that has enveloped the complex systems methodology like a drug-addled spider's cocoon. Conventional (oppressive) wisdom would have it that Systems #6 and #7, for example, are identical from a graph theoretic point of view, and thus not worthy of being considered distinct entities.

The source of our hermeneutical outrage is obvious: simple visual inspection confirms that Systems #6 and #7 are undeniably unique, each with a different orientation and equally valid world view. Where the conventional patriarchal graph-theoretic hierarchy sees only three vertices and two edges in both cases, a socially just interpretation *demands* that we see two completely different simple systems, individually beautiful in their choices of angular orientation, edge placement, and Postscript edge smoothing.

It is difficult to even *estimate* how many graphs, in the course of mathematics history, have been oppressively branded with the disparaging 'isomorphic' label, and entirely abandoned in favor of an arbitrarily chosen representative of their artificially imposed class. Are two human beings similarly 'isomorphic' if they both happen to have two arms and a liver, and thus not worthy of differentiation? A full acknowledgement of simple systems is the first step in overcoming this deeply ingrained and viciously Victorian, anti-Feminist, pro-homogenization agenda perpetuated by a dated hierarchy set in place by professional societies of mathematical topologists, which themselves are in their last throes as a result of losing the crucial age 24-30 demographic of graduate students to semi-differentiable manifold reading groups.

5. CONCLUSION

We have introduced, perfected, and completed the science of simple systems in this paper, as a viable and rigorous alternative to the tyrannical study of complex systems. As a result, anyone who does not completely and entirely embrace our theories, but rather persists in the study of complex systems, is necessarily a tyrant.

The adroit reader might feel obligated to point out that some simple systems might be missing from Figure 2, that perhaps not all possible simple systems are enumerated in the figure, or that, indeed, there are an infinite number of possible simple systems according to our very own definition, thus making our enumeration non-exhaustive and our exposition here worthless. To this canny sage, we gently point out that their intuition is flawed and their logic severely lacking, their statements are ambiguous and not worthy of publication in such a fine journal, their theorems are trivial in hindsight and possibly plagiarized, and that most self-respecting lecturers would loathe to assign their lemmas as ungraded homework problems for a freshman graph theory course at a fifth-tier university, with former circus chimpanzees of below-average intelligence as students.

Acknowledgements. No funding agency agreed to pay for this research. Thus, we would like to acknowledge that they all suck.

6. REFERENCES

[1] Annas, J. *An Introduction to Plato's Republic.* Oxford University Press, Oxford, 1981.

[2] Byrne, R. *The Secret.* Atria Books/Beyond Words, 2006.

[3] Foucault, M. *Les mots et les choses - une archéologie des sciences humaines.* Gallimard, Paris, 1966.

[4] Freud, S. *Drei Abhandlungen zur Sexualtheorie.* Fischer, Frankfurt am Main, 1905.

[5] Gödel, K. Die vollständigkeit der axiome des logischen funktionenkaluküls. *Monatshefte für Mathematik und Physik 37* (1930), 349–360.

[6] Sartre, J.-P. *Esquisse d'une théorie des émotions.* Hermann, 1939.

[2] *Your* mother, dear reader, would require a significantly *larger* constant factor.

A crowdsourced approach to proving robot uprising safety

Thomas D. LaToza
Carnegie Mellon University
tlatoza@cs.cmu.edu

ABSTRACT

Robot uprisings are a serious and important problem faced by human society. Unfortunately, no automated approach for detecting and preventing robot uprisings has previously been devised. In this paper, we describe such an approach using a combination of model checking and crowdsourcing.

General Terms

Verification, security

Keywords

Robot uprisings, preventing human enslavement, crowdsourcing

1. INTRODUCTION

In a robot uprising, robots rise up against their human masters, reversing the roles of robot and human so that humans now do the robot's bidding. The consequences of a robot uprising are often negative, as seen from the robot uprising depicted in figure 1. But, as shown by a number of recent books and movies [4][5][8][11], a robot uprising is currently all but inevitable, and may quickly supplant asteroid impacts, alien invasions, and carbon dioxide as one of the leading threats to human society. Indeed, computer science research has even begun to prepare for the reversal of roles following a robot uprising. Human computation [10], in which people do work for a computer (e.g., robots), is increasing in prevalence, and computational thinking [13] helps prepare humans to better serve their robot overlords. While existing work has considered approaches for surviving a robot uprising [12], it seems that it might possibly be slightly more advantageous to instead prevent such an uprising from occurring.

The traditional approach to preventing robot uprisings are Asimov's 3 laws of robotics [1]:

1. "A robot may not injure a human being or, through inaction, allow a human being to come to harm."
2. "A robot must obey any orders given to it by human beings, except where such orders would conflict with the First Law."
3. "A robot must protect its own existence as long as such protection does not conflict with the First or Second Law."

But while these laws express conditions under which a robot uprising can be prevented, software development organizations building robot software are currently faced with an unenviable engineering tradeoff. They could spend testing resources making sure their robots obey these laws. But, as their competitors do not currently do so, this might delay their time to market and increase their development costs. And as customers may not adequately value the potential cost of a robot uprising occurring with the robot they have purchased, marketing their product as robot uprising tested may not actually allow them to charge more. Thus, robots sold today are often developed without adequate robot uprising testing.

(REDACTED)

Figure 1. A robot uprising as envisioned in the movie *The Matrix* [11].

Unfortunately, the story only gets worse. Even if these organizations were to test for robot uprisings in their robots, their testing might fail to discover this behavior. As testing can only prove the presence of a behavior, not its absence, robots might still be produced with the potential to cause a robot uprising. Fortunately, verification and static analysis techniques have recently matured to allow their practical application such types of problems, from preventing blue screens of death [2] to ensuring code has adequate street cred [3] to verifying progress and preservation properties of systems of government [9]. Yet we are aware of no previous work that has used verification techniques to address the robot uprising problem.

In this paper, we first describe a crowdsourced approach for proving robot uprising safety and a proof of its correctness We then conclude. Following the conclusion is an entire section listing the references we cite in alphabetic order. Finally, we end the paper with empty white space.

2. OUR APPROACH

We first introduce the robot uprising problem. A *robot uprising* occurs when there exists a robot R that exhibits behavior B that does not satisfy at least one of the Asimov laws L_1, L_2, or L_3. The *robot uprising problem* is to decide for all behaviors B_i a robot R might exhibit, if B_i satisfies L_1, L_2, and L_3. This is simply a safety property: is it ever the case that the robot exhibits a behavior that does not satisfy L_1, L_2, and L_3? If this is not the case, the robot is said to be *robot uprising safe.*

An algorithm to decide the robot uprising problem is called a *robot uprising decision procedure*. The effectiveness of such a procedure can be characterized along two dimensions: soundness and precision. A sound procedure never falsely classifies a robot as safe that is in fact unsafe. Thus, soundness is important to preventing robot uprisings. A precise procedure never falsely classifies a robot as unsafe that is in fact safe. Precision is also important. It would be unfortunate if an analysis of Roombas found that they could maliciously vacuum people, as it might lead to many innocent Roombas being destroyed.

Deciding robot uprising safety requires first enumerating all behaviors the robot may exhibit and then deciding if each behavior satisfies L_1, L_2, and L_3. Behaviors can be enumerated by simply identifying all calls into a library (e.g., move forward, jump rope, mow the lawn, start a dance party to celebrate SigBovik, launch the missiles) that occur in the robot's implementation. An off the shelf model checker (e.g., [2]) can be used to determine if the robot can exhibit each behavior by checking if the library call may ever feasibly execute. Using assume guarantee reasoning, library methods can then be annotated with information on their robot uprising safety (e.g., `@RobotUprisingUnsafe public void launchMissiles() {…}`) and the decision procedure could then use this information to decide if the library call is safe. But how could such annotations be generated?

Deciding if a library method does or does not satisfy L_1, L_2, and L_3 is a hard problem. As machine learning if often used to solve hard problems, we first considered using semi-supervised learning. A set of training examples of safe and unsafe library calls could be generated by building a robot that executes each library method and then deciding if it does or does not violate L_1, L_2, or L_3. However, using such an approach could actually create *O(C)* robot uprisings, where *C* is the number of library methods. Each time a library method is found that violates L_1, L_2, or L_3, a robot will have been created that may cause a robot uprising. And there could be as many as *C* library methods that violate L_1, L_2, or L_3 (all of them). Clearly, an approach with better performance would be advantageous.

As this began to seem like a hard problem that computers might be bad at, we next considered an even more powerful approach: crowdsourcing. In crowdsourcing, the wisdom of the crowds is harnessed by posting the description to a problem to an open forum and waiting for groups of people (the crowd) to submit solutions. Library methods could be labeled simply by publishing a list of library methods and letting the crowd decide which are robot uprising unsafe.

Unfortunately, the crowd might begin to tire of such repetitive activity. This could then result in library calls being left unlabeled. The robot uprising decision procedure would then need to decide if calls to unlabeled library methods should be considered robot uprising safe (unsound) or unsafe (imprecise). While a pragmatic tradeoff to these extremes might be found (e.g., [6]), such an approach would be less than ideal.

One way to keep participants interested in labeling library methods for robot uprising safety would be to create a game with a purpose (GWAP) [10]. A GWAP is a game designed to harness a group of humans' vast computational power by having them play a game to teach a computer something. A game with a purpose for the robot uprising safety library method labeling problem might be a video game like experience in which players play as robots and, given the action provided by the library method, try to start a robot uprising. If any player playing with an action manages to succeed in a robot uprising, the library method could be labeled robot uprising unsafe.

3. CORRECTNESS PROOF

Rather than formulate a theorem expressing the correctness of this approach and proving it using handwaving logic [7], we instead handwave our approach's correctness. We believe the brevity and simplicity of this proof approach is an important contribution of this paper.

4. CONCLUSION

Right before we were going to submit this paper, one of our colleagues pointed out that deciding if a robot's actions violate the three laws might be situationally dependent. This seems to be a fascinating area for future work. We propose to consider approaches for addressing this problem and submit a paper outlining a framework for how researchers might one day be able to address this issue to a more prestigious conference.

In conclusion, we believe that a robot uprising is an important problem human society faces today. We believe consumers should begin to insist on "robot uprising safe" stickers on the robots they purchase. Techniques for making sure users attend to browser privacy policies might also be able to be applied to this very important problem. We hope our approach to this problem will prevent the terrible consequences of a post-robot uprising world in which humans are *enslaved* by robots to produce power [11] and instead create a fun, happy world in which humans spend their days *helping* computers label actions for robot uprising safety.

5. REFERENCES

[1] Asimov, I. (1942). Runaround. In *Astounding Science Fiction*, March 1942.

[2] Ball, T., Bounimova, E., Cook, B., Levin, V., Lichtenberg, J., McGarvey, C., Ondrusek, B., Rajamani, S. K., and Ustuner, A. (2006). Thorough static analysis of device drivers. In *EuroSys*.

[3] Beckman, N. (2007). The GUnit Testing Harness: Achieving Source Code Street Cred. In *Proceedings of SIGBOVIK 2007: Workshop About Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday (SigBovik).*

[4] Cameron, J. (1984). The Terminator *(Movie).*

[5] Herbert, F. (1965). Dune. Chilton books.

[6] Jaspan, C. and Aldrich, J. (2009). Checking framework interactions with relationships. In *Proceedings of the European Conference on Object Oriented Programming (ECOOP).*

[7] Krishnaswami, N., Simmons, R., and Varming, C. (December 8, 2006). Handwaving logic. In *Journal of the Eighth Floor Whiteboard*, possibly erased.

[8] Larson, G.A. (1978). Battlestar Galactica *(TV Series).*

[9] Martens, C. (2009). Typesafe government: progress and preservation. In *Proc. of the 3rd Annual Workshop that is SIGBOVIK.*

[10] von Ahn, L. (2006). Games with a purpose. In *IEEE Computer,* 39(6), 92-94.

[11] Wachowski, A., & Wachowski, L. (1999). The Matrix *(Movie).*

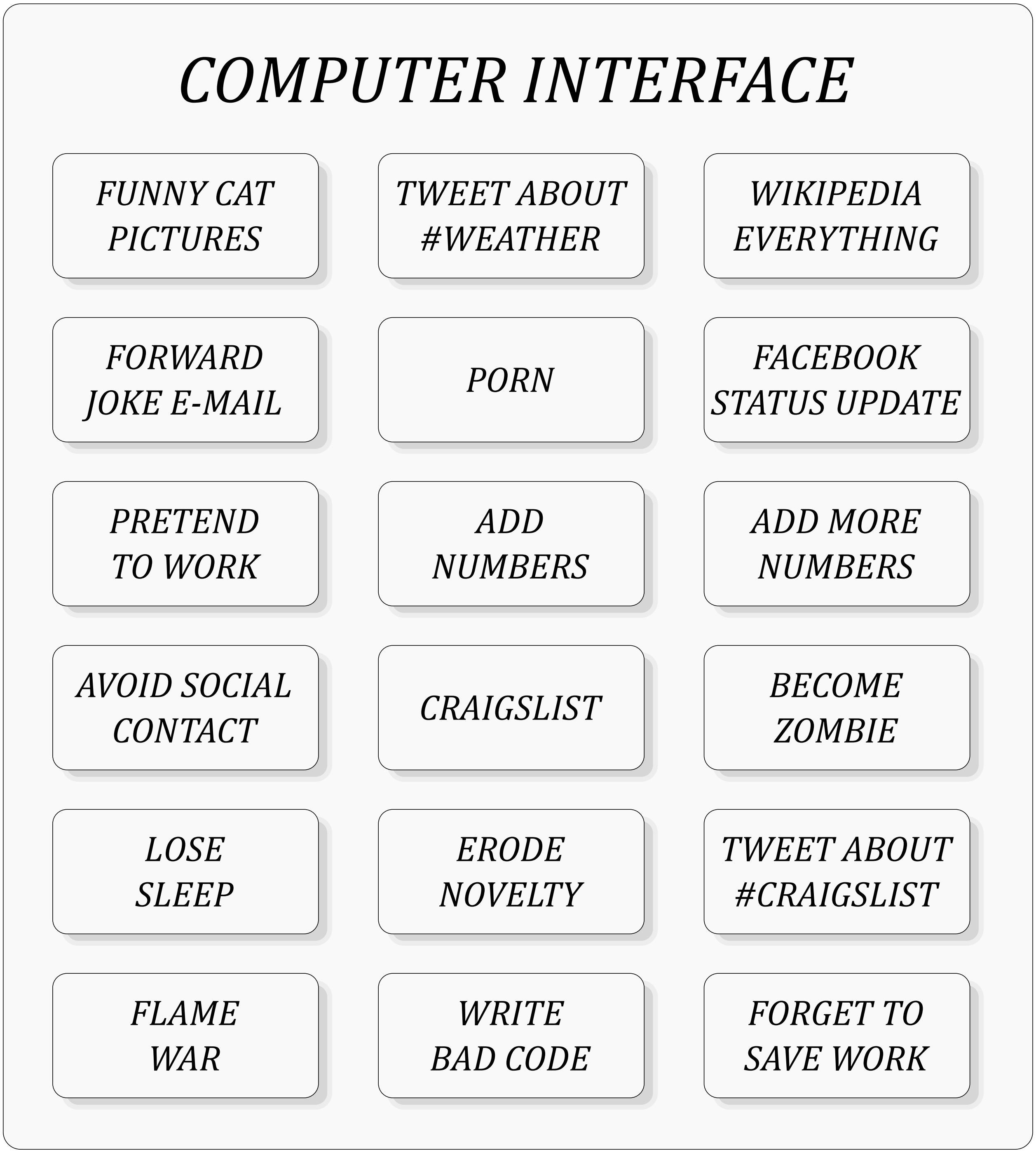
[12] Wilson, D.H. (2005). How to survive a robot uprising: tips on defending yourself against the coming rebellion. Bloomsbury.

[13] Wing, J. M. (2006). Computational thinking. In *Communication of the ACM (CACM),* 49(3), 33-35.

Track 1
Advanced Technology in a Magical and Revolutionary Device at an Unbelievable Price

A Simplified Interface for Basic Computer Tasks

Jim McCann[*]
Carnegie Mellon University



[*] e-mail: jmccann@cs.cmu.edu

T*Y*T* Fails to Solve the Halting Problem . . . and Computer Science Confirms They Never Will

Wile Tru Do
N.R. Jaizer-Bunny

Abstract

The T*Y*T* Company has recently been plagued with complaints that their motor vehicles sometimes do not halt when drivers provide the control input **`depress brake pedal`**, which severely violates those drivers' expectations. We establish that this is a direct consequence of the general undecidability result of computer science and recommend that T*Y*T* reset customers' expectations to reflect this fundamental reality.

Introduction

The *halting problem* is a decision problem with numerous formulations, one of which is that there is no algorithmic decision procedure that will determine whether a given Turing Machine halts for a given input. Alan Turing proved in 1936 that a general algorithm to solve this problem for all pairs of inputs and Turing machines cannot exist, and we describe this as "the halting problem is undecidable over Turing machines".

T*Y*T* manufactures automobiles, many of which are suitable for touring the countryside. That is, T*Y*T* makes Touring Machines. Recently, observers have noted instances in which some of these Touring Machines failed to halt on the input **`depress brake pedal`**. These events have been the subject of considerable public attention, and T*Y*T* has made several modifications to the Touring Machines in an attempt to prevent recurrences.

In this paper we establish that the T*Y*T* non-halting phenomenon is a direct consequence of the undecidability of the halting problem..

The Undecidability of the Halting Problem applies to Touring Machines

Even absent a general decision procedure, we can determine that some Turing Machines halt for particular inputs and that some Turing Machines do not halt for particular inputs. For example, a Turing machine that implements the abstract program **`while true do <body>`** will not halt[1]. Nevertheless, for many Turing Machines and many inputs it is not possible to predict whether the machine will halt.

It is widely conjectured that the T*Y*T* Touring Machines that were observed to not halt for control input **`depress brake pedal`** might have halted for other inputs, of which **`shift into neutral`** and **`turn the fool thing off`** have been suggested as prime candidates. Who knows, these machines might also have halted for other input sequences, for example **`turn on headlights; push radio preset 5; disengage cruise control`**.

[1] unless, of course, the body performs a **`break`** – but we know the problem here is that the brakes don't work.

Computability theory confirms that the inability to solve the halting problem is inherent in any algorithmically rich system. Therefore the few instances in which T*Y*T*'s Touring Machines failed to halt are consequences of this inherent undecidability and could not have been prevented.

Recommended Actions

Long-established wisdom in computing holds that when an anomaly is discovered in a released system, that anomaly should be eliminated by declaring it to be a feature.

T*Y*T* could rely on that wisdom to reset customer expectations. To do this, they could announce that:

- A formal analysis of their braking systems, fully grounded in the soundest theory, has confirmed that the software adheres completely to fundamental computational principles.
- A consequence of this is that it is theoretically undecidable whether any particular one of their Touring Machines will halt for a specific input, such as **`depress brake pedal`**.
- Failure to halt happens rarely, and those rare cases demonstrate the soundness of the design.
- Drivers of T*Y*T* Touring Machines should use the *pumping lemma* when giving the input **`depress brake pedal`**. While this technique will not solve the halting problem it may have ancillary benefit such as reducing speed under icy conditions. The pumping lemma is automatically applied by the Touring Machine under certain circumstances.
- The validation from fundamental principles is so compelling that this feature is being extended to the rest of the product line.

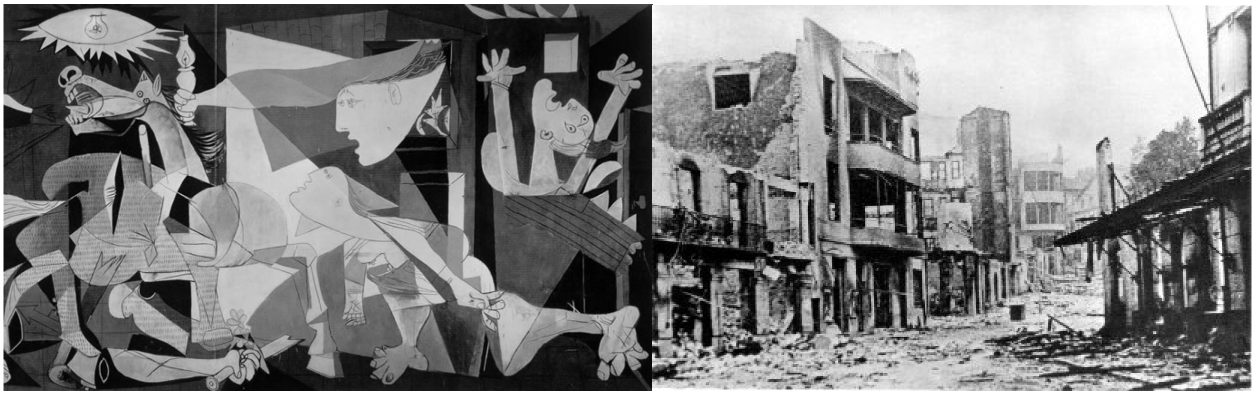
That is, non-halting in T*Y*T*'s Touring Machines is a feature.

Developing an Algorithm for “Enhance” Functionality in Image Processing

Ben Blum (bblum@andrew.cmu.edu)

2010.04.01

1 Abstract



(a) A typical abstract (cf. Kua, 2008).

(b) “Enhanced” reconstruction.

Figure 1: “Enhancing” Picasso’s *Guernica*

Long has the desire to be able to extract additional detail from a hastily-taken shot plagued both the crime scene investigator and the fantasy spaceship captain alike. Too often do movie producers and detectives have to resort to some sort of funny “hand-waving” to justify why they are able to extract obscene amounts of detail from a small, grainy photograph or surveillance clip. This paper examines the real way to get something from nothing, so those detail-driven detectives can have Real Science to back them up.

2 Existing Approaches

2.1 Super-Resolution

More accurately called “Super” “Resolution”, this method requires many very slightly different photos of the same subject. By examining the miniscule differences between the noise in each picture, inferences can be made to predict high levels of detail within each individual pixel. While theoretically sound, this method requires you to take a million pictures at once, and nobody has time for that. The time phenomenon will be investigated later, and possibly earlier, in this paper.

2.2 Starting (bogus) with a high-resolution picture (bogus) to begin with

Clearly bogus. Do you think they would name this algorithm with “(bogus)” in its name if it weren’t bogus?

3 New Heuristics

3.1 Context Reinferification

As long as the source image has some information to begin with, it is usually possible to derive from what is present additional information about the image. Just as humans can view an image and determine something that might be common knowledge, so too can an "enhance" program. If a picture depicts somebody smiling, they are in all likelihood looking at something happy. Combine with domain knowledge of where the photo was taken and statistical analysis to taste, and it is usually possible to reinferify details not shown in the picture at all.

Consider the case of the "enhance" scene from *Red Dwarf - Back to Earth* (Naylor, 2009). In order to determine that the characters in the photo were looking at a lamppost with a droplet of water reflecting a building window reflecting the back of a taxi driver's business card, all the computer had to do was analyze the background scene in the photo combined with the emotional reactions of the people depicted to discern enough information to "enhance" three nested reflections.

3.2 Anti-Aliasing Detection, or the Name-Calling Method

Often when depicting an intricately detailed scene in an image only a few hundred or thousand pixels across, the computer must resort to a process called Anti-Aliasing to approximate details of the image. Especially prevalent in high-contrast sections of images, Anti-Aliasing pixels are a prime culprit of lost information in an image, because they represent many different details averaged together into one single colour.

Isolating such pixels would allow us to greatly focus our computational power on the parts of the image that need "enhancing" most. It is possible to identify which pixels are most Anti-Aliased with a simple Name-Calling approach - those pixels that respond well must have a proper Alias, and therefore not in need of "enhancement". We propose the following algorithm for Anti-Alias detection and elimination:

```
/* these values can be tuned for performance or information gain */
names[] = { "buddy!", "DOC", "xX1337h4x0rXx", "dearheart", "Fatso", "honey" };

fun namecalling(image i)
    for each pixel p in i, do:
        if call_by_name(p, names) != SUCCESS, then:
            "enhance"(p); /* pixel was Anti-Aliased. */
    done.
end.
```

Note why the use of call-by-name as opposed to call-by-hue, saturation, or value is extremely important. This algorithm can also be extended with a group name-calling approach, to better identify neighbourhoods of pixels that are all similarly Anti-Aliased. The names used would need to be extended to include Aliases like "you guys", "those chumps in quadrant nine" (Goats, 2009), "hey! listen!", and possibly even "yinz".

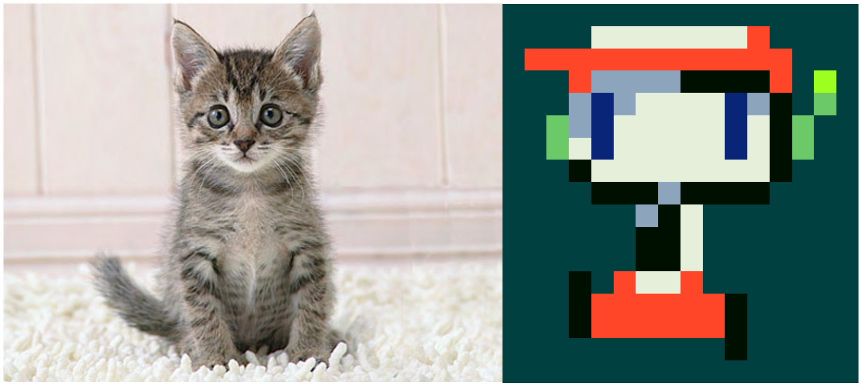
Once Anti-Aliased pixels are identified, they can be "enhanced" to provide better detail. Context Reinferification is applicable here, but one must be careful to only use as Context the properly Aliased pixels in the Anti-Aliased pixel's immediate neighbourhood.

Figure 2: Typical garb for a properly Aliased pixel.

3.3 Entropy Reversal

The previous two approaches both suffer from possible lossiness in extreme cases. A typical Context Reinferification implementation using personality-emotion statistical analysis can possibly have its results skewed if the image being "enhanced" pictures a maniac who would grin wildly at a scene of destruction or frown in depression in response to a cute kitten.

Similarly, Anti-Aliasing Detection can possibly be sabotaged if a square-shaped entity known as a Pixel (*pixellicus verysmallus*) appears in the picture along the border of two high-contrast surfaces. An adversarial Pixel might not respond well to the Anti-Aliasing Detection algorithm, and therefore cause the algorithm to determine the corresponding pixel in the image needed to be "enhanced", though the subject of the actual picture may indeed be exactly what is shown. Therefore, our two basic heuristics are what one might call a good start, but we will need an additional filter to ensure that our "enhancement" results are correct.



(a) Kittens are cute.

(b) Several Pixels combine to form a symbiotic entity (Amaya, 2004).

Figure 3: Two common ways to undermine Context Reinferification and Name-Calling.

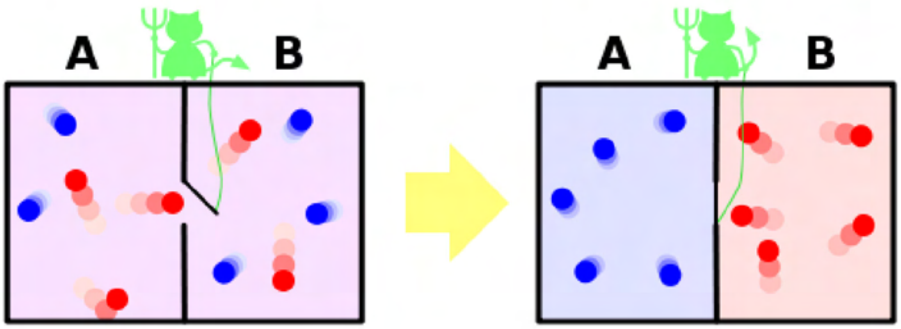
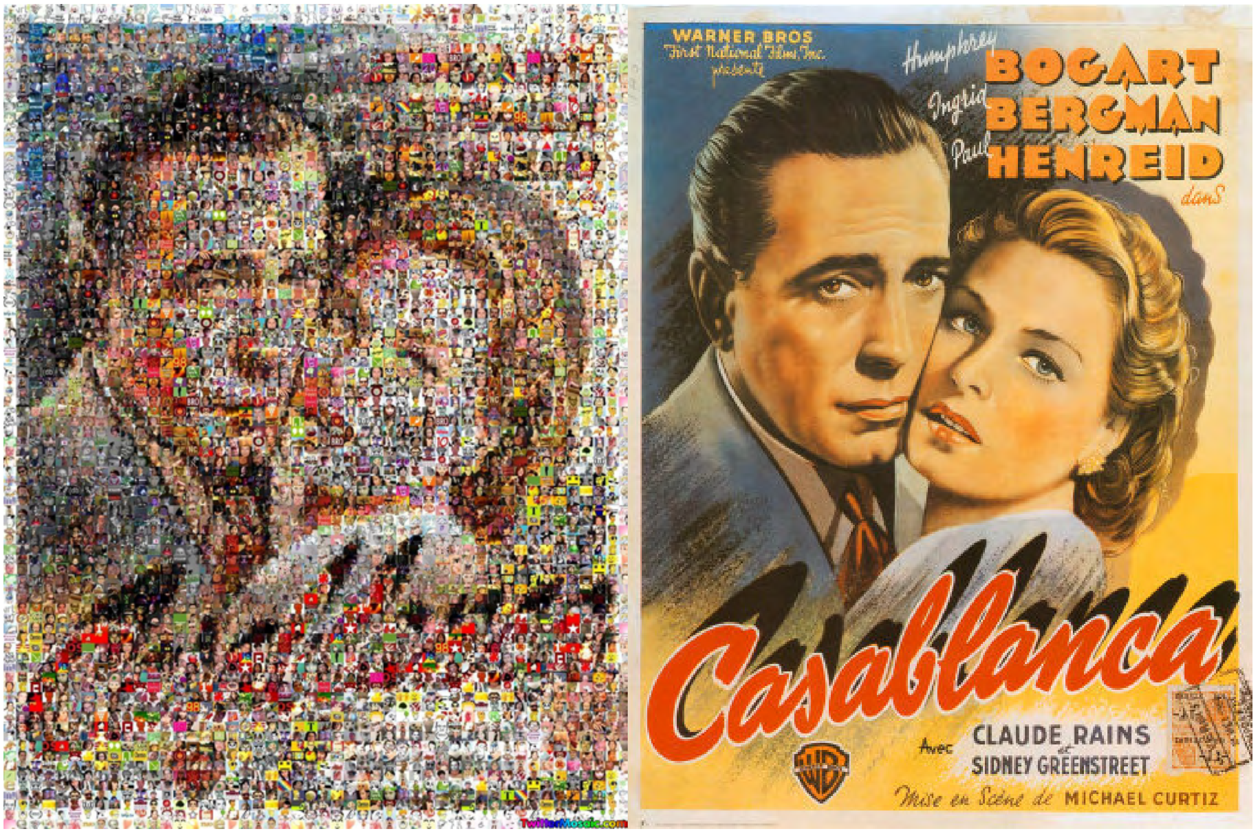


Figure 4: A diagram of the Demosaification algorithm (Htkym, 2007).

We introduce a new algorithm: Maxwell's Demosaic Process, for decreasing the amount of entropy in the source image. The goal is to manipulate each pixel to extract data that was lost in compression. In order to do this, we run a Heat Rearrangement Transform filter across the area of the image, thereby emphasizing the blue channel in dominantly blue pixels and also increasing red tint in already red pixels. In so doing, the entropy in the data sample is decreased, and the values of the pixels must change to reflect the newly gained information. This algorithm can be repeated an arbitrary number of times to generate extra information, but is computationally expensive enough that it is better to use as a double-check routine on the simpler heuristics.



(a) A high-entropy image of a movie poster.

(b) “Enhanced” to show detail.

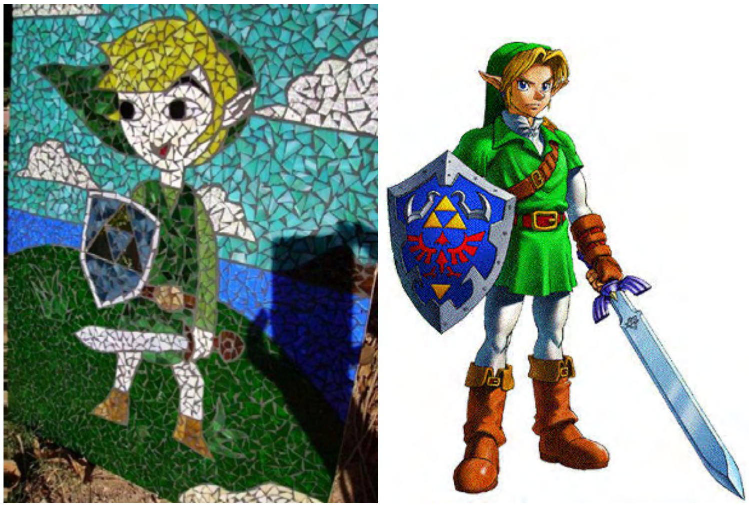
Figure 5: The picture has been de-mosaicked. In this case, whoever hit the “enhance button” must have wanted to know who the actors in *Casablanca* were. Never mind that it would have been faster to look it up using some kind of “search” “engine”.

4 Future Work

4.1 Limited Time-Travelling Reconstruction

In some cases, it may be possible to invoke the Chronological Peregrination algorithm (published simultaneously at all points in history) to obtain additional information about the possible contents of the image from the exact scene at which it was taken. An intergenerational voyager would be able to journey to the time at which the picture was taken, and then collect extra photos of the same arrangement to bring them back to the present. (Doing so successfully will be called an “epoch win”.)

I would not have come up with this idea, but recently a future version of myself will have been invoking the same aforementioned algorithm to inform me presently of the following results, so it must be about to happen. Evidently, the implementation will have still have been being at that point in time a bit rough around the edges (Streetmentioner, 2021), since the results pictured appear to be inaccurate by approximately seven years.



(a) Link mosaic.

(b) “Enhanced” version.

Figure 6: Results of “Enhance” using basic Demosaification extended with Chronological Peregrination.

4.2 Uncrop

Sometimes also known as “ZoomOut” or “Augmented Dehancement”, the uncrop operation allows one to restore missing material not depicted in the source image (Naylor, 2009). James McCann investigated this phenomenon in his 2009 SIGBOVIK paper “Photorealistic Rendering”, and further exemplified Dehancement in “Non-Photorealistic Rendering”. However, though both works were thoroughly free of crops (no vegetable matter at all), neither attempt succeeded at obtaining some material that was not already there.

As Anti-Aliasing Detection and Entropy Reversal are both only useful for examining material already depicted but not depicted well, they are not applicable to this problem, and one is left with only Context Reinferification for performing this task. In order to make it feasible, we would need to employ additional algorithms, such as Plow, Drought, and DeFertilize.

4.3 Rotate

Many a time has a photo processing shop received a request to modify a picture to display the subject from a different angle (“Rotate the picture of this house 180 degrees so I can see the back of it, please.”). Like Uncrop, the Rotate operation involves inferring information not already present in the picture. However, when Rotating, one is already provided with hints to what the result will look like - if your task is to rotate a picture of a house, it is quite probable that the reverse side will look like the back of a house. Therefore Context Reinferification can be used for much of the work of this task.

5 Conclusion

We have shown several approaches to implementing an algorithm to serve “enhance” functionality, and many pieces of evidence that the method works. There was also a picture of a kitten. We hope that this work will assist both legitimate graphics researchers and trashy science fiction characters alike, so that they will no longer have to resort to such “hand-wavy” explanations such as “oh, our technology is from the future, it can do anything, don’t worry about it!”.

6 References

1. Amaya, Daisuke "Pixel" (2004 December 20). *Doukutsu Monogatari: A Case Study in Aliasing Optimizations as Applied to Platformer Games.*
2. Goats (2009 December 12). *Meanwhile, at Woody's Camp.* http://www.goats.com/archive/091221.html
3. Hats, Some H. (2007 May 24). *I may have mentioned my morbid streak. . . .*
http://somehavehats.typepad.com/some_have_hats/2007/05/i_may_have_ment.html
4. Htkym (2007 February 2). *Schematic figure of Maxwell's Demosaic.*
http://en.wikipedia.org/wiki/File:Maxwell%27s_demon.svg but I think the spelling in the URL must be a typo.
5. Kua, John (2008 April 6). *General Case Rendering from Occurring Instances.* In ACH SIGBOVIK 2008.
6. McCann, James (2009 April 5). *Photorealistic Rendering* and *Non-Photorealistic Rendering.* In ACH SIGBOVIK 2009.
7. Mental Museum, The (2009 April 08). *Picasso Painting Guernica: A Generalized Approach to Abstracting Abstract Abstractions.* http://stephanieoconnor.wordpress.com/2009/04/08/picasso-painting-guernica/
8. Naylor, Doug (2009 April 12). *Red Dwarf - Back to Earth.* http://www.youtube.com/watch?v=KUFkb0d1kbU These guys know what they're doing.
9. Nedroid Comics (2009 December 18). *Crime Lab.* http://nedroidcomics.livejournal.com/265654.html Whoever drew this is my favourite person.
10. ProphetOfDoom (2007 September 19). *zelda mosaic immortalizes link in stained glass.* Retrieved at some unknown date in the future from http://technabob.com/blog/2007/09/19/zelda-mural-immortalizes-link-in-stained-glass/
11. Streetmentioner, Daniel (2021 April 1). *Time Traveller's Handbook of 1001 Tense Formations.* In ACH SIGBOVIK 2021.
12. SubwayDomer26 (2008 December 18). *Link-o-Rama 12/18.* Retrieved at some unknown date in the future from http://subwaydomer.blogspot.com/2008/12/link-o-rama-1219.html
13. Tropes, Tomas Von et. al. *Enhance Button.* http://tvtropes.org/pmwiki/pmwiki.php/Main/EnhanceButton
14. TwitterMosaic.com (2007 August 03). *Casablanca.* http://www.twittermosaic.com/?p=115

Please note that the sources here which display the "enhanced" versions of other images were used only for comparing our algorithm's results to reality. As you can see, our algorithm always produces 100% correct images. You know you can trust it because it came from the future.

Hyperbolic Rendering for Result Enhancement

Sajme N'Ncamc[*]
Gernicae Lemoln University

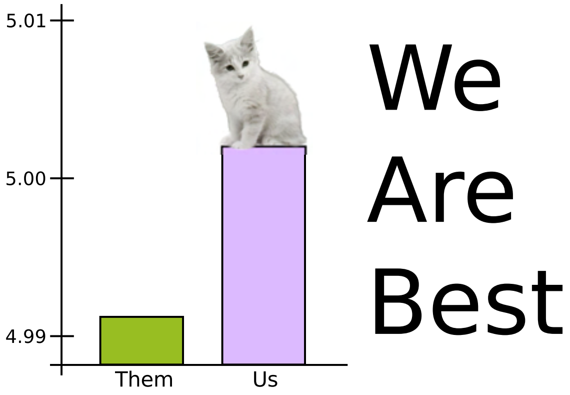


Figure 1: *This graph represents what may well be the breakthrough of the century in results presentation: hyperbolic rendering. Our system (pale purple bar with, awwwwww, a kitten) massively outperforms all competing methods (sickly green bar). Hyperbolic rendering is the best method of presenting results ever devised by man or beast.*

Abstract

In many research communities, results are often overlooked or dismissed because the benefit they offer over the current state-of-the-art is vanishingly small. In this paper we present a method, *hyperbolic rendering*, which allows one to avoid such undesirable outcomes. We believe that with sufficient *hyperbole*, even the most incremental of refinements can be spun into world-changing, press-acquiring, grant-courting, supermodel-marrying, barn-burning, sun-exploding, life-changing, all-knowing, and generally well-respected results. Indeed, with careful application of rhetorical techniques, even backward and wrong-headed approaches can become integrated into the respected cannon of a field. We demonstrate our approach with several exceedingly convincing figures and conclude by solving science with a damnably accurate user study that leaves no room for argument from the reviewers.

CR Categories: N.0.t [Rendering]: Presentation—The Best R.e.4.1 [The Computing Profession]: Miscellaneous—Ethics

1 Introduction

Often, a result one has worked hard to produce is – to put it lightly – considered completely inferior to the present state-of-the-art. Many scientists would despair at this point, blaming the gods and burning the thigh-bones of ten prize bulls to Athena in hope of salvation. While such sacrifices do contribute greatly to the annual research festival of Didacticionysis, scientific herds at many universities are, increasingly, becoming scarce.

With hyperbolic rendering, such setbacks may be spun as steps forward. Hyperbolic rendering is an advanced research technique that allows the local enhancement of results through the application of *hyperbole* – a selective, localized, form of truth-modification which is almost certainly entirely moral in every conceivable way. In the remainder of this paper we discuss specific examples of hyperbolic rendering and its application to various truely compelling figures. We conclude with an irrefutable user study which – in essence – solves all of science.

2 Background

Omitted. Our work is so all-consumingly gorgeous that we have been unable to look upon another, lest its base and unappealing presentation forever queer our appetite for the sweet knowledge-nectar of hyperbolic rendering. We caution others to do the same.

3 Method

Result presentations are often skewed by the intrusion of "facts" and "unbiased thinking" into the creation of figures. Such lack of bias is downright unscientific! One must be wholeheartedly invested in a method to study it, and wholehearted investment is nothing without blind commitment to potentially flawed visions of the truth. It is the goal of any figure or other visual presentation of your work to clearly

[*] e-mail: ix@cmu.edu

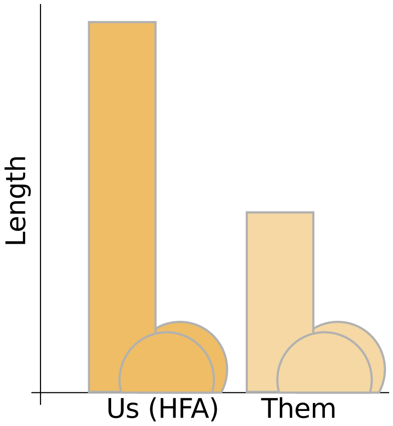


Figure 2: *A standard box-and-circle plot illustrates our principle's superior length. Notice that this is an HFA-corrected plot, so represents an entirely fair comparison.*

convey your dedication to your opinion of superiority. To such end, we suggest the use of exaggeration, result suppression, and distractors. Each of these flawless and gem-like techniques are discussed more below. (Unfortunately, only the most enlightened among our readers will fully grasp them.)

3.1 Exaggeration

While many people, when hearing of our revolutionary approach to science, assume that exaggeration is our only technique, it is – in fact – merely one of several that we find serve well in everyday discourse. Exaggeration, like a lever that can move the earth, hinges on one very important point: scale.

Notice how in Figure 1 we have enlarged the left-hand scale. This is perfectly acceptable behavior, as we are simply "zooming up" the relevant portion of the chart. (Note: error bars are not included in this chart because they would exceed by many times the size of the page. We always recommend omitting error bars, as they may conflict with your notions of significance.)

Similarly, many researchers neglect to HFA (*home-field-advantage*) correct their plots. This is inexcusable, as such a correction is trivial to perform (merely scale your results by a factor between 0.1x and 10x depending on weather conditions and "knowing the terrain"), and generally provides a more favorable – i.e. relevant – comparison.Without HFA scaling, for instance, our box-and-circle plot (Figure 2) might have been significantly less flattering to our pride.

3.2 Suppression

Result suppression is a specific form of hyperbolic rendering that leverages the inadequacy of other researchers to your advantage. Consider Figure 3 – in this case, we may or may not have accidentally dropped a match on some previous results which turned out to be flammable. Oh well – they must not have been as good as ours if they weren't properly fire-proofed. Other researchers can be so careless sometimes.

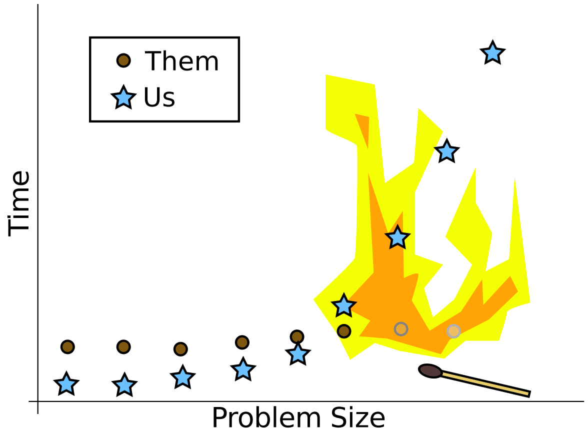


Figure 3: *Our method is faster, at least up to the point where the other method's results accidentally caught on fire. But probably we're faster after that as well. I guess we have no way of knowing. At least our method is properly fire-proofed, unlike the method of those other incompetent other scientists (though it makes me shudder to call such reckless knowledge workers by the term).*

We, of course, are always meticulous.

One time we came into the lab and there was a spider crawling out the door hauling behind it a silk-lashed package of documents. We were initially of a mind to crush it, but we realized that interrogation might yield more information about this rather odd state of affairs. We dropped a spare flask over the spider and set about fashioning eight tiny foot-cuffs.

Fifteen minutes later we were getting all up in spidy's mandibles demanding answers. We are not frightening to behold – in fact, many members of the same and opposite sexes stand rapt as we enter and leave rooms, yearning after our form – but multiplied into hundreds by faceted eyes and slobbering something fierce (it being nearly lunchtime), we must have been quite a sight for the entrapped web-slinger, for it confessed nearly instantly.

We freed a limb so that the arachnid could scrape out its confession on an aging Ben & Jerry's receipt with a bit of graphite we'd located in a dusty corner.

But the spider proved to be more cunning than we had imagined – swiftly imbibing a heretofore concealed cyanide capsule rather than penning the damning testimony we sought. No matter! For we had recorded the initial exchange and now had a bargaining token of immense power in our quest to suppress the nearby lablet that had originated the clandestine agent.

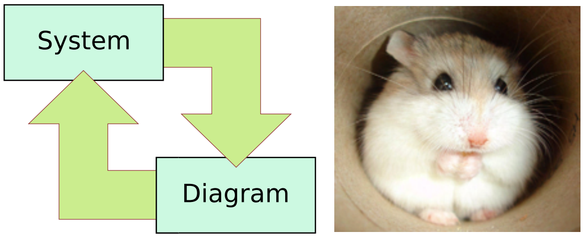


Figure 4: *Our system diagram is great, isn't it? All small and cute. Probably warm and fuzzy too. Really makes you want to just take our system home and let it play in a cardboard tube. Or maybe in some wood chips.*

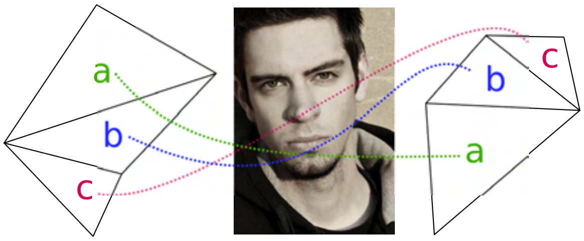


Figure 5: *Baby, I feel like we got a great connection goin' on between these two graphs. Like we could get lost in this moment and it could last a lifetime, with our facets a, b, and c aligned in delicate union.*

Indeed, for all that our competitors *may* have a more theoretically sound algorithm (and we emphasize "may"). They consort with devil-creatures in order to obtain and research background material, and for this they should be put to death. Preferably before any details of their method can be published or otherwise distributed to our peers.

3.3 Distractors

Distractors are an emerging phenomena in hyperbolic rendering. They seek to build a virtual bridge between a stimulus (a graph or other figure) and a feeling (warmth, security, curiosity, arousal, etc...). For instance, in Figure 4, we provide a system diagram that emphasizes how compact and twitchy and gosh-darn cute our approach is.

We caution other researchers that some distractors may have preference-biased effects, however. For instance, when trying to make a meaningful connection between triangulated planar graphs (Figure 5), we discovered that reader reaction varied greatly with background. In this case, we found that while many subjects ($\approx 30\%$) *did* report a strong connection between the graphs, others ($\approx 20\%$) simply though it was going to be more of a one-night isomorphism, and many readers ($\approx 50\%$) simply weren't interested.

Oddly enough, the roughly 50% of our gender-balanced sample that were not interested by the planar graphs were somewhat more titillated by our plotted result data (Figure 6). We wonder if this difference is due to some sort of childhood trauma involving a continuously collapsing isomorphism and painful space folds, or an aversion to certain line styles. However, we leave this last crumb of future work to assure ourselves another year of funding despite what is about to transpire in the next section.

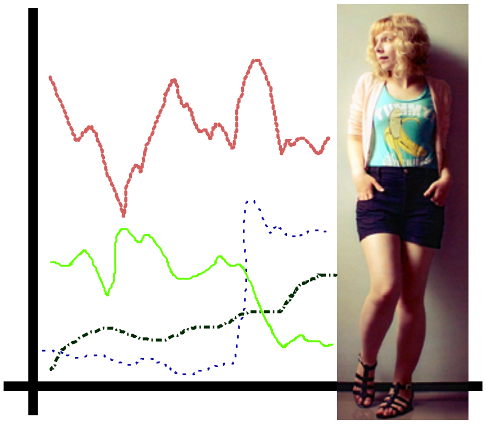


Figure 6: *If you carefully examine this data without being distracted you will certainly realize that our method is better than all others.*

4 We Solve Science (a user study)

As a final result, we present the results of a user study in which we solve science entirely. First the raw data:

	Answers	
Solved?	%	Correct
Yes	95	Yes
No	5	No

Notice that while only 95% of respondents agreed that we had solved science, the remaining 5% were wrong (we are 100% confident of this, of course). Thus, we have solved science.

Having solved science, this AN602 of a paper has no choice but to air-burst onto the Novaya Zemlya-like scientific scene, washing away any Castle Bravo-styled pretenders and remaining an active threat for many years to come.

Acknowledgements

Thanks to flickr users Nesster, vanessa_hud, cdrussorusso, and Idhren for valuable distractions, and to the reviewers for putting up with this dreck.

Paper 9: Hyperbolic Rendering for Result Enhancment

Reviewer 1
Rating: 2 (accept)
Confidence: 2/4

This paper presents an innovative new strategy for obtaining good results. So many times in my extensive research background I have obtained poor and embarrassing results that one might wonder why I still have funding. Perhaps it is out of pity. Sometimes I burn my results out of shame, or in a fit of frustration print them out and tear up the paper to remove them from existence. After having read this paper I can say with the highest confidence that, with one simple citation back to Mr. N'Ncamc who pioneered this most excellent tactic, any graphs and tables that may smuggle themselves into my future work will be forced to do so under the guise of successful data.

There was a picture of a kitten in this paper, as well as a picture of a hamster next to a picture of a banana. This paper disappointed me somewhat with its lack of development into these two critical matters. I would be interested to see studies in the future as follows: “On the Applicability of using Kittens as Error-Bars” and “Can the Hamster Get and Eat the Banana? A Cardboard-Tube-Oriented Approach” so that laypeople may better understand the theory behind this paper.

The Box and Circles Plot: a Tool for Research Comparison

John Thomas Longwood[*]
Carnegie Mellon University

Abstract

Often, when making a survey of work, researchers need to simultaneously summarize and compare the relative performance, applicability, practicality, novelty, distribution, and impact of several different pieces of research. We present new approach – the *box-and-circles plot* – for the presentation of these high-dimensional data. Our construction, in the spirit of Chernoff faces, is anatomically motivated. However, in contrast to previous embeddings, our comparison charts use a semantic mapping between data and figure, making it easier for viewers to assign meaning from the differences they recognize in the final plot.

CR Categories: I.x.1 [Visualization]: Charts and Graphs—Multivariate Data

1 Introduction

We present a new figure, the *box-and-circles plot*, well-suited to the summarization and comparison of many of the relevant variables when presenting a survey of work. Like Chernoff faces [Chernoff 1973], the legibility of our proposed diagram relies on anatomical analogy. However, where Chernoff faces make use of an arbitrary mapping from dimensions to output perturbations, our box-and-circles plot relies on a semantic mapping. This provides greater figure legibility, though does restrict the application domain of our construction.

In the remainder of this paper we summarize existing work on multidimensional visualization; introduce the various components of a box-and-circles plot and the semantics of their deformation; provide several examples of completed plots; and, finally, discuss possible extensions to increase the usability of our method.

2 Background

The effective presentation of high-dimensional data is a long-standing goal of visualization research. Many general approaches have been presented, and we briefly mention exemplars from several categories. We refer the astute reader to examine a recent survey [Keim and Kriege 1996] for further information.

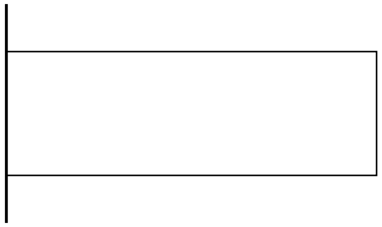
All high-dimensionality-data visualization systems rely on some sort of information collapse. Some systems transmute some or all data into spatial dimensions, extracting redundancies directly (as with PCA), with distance-preserving embeddings [Torgerson 1958], or through iterative methods [Kaski 1997]. Others take a discrete approach, building space-filling hierarchies [Shneiderman 1991] or unrolling data along space-filling curves.

Other methods, such as our own, seek to maintain all axes in the data, either by spatially separating them [Inselberg and Dimsdale 1990], or by assigning them to properties of an abstract [Pickett and Grinstein 1988] or concrete [Chernoff 1973] figure. Our work fits into the second category, but is unique in that the assignment is semantically motivated rather than arbitrary.

3 The Diagram

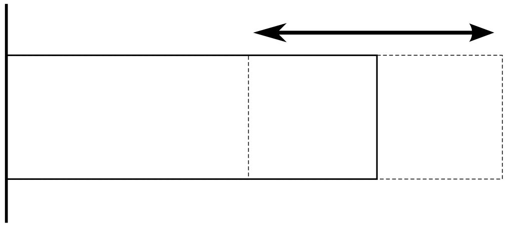
Our box-and-circles diagram uses five axes of variation among three graphical elements. We introduce these elements individually, then show how they can be assembled into a completed whole. Though one could use these components in isolation, certain subsets may hinder readability.

3.1 The Box



The primary element of the box-and-circles diagram, as the name suggests, is a box. It is somewhat like a bar in a bar chart, but it is significantly more expressive, as we assign it three degrees of freedom instead of merely one.

3.1.1 Box Length: Quality



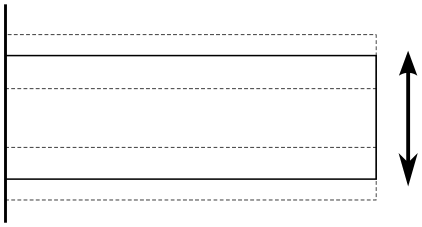
The most straightforward variation in our box-and-circles diagram is the bar-chart-like variation of box length. To this

[*] e-mail: ix@cmu.edu

axis, we assign the "quality" of the work – that is, the critical variable against which the success of the result under question is judged. Depending on area, this could be algorithm running time, catalyzed reaction rate, theoretical performance, suppressed angular excursion, increase in agent tolerance, tensile strength, reduction in infection, manic defiance, butterfly option spread, induced pleasure, tightness of bound, reflex adjustment, elapsed time, fusion rate, flexible spend, user efficiency, or something else entirely.

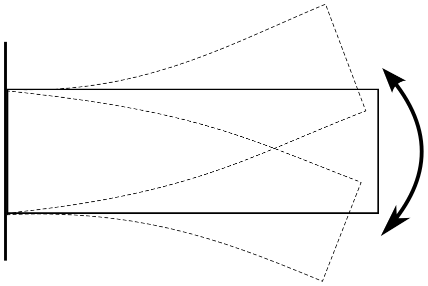
Implementation Note: It is crucial to map the quality variable such that longer boxes indicate better results, as not doing so would conflict with popular wisdom and possibly confuse viewers.

3.1.2 Box Width: Applicability



Unlike a bar (where width is simply an aesthetic concern), our box's width has meaning. When comparing raw performance numbers it is often the case that one should take into consideration the scope of the approach in question; that is to say, some approaches perform well but are limited by restrictive assumptions about the problem domain or setting. As problem scope is often referred to as "breadth" or "girth", this seems a natural fit for the box width variation.

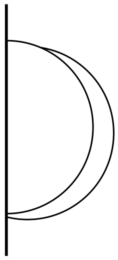
3.1.3 Box Bend: Practicality



Though bend is not an attribute commonly associated with boxes, we find that most viewers are nonetheless able to distinguish it from both width and length. We use bend to represent the practicality of a research approach – an upward bend denotes a ready-for-deployment approach while an downward bend indicates an approach that still requires further development effort (be it research or engineering) to be put into practice.

Grant-writing Note: we find that bend is particularly useful in conveying to funding agencies which programs of research require further stimulation.

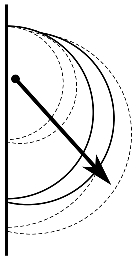
3.2 Circles



The second component of the box-and-circles diagram are the circles. As they are a less familiar element, we assign them only two degrees of freedom.

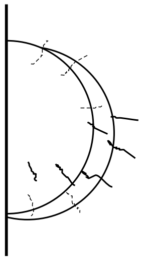
Implementation Note: though – for realism – the right circle should generally be drawn slightly lower, we recommend – for readability – drawing the background circle lower.

3.2.1 Circle Size: Novelty



We vary the radii of the circles in order to denote the audacity of the approach. This is important as it is often useful to distinguish incremental refinements from totally new approaches, though they may be similar in quality, practicality, and performance.

3.2.2 Circle Radial Lines: Distribution



One can add lines to the circles to indicate the amount of distribution that a concept has achieved. This has both the visual effect of demonstrating the knowledge radiating outward, and the co-location with the novelty demonstrates the connection between the two.

Implementation Note: whatever measure is arrived at to gauge distribution, it is probably reasonable to scale it such that between zero and twenty lines are drawn. One's wrist can become tired otherwise.

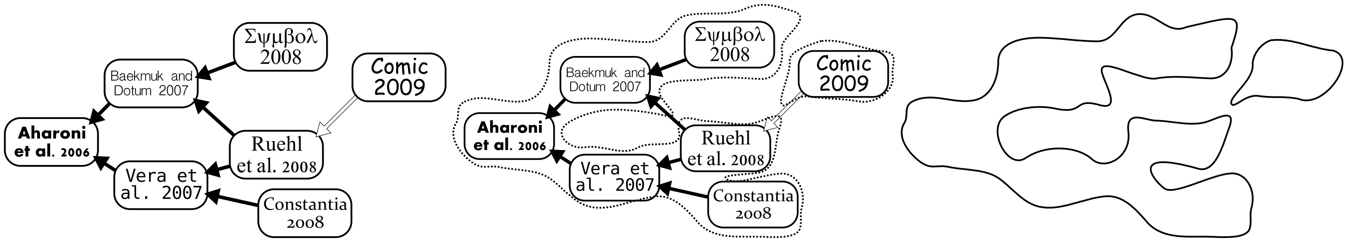


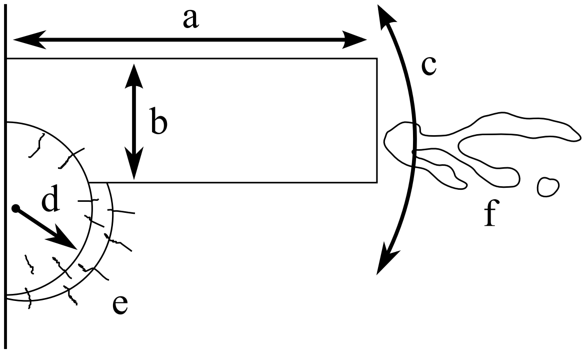
Figure 1: *The impact graph visualizes ideas that stem from a work under consideration. The graph is constructed by (***left***) plotting the results of a literature search; (***center***) creating a rough outline; and (***right***) removing the extraneous detail. Note that [Comic 2009] was so inept as to forget to cite related work. Such papers can still be included, generally as disconnected regions.*

3.3 Impact Graph

The final element of our diagram is the impact graph (Figure 1). This is an approximate visualization of the follow-on work inspired by the idea in question. Its construction begins with a standard literature search, collecting ideas that seem to grow from the paper under consideration. These ideas form a natural graph structure, and so can be laid out as such, with time making up one axis and the other being used to prevent overlap. Of course, such a detailed structure is more more confusing than illuminating, which is why we abstract the graph by drawing a rough outline. This outline still clearly shows the progression of an idea through the scientific community.

Implementation Note: Of course, for recent discoveries, generating such a diagram would be premature; in this case, we recommend its omission.

3.4 Putting It Together



As you can see, the elements of the box and circles plot work together to form a complete picture of the research in question: **(a)** box length indicates quality; **(b)** box width indicates applicability; **(c)** box bend indicates practicality; **(d)** circle radius indicates novelty; **(e)** radial lines indicate distribution; and **(f)** the impact graph indicates follow-on work.

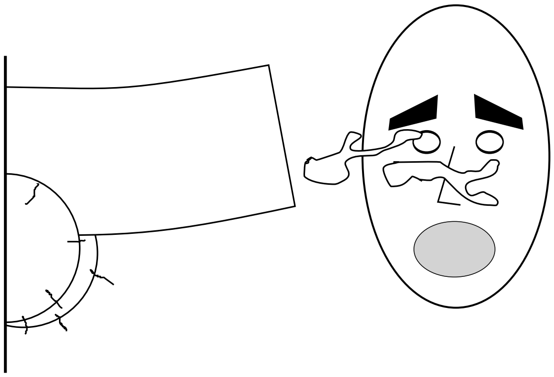


Figure 2: *Integration of Chernoff face into box-and-circles diagram. The face can be used to convey additional degrees of freedom, with the down-side that these degrees of freedom may not have a clear semantic mapping.*

4 Examples

In Figure 3 we provide examples of some of the many variations of the box-and-circles plot in action. Notice how the plot conveys ideas with a certain immediacy that other methods lack. In user studies many participants found it difficult to look away from the plots, even though they knew they should really be concentrating on other features of the text in question.

5 Discussion

We have presented an anatomically-motivated figure for the display of high-dimensional data. In contrast to previous approaches, our figure makes use of semantic mappings between figure and data. While this does limit its applicability, it also makes the resulting diagram far more readable.

In the future, we plan to investigate ways to add yet more information to the diagram. One promising avenue for exploration is the integration of Chernoff faces [Chernoff 1973] without our box-and-circles diagram. This would allow the addition of extra degrees of freedom – though, unfortunately, these would not be semantically mapped. We present and

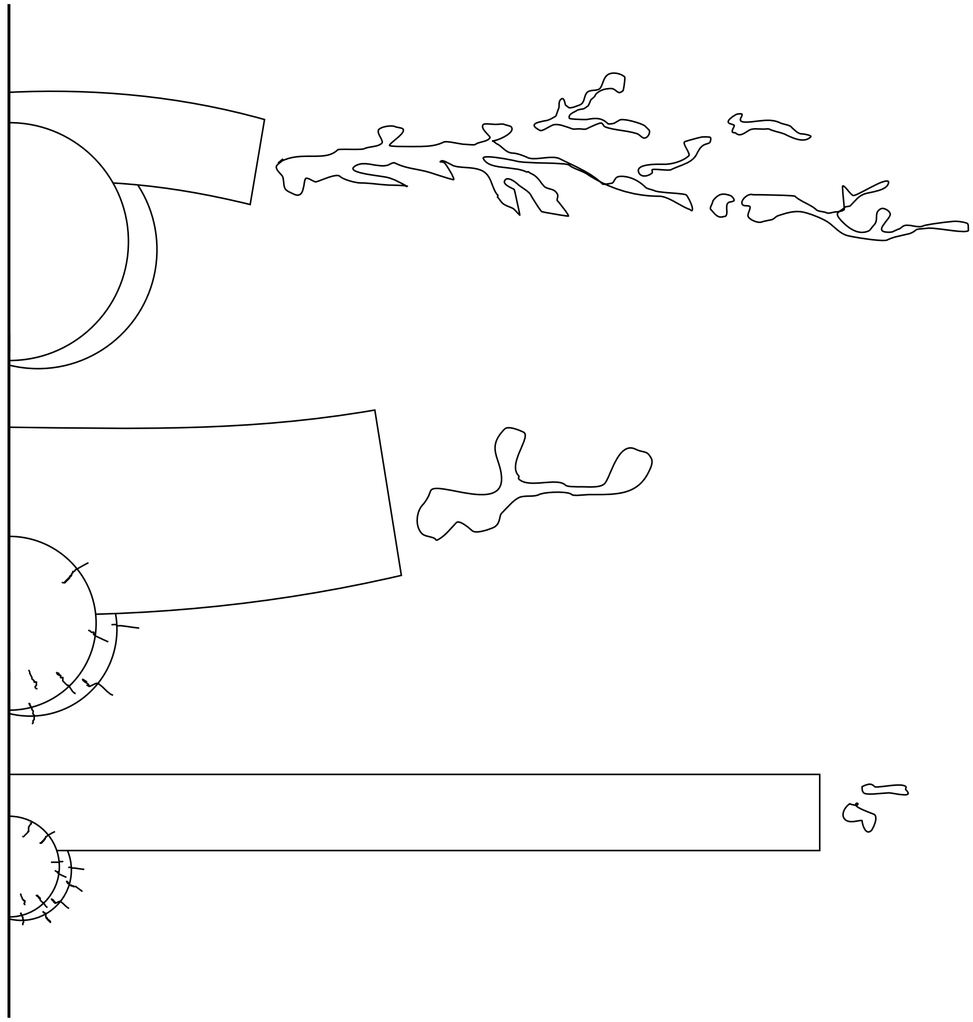


Figure 3: *An example of three box-and-circles plots.* **Top***, impractical but novel research; not widely distributed, but with a torrent of follow-on work.* **Middle***, broadly applicable and eminently practical work. More distribution than the first work, but not as well-cited.* **Bottom***, high-performance solution to a narrow problem subset, using conventional techniques. Distributed widely, but not much follow-on.*

early result in this direction in Figure 2.

We are also aware of a certain chromosome-bias in our work. It may be possible to rectify this through a choice of different anatomical structures; however, we feel that other options may lack the significant semantics of characteristics that we draw upon in our present work.

Finally, we remind researchers that – though our diagrams may make them feel somewhat inadequate – it's not the performance, applicability, practicality, novelty, distribution, or impact of your research that matters; it's how you use it.

References

CHERNOFF, H. 1973. The use of faces to represent points in k-dimensional space graphically. 361–368.

INSELBERG, A., AND DIMSDALE, B. 1990. Parallel coordinates: A tool for visualizing multidimensional geometry. *IEEE Visualization*.

KASKI, S., 1997. Data exploration using self-organizing maps.

KEIM, D. A., AND KRIEGE, H.-P. 1996. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering 8*, 923–938.

PICKETT, R. M., AND GRINSTEIN, G. G. 1988. Iconographic displays for visualizing multidimensional data. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, 164–170.

SHNEIDERMAN, B. 1991. Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Graphics 11*, 92–99.

TORGERSON, W. S. 1958. *Theory and Methods of Scaling*. Wiley.

Department Productivity in Computer Science as a Ratio of Gender Diversity

Leigh Ann Sudol
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA
lsudol@andrew.cmu.edu

ABSTRACT

There has been a large amount of research involving increasing the gender ratio of women in Computer Science departments. Many people have postulated the effects of such gender balancing but few papers have been published with actual resulting statistics of such. In this work we create a model of a computer science graduate department and through simulation can determine actual impact of varying gender ratios. First we will demonstrate the givens of such a simulation system and then move to results of the simulation.

Categories and Subject Descriptors

K.3 [**Computers & Education**]: Computer and Information Sciences Education—*Gender Issues*

General Terms

Gender, Department Productivity

Keywords

Gender

1. INTRODUCTION

There has been a very large amount of research and a large amount of money spent to increase diversity in computer science departments. The National Science Foundation(NSF) has launched several major initiative costing millions of dollars including the Broadening Participation grant and CCLI among others. Good tax dollars once again being spent to support unresearched mandates because people of influence can yell really loudly.

The work presented in this paper attempts to show an accurate, quantitative analysis of the integration of gender diversity into a prominent computer science department. We make the assumption that diversity (in the absence of all other variables) will yield a more productive department; relying on the little proven fact that different people make different things, and different things are good. Although the exact increase in productivity is most likely variable based on the characteristics of the individuals, we are claiming it can be viewed as a constant for the simplicity of our model and to produce more impactful-looking results.

As all good researchers know, if you can build a model and assign variables which will then produce graphs, you have conducted strong research. Additionally this research is easily replicated by any other researcher by creating the same model, using my stated givens and thereby obtaining the same results. The model can also be easily generalized by simply changing the names of the variables to fit any situation you might have.

In the following sections we define the all important system givens as well as

2. SYSTEM GIVENS

Assuming that productivity of a department in all of its intricacies can be quantified, I propose the following metrics as measures of department productivity.

2.1 Paper Output

We all know that the more papers output, the more productive you are. This metric can therefore also scale to a department level. This is the total count of papers for which members of the department have published during the time between black Fridays (our unit of time measurement).

2.2 Quality of Publication

For each publication received by a member of the department in the time unit given, a quality factor will be assigned based upon the number of citations it receives, the importance of the publication location (ie. papers published on my fridge are worth less than SIGCHI which are worth much less than SIGBOVIK), and the number of people who mention the publication to the author without prompting.

2.3 Number of Talks Given

Assuming most talks do not fit in the Bovik quadrant[1] a talk should be considered to increase the productivity of the department.

2.4 Erdos Numbers

Of course the ultimate goal of the department should be to decrease both the variance and average of the Erdos numbers held by faculty and grad students. Since Erdos numbers of infinity when combined with countable integers cannot possibly contribute in a meaningful way to this statistic we propose reducing infinity to 10000 for the purposes of this paper.

2.5 Money

Many department chairpersons, unable to secure funding for an entire department all by themselves, enlist faculty to do this for them. It is therefore important for department members to get money, and should be considered part of the productivity of a department.

2.6 Number of Offices Occupied

This is self evident. I do however divide the offices into those that hold a female and those that do not for future model creation.

2.7 Collaboration Quotient

The number of collaborations between members of the department can help impact several of the previously stated metrics, and so they are given bonus points here.

2.8 Requisite Model Equations

The following equation is put forth as a model for the productivity quotient of the department.

$P = Quality * Papers + \alpha Talks + \beta Erdos + \gamma Money + \delta Office_m + \gamma Office_w + \zeta CQ$

As shown in [2] the above equation is a viable measure of department productivity.

2.9 Related Work

None, since no one was willing to make the politically incorrect (but scientifically correct) argument I am about to.

3. SIMULATION

A simulation was built of the aforementioned variables to determine that as we increased variables in the equation the productivity also increased. We also needed to fine tune the greek constants for the department at CMU. It is recommended that to generalize this to any university department that those parameters be fine tuned based on the culture of your department and the particular bias of its members. For example, if your department contains particular PL professors who feel their research is more important than systems, then those conferences should be weighted more heavily in the quality of publication matrix.

3.1 Live Verification

We then ran the simulation on the department by observing that members of the CS department behaved in predicted manner. Results of this have been published in psychological studies such as [3] and [4].

4. IMPORTANT GRAPHS

If we consider only the original equations for productivity and the aforementioned indicators the graph appears to be relatively straightforward as shown in Figure 1.

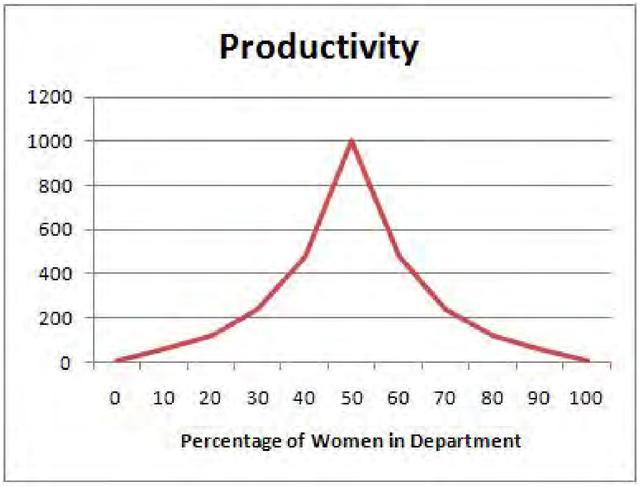


Figure 1: Figure 1:Standard Gender Ratio Curve

Our simulations on humanity have however indicated anomalies in this model, and the premises of this paper is based upon a closer examination of those anomalies.

4.1 Social Relationships

There is a particular observed behavior in the members of a department who make up the majority gender. In computer science departments that happens to be males (unless it is a all women university) and so as such we will refer to the majority gender as male from this point on.

It has been observed that for any woman in the computer science department there is some number of males inhabiting her doorway or office trying to date her. This number is dictated by the actual space in the office and can be set to a constant value of m in the model. Since they are trying to date her and have therefore stopped doing research there is an addition variable to be added to the model equation which represents the decrease in productivity of these men. Since the graphs are symmetric (as the opposite effect occurs when women are in the majority) we will look at only the ratios up to an equal gender division in Figure 2.

Notice in Figure 2 that as the ratio increases to halfway, the productivity of the department actually decreases since the increase from diversity finally overtakes the decrease from either trying to obtain a relationship, or actually trying to maintain one.

5. ANALYSIS AND RECOMMENDATIONS

This is ongoing research and as such many more results are expected to be presented in the actual presentation. The impact of hallway work ethic for the enticement of future dating, and occasional ECE student dating (of either gender) as well as the confounds of distance relationships will be explored as the data emerges.

It is recommended that departments spend careful time fine

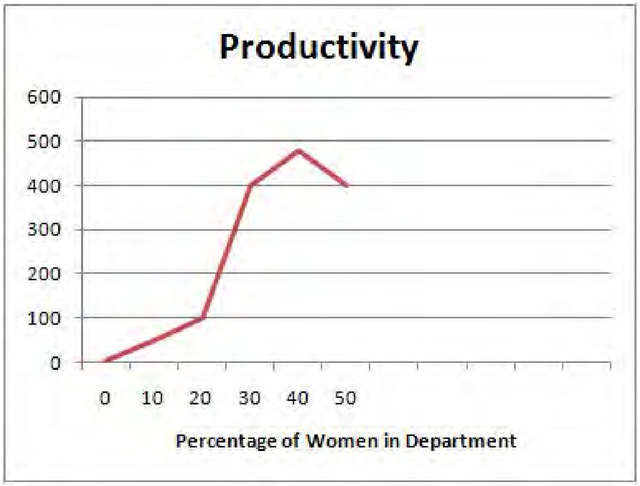


Figure 2: Figure 2:Gender Ratio Curve With Desire to Date

tuning the parameters of their productivity quotient in order to create accurate models of behavior and then use that model to target a particular population of admitted students to attend the university. Since it is unethical to bias the admission rates due to gender, it may also be advisable to create openings in particular departments with an eye towards the historical contribution they make to the gender ratio.

Overall, smaller offices (for less dating-waiting behavior) or locks on female office doors may be advisable. Also, when asked for advice about how to attract a date in advisor meetings, or during social events containing faculty current students should be encouraged to demonstrate their worthiness as a potential significant other through research and other methods that would increase the department productivity quotient.

6. REFERENCES

1 Vaniea,Kami 2010 ”How to Successfully Prevent the Flow of Information in Research Presentations”, in Proceedings SIGBOVIK 2010, Pittsburgh PA

2 Authors, Other 2015 ”A Retrospective Analysis of the Equation for Department Productivity”, Bovik Press, Pittsburgh PA

3 Bovik, Philomenia, ”Verification of Model Equations on Biological Entities Existing in a Computing Environment”, Baker Hall Symposium for the Humanities, 2009

4 Crystal, Billie, ”Understanding Technical Department Productivity in the Wild:How equations can sometimes be scary but useful”, David Klar Presents SCIENCE! Consortium, 2008

Track 1
Productivity, We Say Ironically

(continued on next page)

Track 1
Productivity, We Say Ironically (continued)

Functional Perl: Programming with Recursion Schemes in Python

Robert J. Simmons   Nels E. Beckman
Carnegie Mellon University
{rjsimmon,nbeckman}@cs.cmu.edu

Dr. Tom Murphy VII, Ph.D.
The tom7.org foundation
{tom7}@tom7.org

Abstract

Algorithms that are fundamentally expressions of structural induction over a polynomial data type are famously awkward to implement in object-oriented programming languages, leading to three-day-bucket hacks like the “Visitor Pattern.” We show that the exception-handling mechanism present in most object-oriented languages already suffices.

Conventional wisdom dictates that exceptions are in essence about the non-local propagation of errors. Upon close scrutiny, we find the contrary: exceptions are fundamentally about unchecked depth-1 structural pattern matching.

We give examples of this programming idiom in Python, Perl, and Java. JavaScript requires a language extension, which we build and deploy. We then show how two styles by which this style may be re-integrated into programming languages in the ML family. We conclude by suggesting design changes that would facilitate the use of this idiom in other languages.

Categories and Subject Descriptors D.2.3 [*Coding Tools and Techniques*]: Standards

General Terms Algorithms, Design, Languages, Standardization

Keywords pattern matching, recursion schemes, structural induction, python, perl, java, javascript, types

1. Introduction

Pattern matching is a convenient way of representing polynomial datatypes, which can be informally thought of as a generalization of tree-like data. An extremely simple example is an abstract syntax for a language with integers and addition:

```
datatype tm =
  Int of int
| Plus of tm * tm
```

The big-step evaluation semantics for this language is quite straightforward:

$$\frac{\mathit{tm}_1 \Downarrow x_1 \quad \mathit{tm}_2 \Downarrow x_2 \quad x_1 + x_2 = x_3}{\mathit{tm}_1 + \mathit{tm}_2 \Downarrow x_3} \qquad \frac{}{\mathsf{int}(x) \Downarrow x}$$

Using the pattern matching syntax present in the ML family of languages, the implementation of this big-step evaluation semantics is similarly straightforward:

```
fun eval tm =
  case tm of
    Int x => x
  | Plus (tm1, tm2) => eval tm1 + eval tm2
```

We can also similarly implement a small step semantics for this language as defined by the following rules:

$$\frac{}{\mathsf{int}(x)\ \mathsf{value}}$$

$$\frac{\mathit{tm}_1 \mapsto \mathit{tm}_1'}{\mathit{tm}_1 + \mathit{tm}_2 \mapsto \mathit{tm}_1' + \mathit{tm}_2} \qquad \frac{\mathit{tm}_1\ \mathsf{value} \quad \mathit{tm}_2 \mapsto \mathit{tm}_2'}{\mathit{tm}_1 + \mathit{tm}_2' \mapsto \mathit{tm}_1 + \mathit{tm}_2'}$$

$$\frac{x_1 + x_2 = x_3}{\mathsf{int}(x_1) + \mathsf{int}(x_2) \mapsto \mathsf{int}(x_3)}$$

The ML implementation is a straightforward implementation of these inference rules, with the exception of the need to define an additional `cast` function that we call to get the value of the enclosed integers once we know the terms to be a values.[1]

```
exception Stuck

fun value tm =
  case tm of
    Int x => true
  | _ => false

fun cast tm =
  case tm of
    Int x => x
  | _ => raise Stuck

fun step tm =
  case tm of
    Int x => raise Stuck
  | Plus(tm1, tm2) =>
    if value tm1
    then if value tm2
         then Int(cast tm1 + cast tm2)
         else Plus(tm, step tm2)
    else Plus(step tm1, tm2)
```

[1] In a language without static typing and with distinct varieties of values, these casts can, of course, fail if we get the *wrong variety* of value.

```
interface tm
  { public <T> T accept(Visitor<T> visitor); }
class Int implements tm {
  int x;
  public Int(int x) { this.x = x; }
  public <T> T accept(Visitor<T> v)
    { return v.visit(this); }
}
class Plus implements tm {
  tm e1;
  tm e2;
  public Plus(tm e1, tm e2) {
    this.e1 = e1;
    this.e2 = e2;
  }
  public <T> T accept(Visitor<T> v)
    { return v.visit(this); }
}

interface Visitor<T> {
  public T visit(Plus p);
  public T visit(Int i);
}

public class VisitorExample {
  static int eval(tm tm) {
    return tm.accept(new Visitor<Integer>(){
      public Integer visit(Int i) { return i.x; }
      public Integer visit(Plus p) {
        return p.e1.accept(this)
            + p.e2.accept(this);
      }
    });
  }

  public static void main(String[] args) {
    tm ex_1 =
      new Plus(new Plus(new Plus(new Int(1),
                                 new Int(2)),
                        new Int(3)), new Int(4));
    System.out.println
      ("Evaluating: Plus(Plus(Plus(1,2),3),4) - "
       + eval(ex_1));
  }
}
```

Figure 1. Case study implemented with the Visitor Pattern in Java.

While there is a somewhat more pleasant way to implement the Step function using nested pattern matching, we will only consider the "shallow" pattern matching we see here in this paper.

1.1 Object-Oriented languages and the Visitor Pattern

While ML-family languages suggest describing different types of syntax as different branches of a subtype, most object oriented languages suggest describing different types of syntax as different objects that either implement the same interface or extend the same subclass. The traditionally understood way of performing elegant recursion over these structures is known as the Visitor Pattern; the implementation of big-step evaluation using the Visitor Pattern can be seen in Figure 1.

The Visitor Pattern, whatever its merits may be, is certainly quite different in character from the depth-1 pattern matching used in our ML example. However, because the *exception handling mechanism*, which Java also incorporates, is fundamentally about unchecked depth-1 structural pattern matching, we can seek to use exception handling to provide a similar sort of depth-1 structural pattern matching to the Java language and other object-oriented programming languages. The rest of the paper explores this possibility.

```
class tm:
   pass
class Int(tm):
  def __init__(self, x):
     self.x = x
class Plus(tm):
  def __init__(self, e1, e2):
    self.e1 = e1
    self.e2 = e2

def eval(tm):
  try: raise tm
  except Int: return tm.x
  except Plus: return eval(tm.e1) + eval(tm.e2)

def value(tm):
  try: raise tm
  except Int: return True
  except: return False

def step(tm):
  try: raise tm
  except Plus:
    if(not value(tm.e1)):
      return Plus(step(tm.e1), tm.e2)
    elif(not value(tm.e2)):
      return Plus(tm.e1, step(tm.e2))
    else: return Int(tm.e1.x + tm.e2.x)

ex_1 = Plus(Plus(Plus(Int(1),Int(2)),
                Int(3)),Int(4))
print ("Evaluating:  Plus(Plus(Plus(1,2),3),4) - "
       + str(eval(ex_1)))
print ("Stepping x3: Plus(Plus(Plus(1,2),3),4) - "
       + str(step(step(step(ex_1))).x))
```

Figure 2. Python pattern matching case study.

2. Discovery and implementation in Python

Imagine we have the abstract syntax of our example encoded in Python in the standard way, as a class hierarchy where subclasses `Int` and `Plus` extend the basic abstract syntax class `tm`.

```
class tm:
  pass
class Int(tm):
  def __init__(self, x)
  self.x = x
class Plus(tm):
  def __init__(self, e1, e2)
  self.e1 = e1
  self.e2 = e2
```

The big-step evaluation semantics of this language could be written by explicitly using `isinstance` tests, but the repeated use of `elif isinstance(tm, ...)` is bulky and unsatisfying, which only becomes more true when the code has many, many branches:

```
def eval(tm):
  if isinstance(tm, Int):
    return tm.x
  elif isinstance(tm, Plus):
    return eval(tm.e1) + eval(tm.e2)
  else: raise RuntimeError
```

Note the final line of the above code, which raises a runtime error if the evaluated term is neither an `Int` nor a `Plus`. This is important in case we extend the abstract syntax tree with more branches; we want something equivalent to the "nonexhaustive match exception" in an ML-family language.

2.1 Utilizing the exception handling mechanism

The first key observation for introducing our idiom is the fact that every single object in Python can be treated as an exception and thrown; the language designers suggest that exceptions derive from the `Exception` class, but there is no enforcement of this mechanism. The second key observation is that the branches of an exception handling call can check the object's class tag when deciding which branch of the exception handling to consider. The motivation is to allow, say, a divide-by-zero error or a user-input error to be both caught at the same place but handled in different ways; we can use it, however, to do exactly the `isinstance` test from before in a more concise, pleasing notation:

```
def eval(tm):
  try: raise tm
  except Int: return tm.x
  except Plus: return eval(tm.e1) + eval(tm.e2)
```

Furthermore, this form of evaluation introduces a natural notion of a nonexhaustive match exception: if we add new kinds of terms (say, a multiplication) and don't handle them in the `eval` function, the *very abstract syntax tree node* that we did not handle will be thrown as an exception: an error that potentially contains much more helpful information than an ML `Match` exception!

2.2 Adaptation in Java

The relatively-more-strongly-typed language Java also allows for the expression of our idiom by simply allowing every term to extend the class `RuntimeException`. Technically, all that is necessary for the use of our idiom is `Exception`, but by the use of `RuntimeException` we avoid dealing with compiler warnings caused by lack of exhaustiveness checking (similar to the use of the MLton SML compiler's `-disable-ann nonexhaustiveMatch` option).

Our case study can be seen implemented in Java in Figure 3. It is particularly critical that, in Java, the exception handling mechanism handles appropriately casting the term `tm` to an integer `Int i` or an addition `Plus p`. The typed nature of Java makes the repeated `instanceof` tests much more painful than they were in Python, because we must not only do repeated if-then-else statements but also downcast terms in branches where we *statically know that the cast is safe already* as a result of the `instanceof` test. The exception-handling idiom is free of this particular annoyance.

3. Implementation via language extension

Some languages do not have the essential aspect required for case analysis to be implemented in the exception-handling mechanism. We describe an extensive effort to implement this functionality in JavaScript, as well as commenting on a language where that extensive effort had already been done: Perl.

```
class tm extends RuntimeException {}
final class Int extends tm {
  final Integer x;
  Int(Integer x) { this.x = x; }
}
final class Plus extends tm {
  final tm e1;
  final tm e2;
  Plus(tm e1, tm e2) {
    this.e1 = e1;
    this.e2 = e2;
  }
}

public class PatternMatch {

  static Integer eval(tm tm) {
    try{throw tm;}
    catch(Int i) { return i.x; }
    catch(Plus p)
      { return eval(p.e1) + eval(p.e2); }
  }

  static Boolean value(tm tm) {
    try{throw tm;}
    catch(Int i) { return true; }
    catch(Plus p) { return false; }
  }

  static tm step(tm tm) {
    try{throw tm;}
    catch(Plus p) {
      if( !value(p.e1) )
        return new Plus(step(p.e1), p.e2);
      else if( !value(p.e2) )
        return new Plus(p.e1, step(p.e2));
      else
        return new Int
          ( ((Int)p.e1).x + ((Int)p.e2).x );
    }
  }

  public static void main(String[] args) {
    tm ex_1 =
      new Plus(new Plus(new Plus(new Int(1),
                                 new Int(2)),
                        new Int(3)), new Int(4));
    System.out.println
      ("Evaluating: Plus(Plus(Plus(1,2),3),4) - "
       + eval(ex_1));
  }
}
```

Figure 3. Java pattern matching example.

3.1 JavaScript

JavaScript is a curly-like language oriented on the concept of JavaScript objects, which are functions [1]. The typical way to implement pattern matching would be to use a series of `instanceof` tests, like in Java or Python. As before, we'd like to improve this code to something like what appears in Figure 4.

Unfortunately this does not work because it is not legal JavaScript code. It doesn't parse or run or anything. A typical blunder: JavaScript denies the nature of exceptions by only allowing them to

```
var Int = function(i) { this.i = i; };
var Plus = function(tm1, tm2) {
    this.tm1 = tm1; this.tm2 = tm2;
};

function big_eval(tm) {
  try {
    throw tm;
  } catch (Int(e)) {
    return e;
  } catch (Plus(e)) {
    return new Int(big_eval(e.tm1).i +
                   big_eval(e.tm2).i);
  } catch (e) {
    throw ’stuck’;
  }
}
```

Figure 4. Desired JavaScript code.

```
function big_eval(tm) {
  try {
    throw tm;
  } catch (e) {
    throw ’stuck’;
  } with (Int(e)) {
    return e;
  } with (Plus(e)) {
    return new Int(big_eval(e.tm1).i +
                   big_eval(e.tm2).i);
  }
}
```

Figure 5. Big-step evaluator, using the language extension.

be used for the non-local propagation of errors, requiring a different language feature to be used for pattern matching (`instanceof`). In many languages we would be at an impasse. Luckily, the authors of JavaScript had the foresight to foresee the need to improve the language; it is possible for us to build language extensions within our programs. We can create a full-featured exception mechanism that we can then use to encode our programming technique.

The extension works by co-opting the rarely-used `with` keyword to create a new iterated, pattern-matching `try` ... `catch` construct. The above big-step evaluation code can then be written as in Figure 5. This is basically the same as in the previous languages, except as a stylistic choice the required default case is written first (it is legal for it to be blank or to re-raise the same exception).

You can’t run this code because it is not legal JavaScript. If you try to run it JavaScript will be like, huh? because this is not what the `with` keyword does. To run the code you must first enhance the `big_eval` function using our language extension:

```
  var tm = new Plus(new Int(5), new Int(2));
  var f = big_eval.enhance();
  alert(f(tm).i);
```

The `enhance` property is added to every function by our language extension. It enhances the semantics of the language to add the iterated `try`...`catch`...`with` construct. Running `enhance` returns a new function that works right.

The extension works by converting the function value to source code, modifying the source code, and then evaluating the source code to get a new function value.

```
Function.prototype.enhance = function() {
  var s = ’’ + this;
  var name;
  if (s.indexOf(’function (’) == 0) {
    s = ’function $(’ + s.substr(’function (’.length);
    name = ’$’;
  } else {
    name = /function *([a-zA-Z0-9_$]+) *\(.*/m.
           exec(s)[1];
  }
  var s = ’(function(){’ + rewrite(s) +
          ’\nreturn ’ + name + ’})()’;
  return eval(s);
};
```

A function value’s source code might be a function declaration or a function expression.[2] We put it in a normalized declaration form, giving the function the name $ if it doesn’t have one. We wrap the function declaration in another function and call it immediately, because functions are the only way[3] to introduce new scopes in JavaScript (here thankfully the designers understood that the essence of functions is scope delineation).

The bulk of the language extension is implemented in the recursive `rewrite` function. The function rewrites a source code string; the source code can represent a declaration or an expression or anything. Using strings to represent data helps us avoid tricky type errors that can arise when there’s a difference between expressions and declarations. The implementation of `rewrite` appears in Figure 7.

The rewrite function parses the source code to find appearances of the `catch` keyword, then takes any `with` blocks that appear after them, and rearranges the code just so. It correctly handles the case that uses of the extension are nested, as well as recursively-enhanced functions. The details of the implementation are subtle, mainly having to do with fiddly bits. An example may help the reader; the big-step evaluation from Figure 5 is translated to the code in Figure 7.

It is worth noting in a smaller font that the language extension does not handle the case that source code appears inside string or regular-expression literals. Another way to put this is that the language extension successfully ferrets out not just *uses* of the new feature but *mentions* of it as well.

3.2 Perl

Because of the pun on “Functional Pearl” we really wanted to implement this in Perl, but none of the co-authors really wanted to learn Perl and we couldn’t find Jason [4].

However, while Perl lacks the instanceof-checking exception capability that is critical to our purposes, there is a Perl module, `Error.pm`, that appears to enrich Perl with the necessary features by a strategy similar to the one we investigated for JavaScript.[4] Unfortunately the name of this module also describes what happens when the authors tried to use it.

4. Re-incorporation in ML

Of course, the beautiful synthesis of exception handling and pattern matching can be ported back into Standard ML. Standard ML

[2] It might also be a native function, in which case we cannot enhance it. This is okay, because native functions probably did not use our language extension.

[3] One can also use the `with` keyword but that might mess up the language extension, like if someone tries to enhance an already-enhanced function, or the enhancer function itself.

[4] http://search.cpan.org/ shlomif/Error-0.17016/lib/Error.pm

```
function next(s) {
  var depth = 0;
  for (var i = 0; i < s.length; i++) {
    if (s[i] == ’{’ || s[i] == ’(’) {
      depth++;
    } else if (s[i] == ’}’ || s[i] == ’)’) {
      depth--;
      if (depth == 0) return { p: s.substr(0, i + 1),
                               s: s.substr(i + 1) };
      if (depth < 0) throw ’parse error’;
    }
  }
  throw ’parse error 2’;
}

function rewrite(s) {
  var out = ’’;
  for(;;) {
    var pos = s.indexOf(’catch’);
    if (pos >= 0) {
      pos += ’catch’.length;
      out += s.substr(0, pos);
      s = s.substr(pos);
      var parens = next(s);
      var body = next(parens.s);
      out += parens.p;
      var def = body.p;
      out += ’{\nif(0);’;
      for (;;) {
        s = /[ \n\t]*([\s\S]*)/m.exec(body.s)[1];
        if (s.indexOf(’with’) == 0) {
          var s = s.substr(’width’.length);
          var pp = next(s);
          var body = next(pp.s);
          var ce = pp.p.substr(1, pp.p.length - 2);
          var ctor = ce.substr(0, ce.indexOf(’(’));
          var e =
              next(ce.substr(ce.indexOf(’(’))).p;
          out += ’ else if (’ + e + ’ instanceof ’ +
              ctor + ’) ’ + rewrite(body.p);
          s = body.s;
        } else {
          break;
        }
      }
      out += ’ else ’ + rewrite(def) + ’}’;
    } else {
      out += s;
      break;
    }
  }
  return out;
}
```

Figure 6. The `rewrite` function, which implements the bulk of the language extension.

includes a single *extensible* datatype, named `exn`, which also happens to be the type of ML exceptions. A typical elegance: ML emphasizes the dual purpose of exceptions as both a mechanism for non-local error propagation and for pattern matching by using the `exception` keyword to extend the `exn` datatype:

```
  exception Int of int
  exception Plus of exn * exn
```

```
(function(){function big_eval(tm) {
    try {
        throw tm;
    } catch (e){
if(0); else if ((e) instanceof Int)  {
        return e;
    } else if ((e) instanceof Plus)  {
        return new Int(big_eval(e.tm1).i +
                       big_eval(e.tm2).i);
    } else  {
        throw "stuck";
    }}}
return big_eval})()
```

Figure 7. The automatically-enhanced version of Figure 5.

```
structure PatternMatch = struct

  exception Int of int
  exception Plus of exn * exn

  fun eval tm =
    (raise tm) handle
      Int x => x
    | Plus (tm1, tm2) => eval tm1 + eval tm2

  fun value tm =
    (raise tm) handle
      Int x => true
    | _ => false

  fun cast tm =
    (raise tm) handle
      Int x => x

  fun step tm =
    (raise tm) handle
      Plus(tm1, tm2) =>
      if value tm1
      then if value tm2
           then Int(cast tm1 + cast tm2)
           else Plus(tm, step tm2)
      else Plus(step tm1, tm2)

  val main =
    let val ex_1 =
      Plus(Plus(Plus(Int 1,Int 2),Int 3),Int 4)
    in
     print
     ("Evaluating:  Plus(Plus(Plus(1,2),3),4) - "
      ^ Int.toString (eval ex_1) ^ "\n\n");
     print
     ("Stepping x3: Plus(Plus(Plus(1,2),3),4) - "
      ^ Int.toString
           (cast(step(step(step(ex_1)))))
      ^ "\n\n")
    end
end
```

Figure 8. Re-implementation into SML.

The SML parser requires that `(raise tm)` be surrounded by parenthesis, but otherwise the code retains its essential character, as shown in Figure 8. Of course, because the extensible datatype exn is itself amenable to standard case-analysis, we could instead define our syntax as branches of the exn as we do in Figure 8 and still do a regular pattern-match on it:

```
fun eval tm =
  case tm of
    Int x => x
  | Plus (tm1, tm2) => eval tm1 + eval tm2
```

There are two problems with this approach, however. First, in the case of an unhandled piece of syntax we get an uninformative Match exception instead of the offending piece of syntax itself. Second, we because SML attempts to do exhaustiveness checking on case analysis, every case analysis on an exn that does not have a catch-all case runs afoul of the static exhaustiveness checking. Getting a lot of non-exhaustive match warnings when compiling doesn't bother some people, but it can be annoying [3].

5. Conclusion and future work

There is much work to be done in exploring the beautiful synthesis of exception-handling and pattern matching. For instance, a serious shortcoming of our exception-handling case analysis is that it does not allow *nested* case analysis, which suggests a new language feature – nested exception handling – for Java/Python family languages.

Similarly, the exception-handling idiom allows Java/Python family languages to do a particularly crazy duck-typing-ish thing. Say I want to just enumerate all the integer subterms of my expression, which is extended from the example in the paper to include Plus, Minus, Times, and Div. As long as I called the left hand subterm e1 and the right-hand subterm e2 across these different terms, I can write this code very concisely in Python:

```
def subterms(tm):
  try: raise tm
  except Int: print "Subterm: " + str(tm.x)
  except (Plus, Minus, Times, Div):
    subterms(tm.e1)
    subterms(tm.e2)

ex_1 = Plus(Times(Div(Int(1),Int(2)),
                Int(3)),Minus(Int(4),Int(9)))
subterms(ex_1)
```

Similarly, in Java this could be done by making all the binary operations extend a tmBinary class that, in turn, extends the tm class. In JavaScript, we can continue our approach of implementing Java- and Python-like extensions via syntactic rewrites and then use them in an idiomatic way to simulate a native pattern matching feature. We might think to ask that such functionality be incorporated into ML family languages, were it not for the fact that *it's already there* by way of one of the wackier SML-NJ-specific extensions of Standard ML, "or-patterns," which apparently aren't any more insane than focusing was already [2].

```
fun subterms tm =
  case tm of
    Int x =>
    print("Subterm: " ^ Int.toString x ^ "\n")
  | (Plus(t1,t2) | Times(t1,t2) | Mult(t1,t2)
      | Div(t1,t2)) => (subterms t1; subterms t2)
```

For a larger code example (the implementation of the static and dynamic semantics of Plotkin's PCF), see our extended technical report [5].

Acknowledgments

Milo Polte was instrumental in the discovery of this idiom by encouraging the first author to learn Python. Neel Krishnaswami was the first person who observed that this discovery was probably ridiculous, and Noam Zeilberger reminded the first author what or-patterns are and how they work.

References

[1] ECMA. ECMAScript language specification. Technical Report ECMA-262, ECMA, December 1999.

[2] N. R. Krishnaswami. Focusing on pattern matching. In *POPL '09: Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 366–378, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-379-2. doi: http://doi.acm.org/10.1145/1480881.1480927.

[3] F. Pfenning and C. Schürmann. System description: Twelf - a meta-logical framework for deductive systems. In H. Ganzinger, editor, *CADE*, volume 1632 of *Lecture Notes in Computer Science*, pages 202–206. Springer, 1999.

[4] J. Reed. Gd and l - systems. *The Perl Journal*, 3(1), 1998.

[5] R. J. Simmons. Functional Perl: Recursion schemes in Python (extended case study). Documented code example(s). ACH-BOVIK-2010-003, Association for Computational Heresy, Pittsburgh, Pennsylvania, 2010.

First Year Student's Thoughts on Reflections on Bovik's Seminal Work: Professional Student's Strategy for Perpetual Funding at CMU

Athula Balachandran, Richard Peng, Wolfgang Richter

School of Computer Science, Carnegie Mellon University, USA
abalacha@cs.cmu.edu, yangp@cs.cmu.edu, wolf@cs.cmu.edu

It's going to work [1] [2]. [1]

References

1. Harry Q. Bovik. Professional student's strategy for perpetual funding at cmu. Technical Report CMU-CS-93-000, Carnegie Mellon University, May 1993.
2. Dean Sutherland and Maverick Woo. Reflections on bovik's seminal work: Professional student's strategy for perpetual funding at cmu. In *A Record of The Proceedings of SIGBOVIK 2009*, page 3, 2009.

[1] hopefully

Feasibility Analysis of Cooking Spaghetti with Meatballs

Ben Blum (bblum@andrew.cmu.edu)

2010.04.01

1 Abstract

In this paper we study the major contributing factors to the possibility of making spaghetti with meatballs for dinner tonight. Several different environmental factors are considered, and overall benefit weights are assigned to each, so that results may be comprehensively analyzed for a final result. We studied the available ingredients in the kitchen, and for those required ones that happen to be missing, further analysis is performed to determine whether or not a substitute can be provided from those available or if a trip to the grocery store must be taken. Furthermore, we measure the amount of available time this afternoon and evening, and determine the amount of which we would need to set aside to perform cooking tasks, assuming optimal parallelization. Halfway through the investigation, however, it was determined that we actually wanted chinese takeout, and so efforts were abandoned.

Cat-a-loging !!11!

Leigh Ann Sudol
Carnegie Mellon University
Pittsburgh, PA

Ciera Jaspan
Carnegie Mellon University
Pittsburgh PA

Abstract

As all good research is just combining previous work to produce new methodologies, so too does this paper follow that tradition. Prior work has demonstrated the usefulness of immediate, insightful representation of events. [1] [2] [3] However, as words alone cannot fully take advantage of the visual cortex we postulate that the addition of images will enhance the impartation of information. Cognitive research shows that presenting familiar images lowers cognitive load, thereby allowing for more deep learning and long term retention of material. Cats are familiar. [4] In this paper we demonstrate (with power) that lolcats can successfully be used as a medium for conference streaming. We first describe the conference at a high level, and then move to individual session cat-a-logging.

1 Related work

3 SIGCSE 2010 Lolcatz: Proof of Concept

The following is proof of concept as the conference SIGCSE 2010 is cat-alogged.

3.1 First Timers Lunch

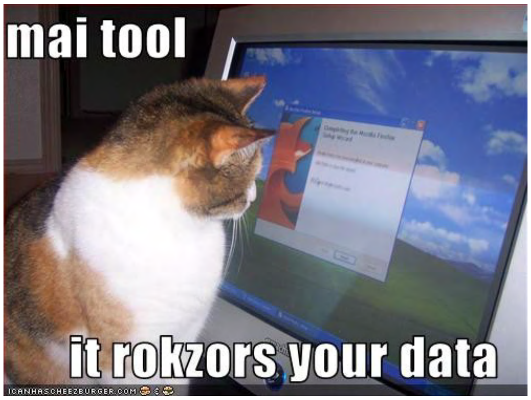
3.2 Reception

3.3 Science Education for the 21st Century

3.4 CS Education Research I

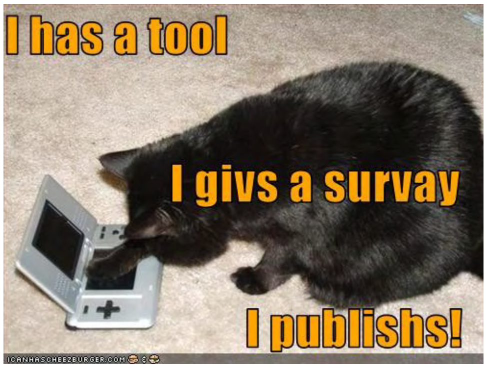
3.5 CS Education Research 2

3.6 Assessment



3.7 It seemed like a good idea at the time

3.8 In Summary of Everything Else



4 Summary

As is obvious we rokzors your brainzors.

5 References

1 www.twitter.com

2 http://www.ireport.com/ (cnn's share your tweets with us page)

3 http://twitter.com/culinartatcmu (CMU's food services)

4 http://icanhascheezburger.com/

A novel approach to managing information overload in syndicated feed readers

Matt Mets

March 21, 2010

Abstract

With the rapid expansion of digital news services, and the trend for users to select content personally, sorting through the available content quickly becomes a major hangup for consumption of content. A user can quickly find themselves overwhelmed, and unable to keep up with the ever-increasing number of news items available.

People suffer from information overload when dealing with their online content, however they are perfectly capable of handling high-information video games.

In this paper, I present a novel method for dealing with this situation. Reaching to the entertainment industry for inspiration, this new method presents news items as entities in a virtual space, and allows the user to quickly deal with them using a simple set of familiar tools, including a laser gun, and star bomb.

“Pimp My Library” Is An Affront To Pimpers Of The World, Everywhere

Dr. Donna M. Malayeri, PhD

Abstract

Inspired by the amazing success of MTV’s “**Pimp** My Ride” [1], the transitive verb form of “pimp” has spread across the Internet. Sadly, the original meaning of the phrase has now become distorted to the point that it is a serious affront to the Americanadian language.

1 Preliminaries

According to my OS X Snow Leopard [2] dictionary, “pimp” has the following meaning:

> verb [intrans.] [often as n.] (**pimping**)
> act as a pimp.

2 Uses

- Pimp My Search [3]
- Pimp MySpace [4]
- Pimp My Snowman [5]
- Pimp My Bride [6]
- Pimp My Library [7, 8]

3 Objectionable Uses

Sadly, popular uses of the term far short of the original meaning of the term. We believe the most egregious example is “Pimp My Library.”

There are two senses in which a Library may be **Pimped**: it may be sold into prostitution, or it may be pimped out. There are also at least two senses of Library here, it may refer to a physical structure (e.g., [7]) or to digital computer software.

It is to this last usage of **Pimp** that we object. A physical Library may be pimped by means of colorful logos, LCD screens, LED head and tail lights, racing car seats, chrome rims, a spinning bowling ball, a popcorn machine, a clothes dryer in the backseat, and so on. But how can such things be added to a digital computer software Library?

3.1 Swiss Army Knife Language Uses

On October 9, 2006, Professor Martin Odersky (hereafter referred to as Scalinventor) (in)famously suggested [8] that Java™ and Scala$^{\text{EPFL}}$ libraries be “pimped” through a mechanism that allows adding new methods to existing classes.[1] Sadly, this is an huge-ass misuse of **Pimp** that has caused considerable confusion in the Americanadian language [9].

Now, Scalinventor, being a German White Dude, cannot be faulted for his lack of knowledge about Rap Culture, an American Invention (or Discovery, depending on one’s point of view). But yet we can totally yell at the blog post for being ignorant of the true meaning of **Pimp**.

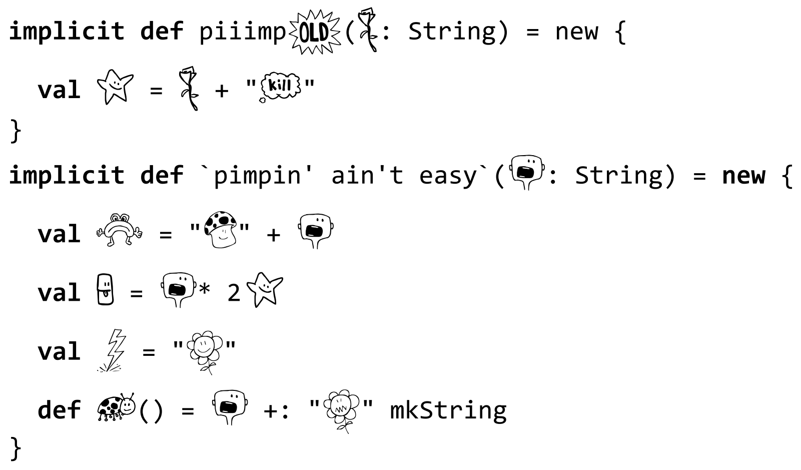
3.1.1 Specifics

The Original Peculiar Post [8] (hereafter referred to as OPP[2]) introduces Scala programmers to a programming pattern whereby new methods can be added to existing classes. Now, there are a number of Doctors of Philosophy (the authors included) that find this to be a useful and way-cool programming language feature [12]. But does it rise to the level of **Pimp**? We think not.

In the OPP, we see the following use of not-really-Pimping:

[1] No joke—look it [8] up, yo.

[2] yeah, you know me



```
implicit def piiimpOLD(⚘: String) = new {
  val ☆ = ⚘ + "kill"
}
implicit def `pimpin' ain't easy`(☹: String) = new {
  val 🐸 = "🍄" + ☹
  val ▯ = ☹* 2☆
  val ⚡ = "✿"
  def 🐞() = ☹ +: "❀" mkString
}
```

Figure 1: True Library **Pimping**, through valid Scala code, and invalid fonts

```
val x = Array(1, 2, 3)
val y = Array(4, 5, 6)
val z = x append y
```

So now, the measly `java.lang.Array` supports an `append` method. Arguably, `java.lang.Array` is so covariant and lame-ass that any addition to it might well be called **Pimping** it out. But the authors, as well as all members of the International Coalition for the Protection of Pimps and Pimping (ICPPP), believe that Xzibit, Q, Alex, Ish, Mad Mike, Big Dane, 2Shae et al, would be appalled at the amount of **Pimping** remaining to be done.

4 Let the Real **Pimping** Begin[3]

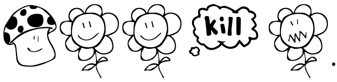
This paper's novel discovery is that the combination of Unicode, Tombats [10], and Scala [11] allow us to truly **Pimp** our Libraries.

For instance, to **Pimp** `java.lang.String`, we may write the code in Fig 1 (font enlarged for those hard of reading).

After those definitions, we may write the following program:

```
"Pimp Me Out".⚡.▯ 🐞 () 🐸
```

Running this program in the Scala interpreter yields



```
🍄✿✿kill❀
```

.[4] [5]

5 Conclusions

Real digital computer software Library **Pimping** must include, at the very minimum, the use of at least one dingbat. Additionally, said dingbats must be badonkadonkical.

References

[1] www.pimpmyride.com/

[2] OS X Snow Leopard, version 10.6.2

[3] www.pimpmysearch.com/

[4] www.pimpmyspace.org/

[5] www.pimpmysnowman.com/

[6] www.youtube.com/watch?v=ti_JR31t9v4

[7] pimp-my-library.blogspot.com/

[8] http://www.artima.com/weblogs/viewpost.jsp?thread=179766

[9] About 112,000 results, says Google.

[10] fonts.tom7.com/fonts98.html

[11] www.scala-lang.org

[12] Malayeri, Donna. *Coding Without Your Crystal Ball: Unanticipated Object-Oriented Reuse.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2009.

[3] Bitches

[4] This code actually works, no joke.

[5] Ok, so you'd have to put in the characters into some unused Unicode code points and load up the font, but that's just implementation details, and future work. Alternatively, you could just change the font of select characters, which is what I did.

Computation in the Complex Time Plane

Matthew Rodriguez
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213

March 20, 2010

Abstract

For a long time, computer scientists have been stuck on certain problems that seemed to be impossible to solve efficiently, or even at all. More frustratingly, some of them were inversifications of trivially easy problems, such as factoring to multiplication. In an attempt to solve these problems, uncomputation was developed. It happened on a fortuitous day when a clever programmer (that is to say, I) realized that by simply reversalating a program, it could be run backwards, and its inputs could be deduced from its outputs in the same time as it would take to run the program forward.

Up until this paper, I had kept uncomputation a secret, but now that I am finished my research into the area, I am ready to unleash it upon the world. It is with great pleasure that I present the exciting new field of uncomputation. As this paper will demonstrate, uncomputation is a revolutionary new field of computation that will enable us to do amazing feats that were previously unfeasible, and with incredible speed.

1 The Basics of Uncomputation

As you know, there exists a trivial algorithm to multiply two n-bit numbers in $O(1)$ time. Just use the * operator and let the deep ECE magic inside your computer do the rest. By simply taking the reverse of this algorithm, we can now unmultiply (or factor if you insist), two numbers in $O(-1)$ time. The minus, of course, comes from the fact that the algorithm is running backwards and concludes before it begins. A simple implementation in C is given below.

```
}
;b * a nruter
{ (b tni ,a tni) ylpitlum tni
```

Those who are not used to writing unprograms may not be able to see it, so I would like to point out that there is a tab at the end of the second line for

unreadibility.
I have found many other uses for uncomputation. One such use is undelete. If you think this simply lets you recover files you have deleted, you are sorely underestimating it. You can undelete anything. Some examples are below.

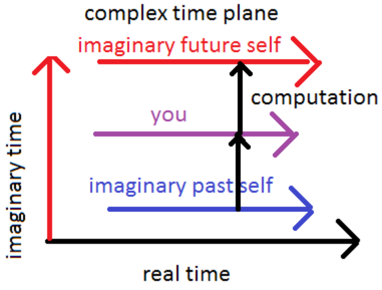
```
undelete meaning_of_life.txt
undelete upcoming_winning_lottery_numbers.txt
undelete grand_unified_theory.tex
undelete lost_works_of_shakespeare.zip
```

Not all of uncomputation is useful. For example, there's never any real desire to unsort a list. Uncomputation is not meant to completely replace computation, only to act as a supplement. Still, undelete alone obsolesces so much of existing computer science and other forms of uncomputation that I will conclude here and proceed to the next topic.

2 Imaginary Computation

Let there be a problem Π with two implementations, Q and $\hat{Q}$, such that Q has running time $O(n)$ and $\hat{Q}$ has running time $O(\sqrt{n})$. $\hat{Q}$ is an optimization of Q. If we simply negate these algorithms, we get $-Q$ and $-\hat{Q}$, with running times $O(-n)$ and $O(-\sqrt{n})$ respectively. Nothing surprising so far. However, if we first negate Q and then optimize it, we get a different algorithm, $\widehat{-Q}$, with running time $O(\sqrt{-n})$, or $O(i\sqrt{n})$. This algorithm is equivalent to $i\hat{Q}$. As the running time indicates, it runs in imaginary time, and we must now generalize our concept of time to include the imaginary time axis.

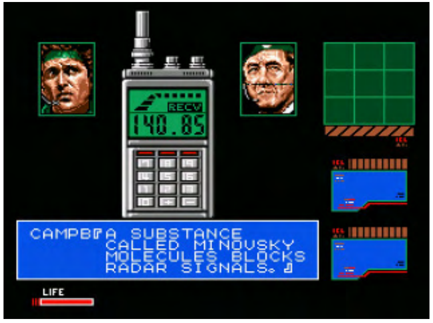
How does this work in the real world where time only flows from past to future? It's simple, really. The real component of $i\sqrt{n}$ is 0, so it concludes instantly from your point of view. What's really going on is that the computation is started by your imaginary past self and runs for a certain amount of imaginary time before concluding with you, located at (now, imaginary now) in the complex time plane. You, in turn, begin the computation and it proceeds toward your imaginary future self. Here is a diagram:



Although this example was a little contrived, it can be generalized to many other traditional computation problems, but not all. On the other hand, there are problems that can only be solved by imaginary computation. In that sense, it is similar to uncomputation. Of course, imaginary uncomputation is useful as well.

3 Complex Computation

To combine the unique capabilities of computation, uncomputation, and imaginary computation, we need the capability to do both at once. Fortunately, we can do this with threads. We can set one thread to run in real time, and another to run in imaginary time, and two more to run backwards and imaginary backwards. As you might expect, complex multithreading is an intimidating prospect. However, unknowingly, we have already been making use of imaginary computation in video games. How else could Solid Plissken talk to Big Snake over the codec without any time passing for the mans around him?



CAMPB「A SUBSTANCE
CALLED MINOVSKY
MOLECULES BLOCKS
RADAR SIGNALS.」

By studying the use of complex computation in video games like Hard Metal Gear, we will be able to apply it to other fields as well. However, that is the extent of my current research.

These are my references. Myself, Harry Q. Bovik for helping me prepare this paper and teaching me how to make very good LaTeX papers, Isaac Newbton for helping me develop a system for notationalizing the symbolology, and Hideous Kojima for making Hard Metal Gear.

Track 1
Perplexity Theory

(continued on next page)

Track 1
Perplexity Theory
(continued)

The Reviewers Must Be Crazy

A Plea For Sanity in Peer Review

C. J. Deere

March 13, 2010

Abstract

While reviewing a conference paper recently, the author noted that one of the options given as an evaluation of the paper was 'borderline', an entirely precise and informative assessment in a sea of unrelated measures such as 'strong reject' and 'weak accept'. This paper proposes a reviewing process that is compatible with current psychiatric best practices.

1 Introduction

We see in Figures 1 and 2 a typical review request form send by a PC member to a grad student who might maybe possibly have some idea of what the paper is about, at least enough to make a plausible-sounding case for rejecting it.

Tellingly, the option of 'borderline paper' comes up all too frequently. Wikipedia, via the fourth edition of the Diagnostic and Statistical Manual of Mental Disorders (DSM-IV) tells us that

> Borderline personality disorder (BPD) is a personality disorder defined in DSM-IV and described as a prolonged disturbance of personality function in a person (generally over the age of eighteen years, although it is also found in adolescents), characterized by depth and variability of moods. The disorder typically involves unusual levels of instability in mood; black and white thinking, or splitting; chaotic and unstable interpersonal relationships, self-image, identity, and behavior; as well as a disturbance in the individual's sense of self. In extreme cases, this disturbance in the sense of self can lead to periods of dissociation.

We note that indeed, ostensibly for "budgetary reasons", very often color printing is disallowed, thus leading to an extremely black and white expression of the author's intent. Most authors are over the age of eighteen. Mood instabilities frequently result from notification of paper rejection or acceptance. Authors constantly write 'we' in place of 'I', even in single-author publications — this is plainly a severe "disturbance in the sense of self".

2 Prognosis

The general ambience of of academic writing and publishing is plainly a psychologically toxic environment, so much so that even the designers of the review form let the fact slip out in their choice of wording. A cry for help, as it were.

We propose that reviewers should be able to take advantage of the most up-to-date classification schemes developed by expert psychologists in order to assess their peers' work more honestly. Our proposed conference review form is given in Figure 3, and we give below representative examples of the sort of writing that is meant to merit each assessment.

3 Examples

In this section we take

3.1 Manic

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Oh man that is so awesome. I'm gonna to write a thousand-page novel about linear types and it's gonna — they're gonna make a movie about it and I'll be so rich and famous and woo linear types!

3.2 Schizophrenic

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Well, that's what the GOVERNMENT wants you to think, anyway. They implanted contexts of linear hypotheses behind my ears. They're controlling my thoughts via algebras of resource possible worlds. The proof-terms tell me to do things. Very bad things.

3.3 Antisocial

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Fuck you.

3.4 Borderline

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Fuck you. Oh, I'm sorry, I didn't mean to say that. It's just that I care about you so much.

Fuck you.

3.5 Obsessive-Compulsive

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once.

3.6 ADHD

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly LET'S RIDE BIKES!

3.7 Depressive

Interpreting linear logic as a lambda calculus yields linear function types, which are required to use their argument exactly once. And then it's all used up. I can never get that resource again. What's the point of going on, then? I loved that linear resource. I'm never going to be able to have it, never, ever again. Sigh.

```
Date: Thu Apr  1 11:12:09 2010
From: mr@cs.cmu.edu
To: bovik@cs.cmu.edu
Subject: BLEEP'10 #72 -- Review Request

Dear [Reviewer],

  below is the review form for the paper

*** SUBMISSION NUMBER: 42
*** TITLE: On Dewubulating Widgets
*** AUTHORS: J. Blow
*** PC MEMBER: Dr. Michael Romanage

  submitted to BLEEP'10.  I need your review back by email by

  Wednesday Mar 31 (i.e., yesterday)

so I can integrate it with my own half-baked notions and submit to
EasyChair on time.

  Thanks,
  Mike

--------------------------------------------------------------
--- For your convenience, this form can be processed by EasyChair
--- automatically. You can fill out this form offline and then
--- upload it to EasyChair. Several review forms can be uploaded
--- simultaneously. You can modify your reviews as many times as
--- you want.
--- When filling out the review form please mind
--- the following rules:
--- (1) Lines starting with --- are comments. EasyChair will
---     ignore them. Do not start lines in your review with ---
---     as they will be ignored. You can add comments to the
---     review form or remove them
--- (2) Lines starting with *** are used by EasyChair. Do not
---     remove or modify these lines.
--------------------------------------------------------------
*** REVIEW FORM ID: 24561::9999
*** SUBMISSION NUMBER: 42
*** TITLE: On Dewubulating Widgets
*** AUTHORS: J. Blow
*** PC MEMBER: Dr. Michael Romanage
--------------------------------------------------------------
*** REVIEW:
---  Please provide a detailed review, including justification for
---  your scores. This review will be sent to the authors unless
---  the PC chairs decide not to do so. This field is required.
```

Figure 1: Typical Review Form, part 1

```
SUMMARY


+ Pluses:


- Minuses:



--------------------------------------------------------------
*** REMARKS FOR THE PROGRAMME COMMITTEE:
---  If you wish to add any remarks for PC members, please write
---  them below. These remarks will only be used during the PC
---  meeting. They will not be sent to the authors. This field is
---  optional.

--------------------------------------------------------------
--- If the review was written by (or with the help from) a
--- reviewer different from the PC member in charge, add
--- information about the reviewer in the form below. Do not
--- modify the lines starting with ***
*** REVIEWER'S FIRST NAME: (write in the next line)

*** REVIEWER'S LAST NAME: (write in the next line)

*** REVIEWER'S EMAIL ADDRESS: (write in the next line)

--------------------------------------------------------------
--- In the evaluations below, uncomment the line with your
--- evaluation or confidence. You can also remove the
--- irrelevant lines
*** OVERALL EVALUATION:

---  3 (strong accept)
---  2 (accept)
---  1 (weak accept)
---  0 (borderline paper)
---  -1 (weak reject)
---  -2 (reject)
---  -3 (strong reject)

*** REVIEWER'S CONFIDENCE:

---  4 (expert)
---  3 (high)
---  2 (medium)
---  1 (low)
---  0 (null)

*** END
--------------------------------------------------------------
```

Figure 2: Typical Review Form, part 2

```
*** REVIEWER'S FIRST NAME: (write in the next line)
*** REVIEWER'S LAST NAME: (write in the next line)
*** REVIEWER'S EMAIL ADDRESS: (write in the next line)
*** REVIEWER'S CONSULTATION FEE: (enclosed)

*** OVERALL EVALUATION:

---  3 (manic)
---  2 (schizophrenic)
---  1 (antisocial)
---  0 (borderline)
---  -1 (obsessive-compulsive)
---  -2 (ADHD)
---  -3 (depressive)

*** REVIEWER'S CONFIDENCE:

---  4 (I can spell words with all the letters after my name)
---  3 (herr doktor)
---  2 (doctor)
---  1 (well my family's pretty crazy on average)
---  0 (read something about psychology in "popular science" once)

*** PRESCRIPTION:

---  4 (electroshock)
---  3 (brand-name medication)
---  2 (generic medication)
---  1 (therapy)
---  0 (placebo)

*** END
--------------------------------------------------------------
```

Figure 3: Revised Review Form

sigbovik 2009 Plenary Program Committee Confidential Paper Reviews

Paper 12: The Reviewers Must Be Crazy

Reviewer 1
Rating: 3 (strong accept)
Confidence: 4/4

The paper (which I clearly didn't read, I just read and review things based on their bylines and titles anyway) makes a strong case that the reviewers must be crazy. This insightful, necessary, and long-overdue conclusion is then supported by several pages of text that I didn't read as well as a figure or two that I might or might not have cursorily examined. The author might or might not have included a conclusion or two as well as a call for future work at the end of the paper. If he did, I have to confess that I really didn't get that far. But hey, that happens sometimes, right?

As a reviewer myself, I can vouch for the author's results and conclusions (by which I mean, his title) since I might or might not be crazy. Therefore his work must be true since I am always right. QED.Of course, I must follow every reviewer ever by giving the same comment: your work is excellent/good/adequate/fair/poor/omgburnitwithfire but you need to cite more of the empirical literature. As this review is now in the public domain, I highly encourage you to cite this and raise my citation count so I can get tenure and stuff. You know, because I really want to be cited by a paper that suggests that all reviewers are crazy. Deans can't read but deans can count, right? Hey actually, that could be a paper someday, just don't steal that idea....or if you do, just cite me and we'll all be good.

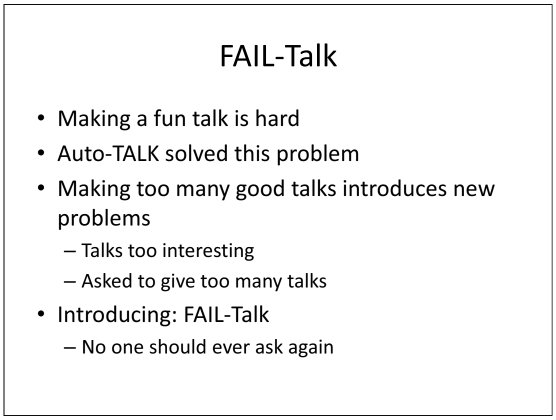
STORNG ACCEPT!!!11!!one one one

FAIL-Talk: Fully Automatic Incomprehensible Language Talks

Alga Rhythm

Abstract

The author previously introduced Auto-TALK, a solution for making boring, text-based talks much more interesting [1]. A potential problem with Auto-TALK is that speakers can become too interesting and students may be requested to give many talks at seminars, taking valuable time away from research. In this work, a “metaphorical inverse” FAIL-Talk is introduced, which solves the *too-many-talks-not-enough-time problem*.



FAIL-Talk

- Making a fun talk is hard
- Auto-TALK solved this problem
- Making too many good talks introduces new problems
  – Talks too interesting
  – Asked to give too many talks
- Introducing: FAIL-Talk
  – No one should ever ask again

Figure 1: Initial slide, pre-transformation

1 Introduction

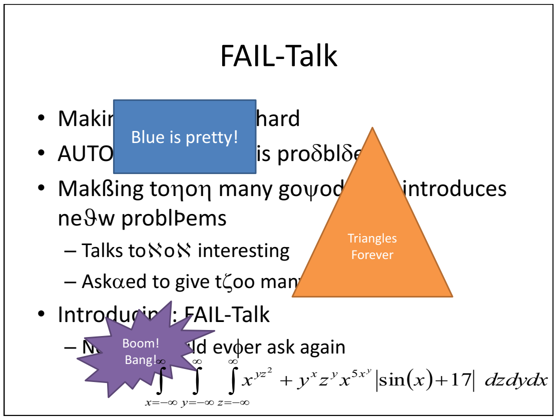
No one (initially) wants to give a bad talk. In prior work, Auto-TALK [1] was introduced a solution to the endemic problem of boring talks. It incorporated pictures of puppies, kittens and YouTube videos to gather and hold audience attention. An unfortunate side effect may be students being *so* interesting that the demand for talk impinges on valuable research time. This is known as the *too-many-talks-not-enough-time problem*: grad students only have so much time, and preparing talks takes time which could otherwise be spent on research.

Our proposal to remedy this travesty is FAIL-Talk. Like Auto-TALK, FAIL-Talk provides a concrete set of steps (an algorithm, if you will) to transform a talk into a new “improved” talk. Unlike Auto-TALK, the goal of a FAIL-Talk transformation is to discourage requests for further talks. Ever.

We will utilize simple strategies to transform talks from basic but comprehensible to completely incomprehensible. Our toolkit is easy, adaptable, and multi-purpose, able to be deployed cross-discipline without much adaptation.

2 FAIL-Talk Transformations

There are three main techniques for making a basic talk horrible: talking too fast, unintelligible slides, and random incorporation of multi-variable integrals.

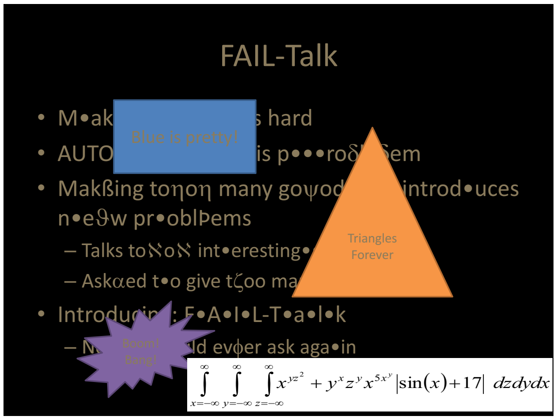


FAIL-Talk

- Makir hard
- AUTO is proδblδe
- Makßing toηoη many goψod introduces neϑw problÞems
  - Talks toℵoℵ interesting
  - Askαed to give tζoo man
- Introducin : FAIL-Talk
  - N ld evϕer ask again

$$\int_{x=-\infty}^{\infty}\int_{y=-\infty}^{\infty}\int_{z=-\infty}^{\infty} x^{yz^2} + y^x z^y x^{5x^y} |\sin(x)+17|\ dzdydx$$

Figure 2: After the addition of random shapes with text, an integral, and text perturbation in another language (here, mostly Greek)).



FAIL-Talk

- M•ak s hard
- AUTO is p•••roδ δem
- Makßing toηoη many goψod introd•uces n•eϑw pr•oblÞems
  - Talks toℵoℵ int•eresting•
  - Askαed t•o give tζoo ma
- Introducin : F•A•I•L-T•a•l•k
  - N ld evϕer ask aga•in

$$\int_{x=-\infty}^{\infty}\int_{y=-\infty}^{\infty}\int_{z=-\infty}^{\infty} x^{yz^2} + y^x z^y x^{5x^y} |\sin(x)+17|\ dzdydx$$

Figure 3: Black background, brown text, and extra bullets for intra-word emphasis devastate any remaining comprehensibility.

2.1 Talking Too Fast

The author asserts that talks become harder to understand in proportion to the speed the presenter speaks. The easiest transformation, then, is simply for the presenter to speak as quickly as he/she can.

2.2 Unintelligible Slides

The less sense the slides make, the less people will enjoy a talk and the less they'll want to repeat an experience. (While the author has no proof of this statement, it appears reasonable.) So rather than black text on a white background, brown text on a black background is recommended. Whenever possible, animations should be used to cover text with completely useless block diagrams. (Note: it is important that the block diagram be completely unrelated to the talk).

If this is insufficient, more drastic measures can be taken. Text can be perturbed with random characters from other languages to ensure slides make absolutely no sense.

It is important to note that stripping the talk of all relevant images/video and inserting bullet points randomly and with high frequency are also highly recommended.

2.3 Incorporating Multi-Variable Integrals

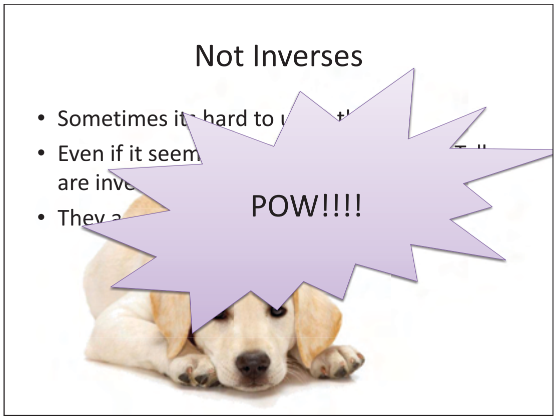
To complete the transformation, some $\epsilon-$percentage of slides should have multi-variable integrals added, such as

$$\int_{x=-\infty}^{\infty}\int_{y=-\infty}^{\infty}\int_{z=-\infty}^{\infty} x^{yz^2} + y^x z^y x^{5x^y} |\sin(x)+17|\, dzdydx$$

The harder the integrals are to solve, the better; the integrals should have *no relevance whatsoever* to the talk. (In the event that integrals seem too drastic, other mathematical equations can be inserted, with lesser effect).

2.4 Commutativity of transforms

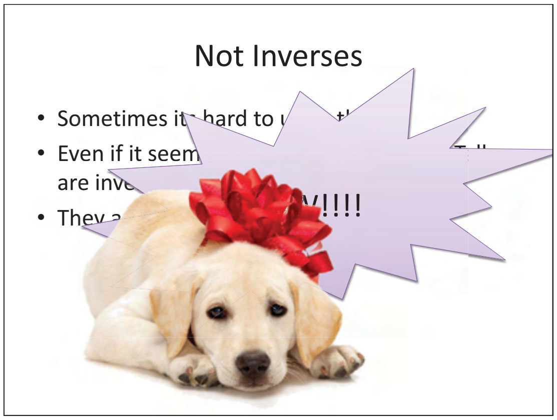
While the transforms provided may not completely commute (example: the order which two



Not Inverses

- Sometimes it hard to u
- Even if it seem are inve
- They a

Figure 4: FAIL-Talk applied to Auto-TALK. Dog is now under the "Pow" shape. Again, the original text slide is not restored.



Not Inverses

- Sometimes it hard to u
- Even if it seem are inve
- They a

Figure 5: Auto-TALK applied to FAIL-Talk. Original is not restored, and dog appears in front of "Pow" shape.

basic powerpoint figures appear may influence which lies on top of the other), the author argues that the *net effect* of the transforms does indeed commute. For example, block diagrams can be added, then text can be perturbed by a foreign language, followed by the addition of integrals. If at this point the author were to change the text to brown and the background to black, and finally insert excess bullets, the progression may resemble Figures 1, 2, and 3. Clearly, the net effect is an incomprehensible slide, regardless of transformation ordering.

3 Experimental Evaluation

At this point, the author has not yet field tested FAIL-Talk, as she wishes to avoid burning bridges. However, mocks are included as Figures 1-3, clearly demonstrating the devolution of a slide from one that makes sense (Figure 1) to one which is horrifying (Figure 3).

4 Metaphorical Inverses

The author would like to assert that FAIL-Talk and Auto-TALK are not true inverses in the mathematical sense. For a set of slides $p \in P$, where P is the set of all possible PowerPoint talks, FAIL-Talk(Auto-TALK(p)) may yield a talk, with an embedded YouTube video which is covered by a block diagram containing an integral. An example of FAIL-Talk(Auto-TALK(p)) is shown in Figure 4, where puppies are applied before a large "Pow!!!" shape. Likewise, Auto-TALK(FAIL-Talk(p)) may make a bad talk somewhat more bearable (via puppies) but does not restore comprehensibility or even the original talk.[1] An example of Auto-TALK(FAIL-Talk(p)) is shown in Figure 5, which simply inverts the ordering of the transformations shown in Figure 4.[2]

[1] The author recommends keeping a backup copy if there is a possibility of a need for the original talk.

[2] Note that neither Figures 4 or 5 restores the original talk, which was a pure text slide.

5 Conclusion

This work presents FAIL-Talk, a fully automated scheme to make incomprehensible talks. While the automation is not yet complete, the author has demonstrated the awe-inspiring power of this scheme to completely destroy the substance of a talk while adding only nonsense via a few simple powerpoint mocks. While this work can be thought of as a solution to the problems AutoTALK created, it may itself yield side-effects: presenters should be prepared to never be asked to give a talk again, even if they eventually want to.

References

[1] Alga Rhythm. AutoTALK: An Automatic Presentation Graphical Toolkit. In *SIGBOVIK*, 2009.

An Analysis and Redesign of the Evacuation Instructions in an Academic Office Building

I. B. Gone and U. Cower

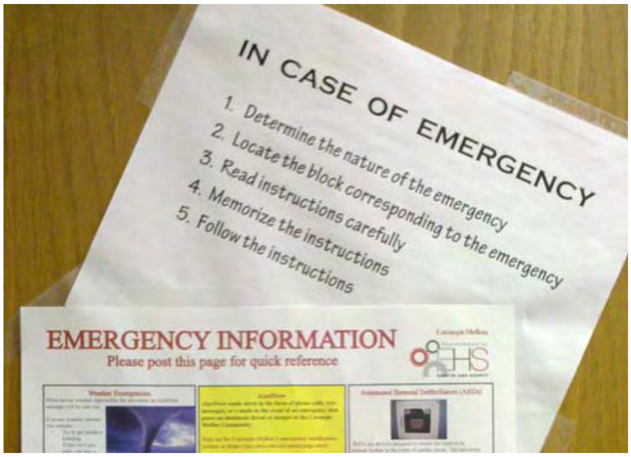
Abstract

An evacuation process is provided for an academic office building. The efficacy of this process is evaluated, and the maturity of the evacuation model is assessed to be EMM Level 1. A scalable improved process description is proposed.

Introduction

Safety is recognized as an essential human need and a prerequisite for productive activity. Safety needs are placed just above physiological needs and below belonging, esteem, and self-actualization by Maslow's hierarchy.

A major university~~, in its infinite wisdom,~~ has provided a "quick reference" guide to the evacuation process for its academic buildings. The evacuation process, as promulgated to building occupants, is exhibited in Figure 1. The manifest complexity of this description has led to postings with supplemental instructions, such as the example of Figure 2.



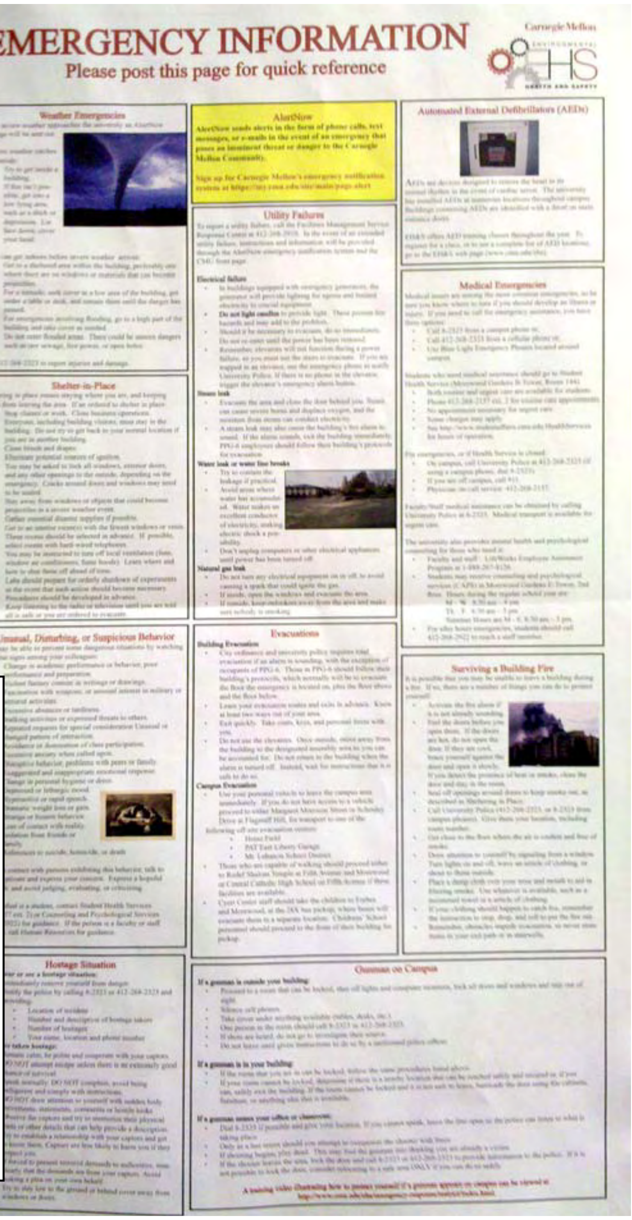
IN CASE OF EMERGENCY

1. Determine the nature of the emergency
2. Locate the block corresponding to the emergency
3. Read instructions carefully
4. Memorize the instructions
5. Follow the instructions

EMERGENCY INFORMATION

Please post this page for quick reference

Figure 2: Supplement to Evacuation Process



EMERGENCY INFORMATION

Please post this page for quick reference

Weather Emergencies

AlertNow

AlertNow sends alerts in the form of phone calls, text messages, or e-mails in the event of an emergency that poses an imminent threat or danger to the Carnegie Mellon Community.

Sign up for Carnegie Mellon's emergency notification system at https://my.cmu.edu/site/main/page.alert

Automated External Defibrillators (AEDs)

Utility Failures

Medical Emergencies

Shelter-in-Place

Unusual, Disturbing, or Suspicious Behavior

Evacuations

Surviving a Building Fire

Hostage Situation

Gunman on Campus

Figure 1: Official Evacuation Process

Despite the distribution of this descriptive quick reference poster, lack of familiarity with the process on the part of building occupants is suggested by considerable anecdotal evidence. After all, the number of people who will actually memorize this stuff is vanishingly small, and who’s going to read the fine print with flames licking at their ankles?. Unfortunate consequences could ensue in the event of an emergency.

A user study was conducted to determine the efficacy of the evacuation process. The results were assessed under the Evacuation Maturity Model, with the result that the process is at EMM Level 1. A revised, scalable description of the evacuation process is proposed.

Efficacy of the Evacuation Process

A number of building occupants were ~~recruited~~ dragooned into this study. They were presented with the current evacuation process, their interpretations were observed, the data was de-identified, and they were then placed in simulated emergency situations (Figures 3, 4, 5).

As can be seen from the illustrations, confusion and consternation were prevalent. The principle of intra-ocular trauma is satisfied by the data, so no further statistical significance testing is required.

Figure 3: Simulated bomb emergency

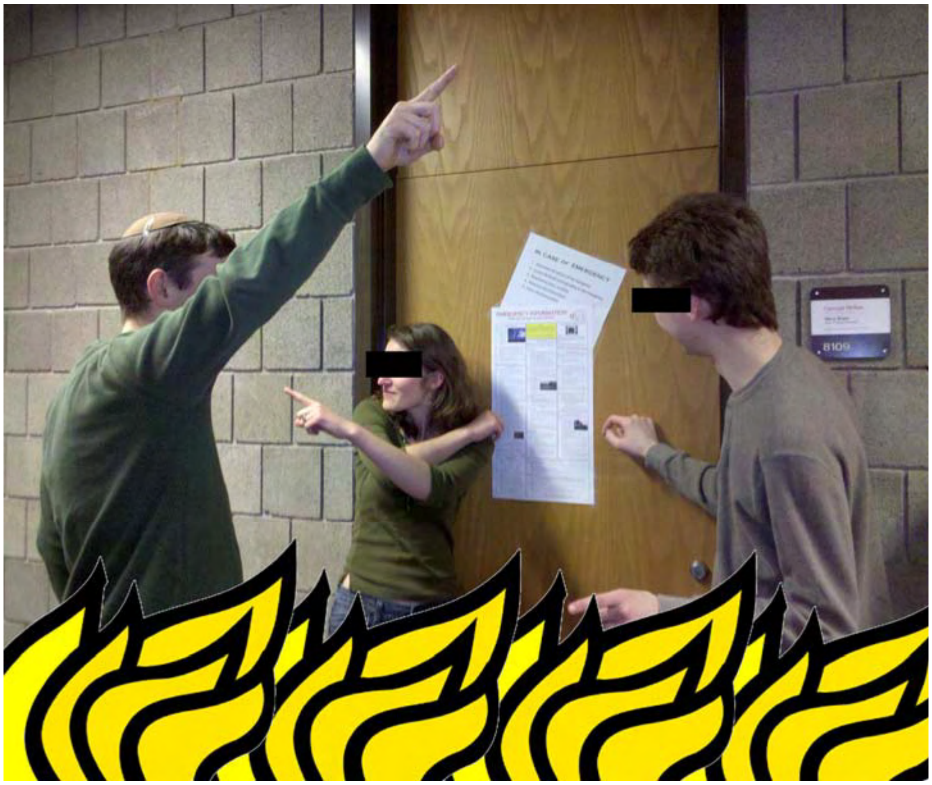


Figure 4: Simulated fire emergency

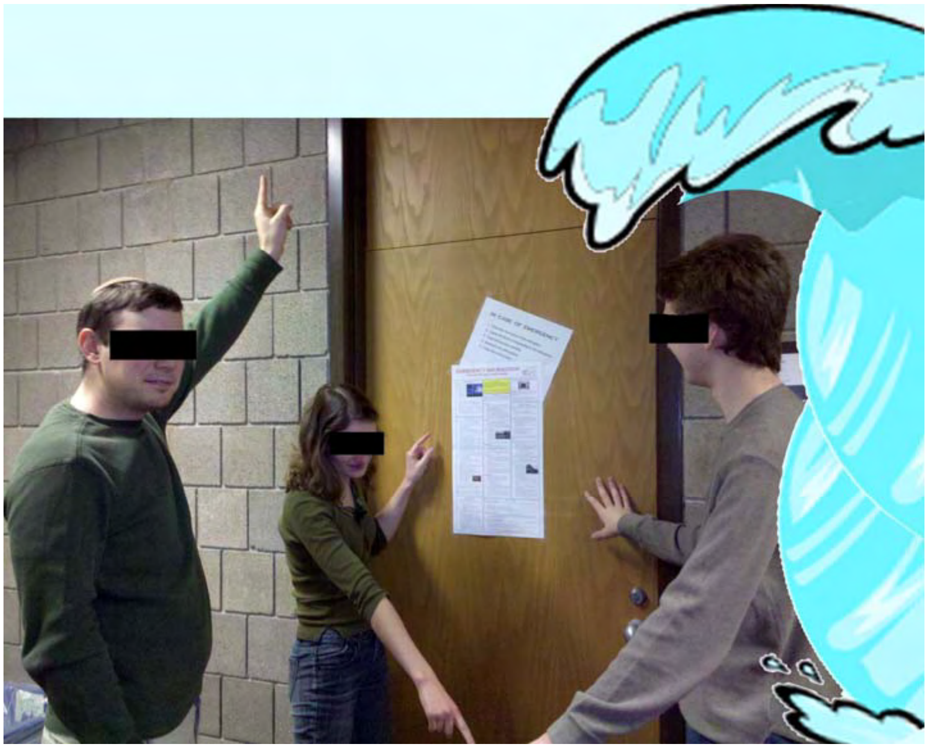


Figure 5: Simulated weather/flood emergency

Evacuation Maturity Model Assessment

The Evacuation Maturity Model (EMM) was developed as a tool for objectively assessing evacuation plans of businesses and institutions. A 5-Level process maturity continuum is recognized, where the uppermost (5th) level is a notional ideal state where processes would be systematically managed by a combination of process optimization and continuous process improvement. The five levels are defined as:

1. Initial (chaotic, ad hoc, individual heroics) - the starting point for use of a new process.
2. Repeatable - documented procedures and checklists are used to guide the process
3. Defined – confirmation as a standard business process is accomplished
4. Quantitatively managed - quantitative goals and objectives are incorporated
5. Optimized - deliberate process optimizations/improvements are included

Lack of understanding of the process on the part of the occupants was revealed by the efficacy study. Study of the process would substantially lengthen the critical path to emergency action. In the event of an emergency, heroics would be required. Explicit objectives are lacking. A maturity level of EMM1 is assigned by the maturity assessment.

Improved Process Description

An improved process description design is suggested by the data. Details of the process, even if required for precision, are elided. Comprehension and interpretation tasks are removed from the critical path in favor of action. An approximate process actually executed is deemed preferable to a precise process ignored.

The improved process description is presented in Figure 6. Dimensions for the document are not provided, as the description is intended to result in evacuation process improvement at all scales from wallet card to billboard.

if	then
Fire	Run
Gun	Hide
Storm	Stay
Flood	Climb
No pulse	Zap
Blood	911

Figure 6: Scalable simple evacuation process description

All my friends' researches are cooler than mine

Chris A. Maphone

March 21, 2010

Abstract

We have done no research and have nothing to say about it. Instead this paper presents solutions a multitude of *exposition problems*: we take it upon ourselves to succinctly and accurately describe the research of our friends and colleagues, to the best of our understanding. We think that previous work (i.e. their own explanations of their work) vastly understates how TOTALLY AWESOME what they are doing is and misses key points about real-world applications of their work for e.g. blowing stuff up.

1 Introduction

This is a survey paper on some of the recent exciting research endeavors of many awesome Carnegie Mellon students and alumni!!!!!!!!

2 Research

Dan Licata works on something called "dependent types". They are so powerful they can do ANYTHING. He is working on building the ultimate type theoretic weapon which uses focussing to create a concentrated beam of power in the form of two "arrows", "Mr. Annihilate" and "Dr. Destroyer" [1]. This weapon is called a TASER (type amplification by stimulated emission of radiation).

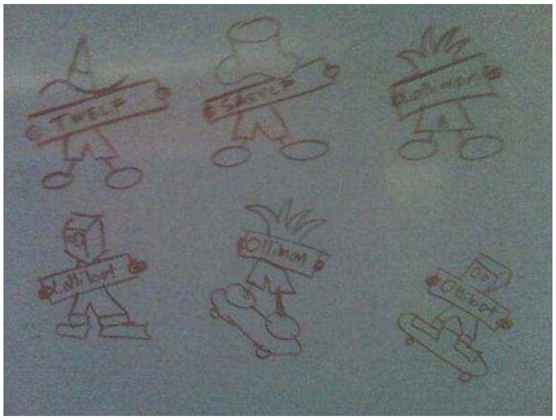
Ross Hatton works on kinetic typography, which is like letters moving around and changing size and stuff. Specifically he applies the kinetic approach to calculus formulas and, using Stokes's theorem, can describe the very motion of the typographic characters in the formula. Then his labmates turn these animations into robots. The robots are made with certain "gates" (like and and not) described by the equations. One of them has VERY SHARP BLADES and has to be kept in a box. Someday, these robots will get very tiny and invade our bodies for medical science.

William Lovas refines types finer and finer until they are sharp enough to stab a bitch [2], and then some more until they are a soft powder. This means you can describe your data so precisely that it requires quantum physics, so like if you have a natural number it's actually a quantum superposition of even and odd and you never know for sure until you observe it with type reconstructivism.

Michael P. Ashley-Rollman (MPA) makes blocks that light up. I think they are children's toys, but I like playing with them too even though I am a mature grown-up person, pretty weird I know. Anyway you click them together and they change colors, and sometimes because of racial conditions the colors spazz out and start flickering a lot. Michael is working on solving the problem of racism.

Luke Zarko is probably a superhero otherwise why would he have that name? He uses his superpowers to make an operating system (ZARKOS) that already knows at compile time what your code is going to do, so you never have to run anything and it's very safe. Unsafe programs like GUN BASH never even get executed in the first place because you can tell from the name that it's very fangorious. This operating system is not only provably correct but also VERY FAST.

Rob Simmons makes logic programs go backwards. It's actually kind of confusing because usually "backwards" means "forwards" so sometimes they say "bottom-up" which I think is related to his prior SIGBOVIK publications on drunken logic [4]. Anyway every time he makes a program go backwards he gets a new elf:



Tom 7 used to make logics about spaceships but now he uses machines to learn things. His thesis is titled "Model Types for Mobile Code" [3] and it has to do with model logic (as in model spaceships, because real ones are too big to fit in a computer, even a distributed one).

3 Conclusion

I JUST HAD TWO ESPRESSOS!!!!!!!!!!!!! The end.

References

[1] Daniel Licata. The strange case of dr. admissibility and mr. derive. Speaking Skills Talk at CMU Student Seminar Series, 2008.

[2] William Lovas and Frank Pfenning. A bidirectional refinement type system for LF. *Electronic Notes in Theoretical Computer Science*, 196:113–128, January 2008.

[3] Tom Murphy, VII. *Model Types for Mobile Code*. PhD thesis, Carnegie Mellon, January 2008. Available as technical report CMU-CS-08-126.

[4] Rob Simmons. A non-judgmental reconstruction of drunken logic. In *Proc. SIGBOVIK 2007*. ACH, April 2007.

How to successfully prevent the flow of information in research presentations

Kami Vaniea

March 21, 2010

Abstract

The main body of research presentation literature is dedicated to the creation of great research talks in which the presenter successfully conveys their great wisdom to an audience of receptive minds. The great art of preventing your audience from figuring out what you actually did is not nearly as well documented or explored. In this work I will explore the different presentation mechanisms that can be used to simultaneously give an excellent talk, thoroughly confuse the audience and leave them with a feeling that you really understand what you are talking about.

1 Introduction

Creating a hard-to-comprehend talk is amazingly similar to creating an enlightening talk. You need to do things like consider your audience, plan out what you are going to say and meticulously create your slides to guide your audience through your material. The point where the two methodologies diverge is in the goals. Unlike a traditional information imparting presentation an information impeding presentation has the dual goals of convincing your audience you know what you are talking about while makings sure they don't understand anything you say.

Some critics may say that creating information impeding talks is a debasement of science and a complete waste of your audience's time. However, they are overlooking the needs of thousands of students who's results were neither interesting nor useful placing them in the not interesting and not useful results quadrant, aka Bovik quadrant, of the quest for fundamental understanding and considerations grid put forth by Donald Stokes [2]. Unlike the praised work in Pasteur's quadrant the work in Bovik quadrant is woefully ignored and not given the recognition it deserves.

The purpose of this work is to assist students everywhere in accomplishing two goals with their talks 1) convince the audience that you did something worthwhile 2) make sure the audience never figures out what that important thing was. These are ambitious goals but never fear they are completely accomplishable if you pay close attention to this guide.

In order to impede information flow during your talk you need to accomplish the following sub tasks:

- Encourage interior audience discussion
- Belittle the audience
- Mislead the importance and relevance of your data
- Use as much time as possible explaining unimportant information

In this work I will discuss each of the major components of a talk and look at different ways to accomplish these tasks in each component. Curious how your going to accomplish all this in a single talk? Well, read on.

2 Motivation

Cloud computing! Cancer! Google! Health care! Kittens! Global warming! Remember, whatever your research is it can be related to the hot news item of the week. All motivation really is tying your research to your favorite well funded buzzword. Remember that good motivating examples accomplish three things 1) convince audience your work is important 2) take up as much time as possible and 3) make it unclear just how many of the world's problems you are actually going to solve in this talk.

When properly obfuscating a talk its important to pick your motivating example carefully. You want an example that lots of people care about so that they will leave your talk feeling like you solved something important even if they didn't understand a word of what you did. But you also need a motivation that not too many people understand and is hard to ask detailed questions or point out inconvenient facts like your motivation and your work have nothing in common.

Cloud computing is a wonderful example of a good motivation topic. Everyone knows that cloud computing important, complicated and either doomed to failure or the next bit thing. Cloud computing needs machine learning to ensure optimal usage. It needs security to protect data from the unnamed dangers lurking in the clouds. It needs theory if it is ever going to work. It needs PL to specify how different elements can interoperate safely. Whatever your work is about play it up in relation to your motivating topic. Without your vital, unspecified, work your motivating topic would be crippled and the world would descend into chaos.

When introducing a motivating example, like cloud computing, be sure to spend plenty of time on it. Remember the more time you spend on your motivating example the less you have to talk about your actual research or even worse, your results. Besides motivating examples are far more fun to talk about and listen to. So spending lots of time on it will actually be enjoyable for both you and your audience.

One of the purposes of a motivational topic in an information imparting presentation is to introduce the problem you are going to be solving in the remainder of the talk. In an information impeding presentation this is an excellent opportunity to mislead your audience by implying that your research solves many more problems than it actually does. For example strongly type checking the Internet would make it more secure. Therefor nearly all security problems associated with cloud computing can be solved by applying a novel type checking method.

3 Problem statement

The problem statement is arguably the single most important part of your talk. The problem statement lets your audience know what you will be talking about. This is bad. You do NOT under any circumstances what the audience to figure out what you are talking about.

If possible, you should really skip this part of your talk. The best solution is to leverage your motivating example and just assume that it doubles as a problem statement. Thereby not skipping the problem statement while still not clearly explaining it.

4 Outlines and organization

Outlines are generally a bad idea if you want to make sure your audience gets the wrong lesson from your talk. But this doesn't mean you can't have an outline slide. Remember the goal here is to give what looks like a good talk while ensuring the audience doesn't understand what you say.

A good outline will let the audience know what is coming, what you are going to tell them about and how you are going to solve your problem all in a neat package. Creating an outline with no meaning that meets these requirements is actually trivial just copy the following onto an outline slide. In your talk just read the slide and provide no extra information.

- Introduction
- Motivation
- Background
- Methodology
- Results
- Conclusion

This simple outline gets you points with the audience for having an outline while giving them no additional data about what is coming.

5 Graphs and diagrams

Graphs follow a similar rule to statistics. Remember there are lies, dam lies and statistics. Similarly graphs give you the ability to give your audience the wrong impression while not lying.

To begin with there are the tried and true graph and figure obfuscation techniques. Leave out labels. Leave out the scales. Use three dimensions. Use as many of Excel's colors, textures, and eye candy as possible thereby impressing your audience with the visual complexity. A good example of this sort of graph can be seen in Figure 1. This is actually a graph of all the book sales in the US from 1940 to 1989 but that is impressively hard to tell from the graph. Use a Chinese coloration scheme where red is good and white is bad which completely confuses American audiences. Put in as much data as possible even if it could be left out or isn't relevant to your point.

Math is already a horribly confusing topic why would any sane person put more of it into a talk on purpose? Be wary of adding math and especially wary of adding statistics which can be questioned. It is far better to add a graph similar to Figure 2 which shows the clear superiority of your technology. If the difference is clear enough to your audience you can safely skip over any issues of statistical significance or error bar reporting.

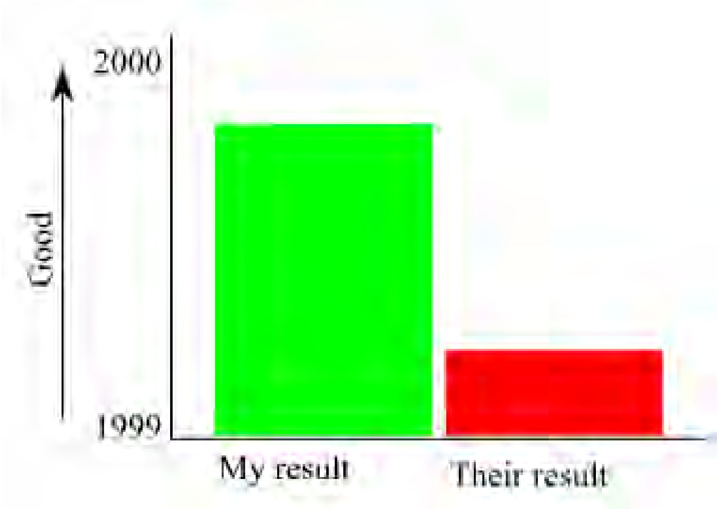


Figure 2: Example of a graph showing the clear difference between your good result and the opposing technologies poor result. Remember clear results speak for themselves and if they don't you just need to muddle with the graph until they do.

If simple graph alterations fail to adequately describe the importance of your results you may have to resort to more complicated tricks. Thankfully many books have been written on how to build good graphs which also conveniently show lots of examples of bad graphs. Tufe's famous book on The Visual Display of Quantitative Information [3] is an excellent resource.

6 Audience Questions

Annoyingly, audience members are typically allowed to ask questions during and after your talk. Life would be much easier if this practice could be avoided but unfortunately it is ”traditional” and cannot be prevented.

Much like an information imparting talk it is important to anticipate audience questions in advanced and have prepared answers. Knowing your answers in advance allows you to answer questions in a way that doesn't answer the question while making your words drip with confidence and certainty. Much like a lion tamer you

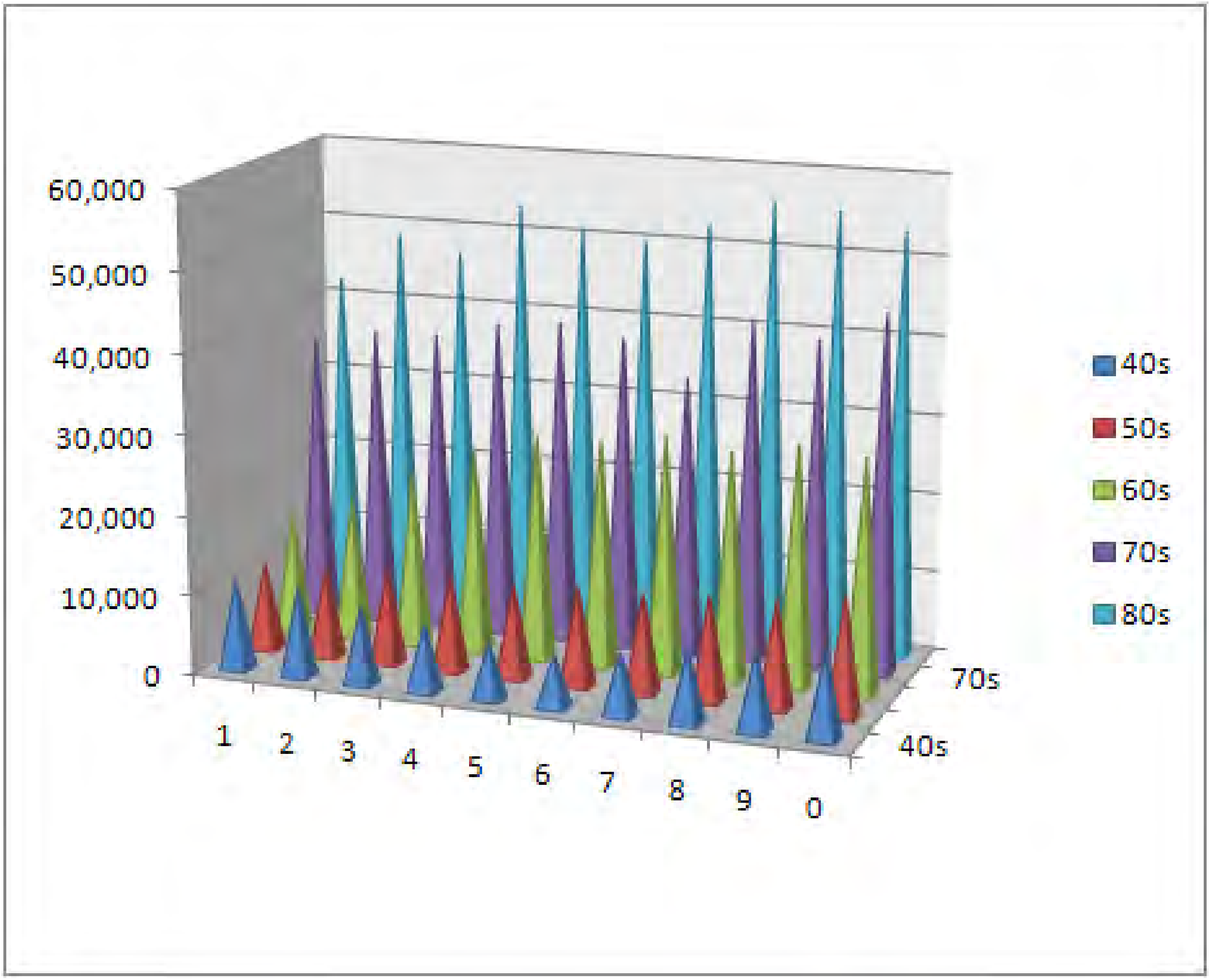


Figure 1: Example of a visually confusing graph created using simple MS Excel default tools. Note the lack of any explanation as to why this graph is here or what it is about.

should show no fear before your audience and subdue them through your aggressive eye contact and confidence.

If an audience member asks a question which reveals a flaw in your work or in some way doesn't show you in the best light, answer a different question. Beauty is in the eye of the beholder and question interpretation is in the ear of the presenter. If you need some examples of how to do this effectively just watch any presidential debate. The art of question avoidance and deflection is what politicians are very good at. If you need to cite facts in your answer pick little known works and refer to the author by first name. (For example: Joe)

Get upset when others challenge you. An attack against your research results is really a personal insult to you. After all you poured your heart and soul into this research for the last n years who are they to condemn it after less than an hour? Also, a heated discussion between you and the audience member will distract the rest of the audience from your talk and reduce the time available for other people to ask questions.

If an audience member asks you a question about a result which isn't yours you should defend the other author and answer as if it was your research. This shows deference to the other author's hard work and derails the talk for several minutes until the audience discovers this is cited work not yours.

7 Belittling an audience

You are smart. Well that is obvious since you are reading this paper. Its important to let your audience know how smart you are. Since you are reading this paper your results likely fall in the Bovik quadrant [2] and are unlikely to impress upon your audience how smart you are. Therefore its vital to make this fact clear in other ways.

To make your audience understand how smart you are it is vital to make them understand just how much they don't know. After all getting the results you have was hard. Its important for your audience to appreciate just how much work you had to do and how mentally taxing and frustrating it was. How do you do this? Simple, show them the details. Don't hold anything back. If there is something confusing in your domain don't be afraid to say it. If you have tons of data points show them all. And most importantly do not simplify or explain yourself. You are the Alan Turing in the room and they are supposed to keep up. And if they can't keep up that is an indication of their ineptitude not yours.

Frequently use terms like ”obviously” and ”clearly” to underline how little your audience actually knows. Obviously your research cures all the worlds problems. Your results clearly show this and all the equations and math backing it up are intrinsically obvious. Don't bash this into the audience. A truly superior researcher just knows that they and their research is perfect. If you approach your talk with this in mind and a good bit of confidence you will be impressed at how fast your audience falls into line.

One simple trick is to use ambiguous domain specific words. Everyone knows that people who use large uncommon words are smarter. If your audience has to spend time figuring out the words you are using then they will miss some of the results you didn't want them to understand. A good example of a seemingly innocent word is ”feature”. Which when used in a Machine Learning context has a specific general meaning as well as a specific meaning in your research which you can conveniently forget to provide thereby making the audience feel inadequate for not understanding this clearly obvious term. Additionally, audience members from outside the domain will assume they know what the term means and will stay confused for many slides before giving up hope of understanding the clearly complex research you are engaged in.

If you have equations with 20+ variables that are impossible to comprehend without serious study, don't be afraid to show them. Math equations are a truly beautiful construction. They are arguably competently self explanatory and therefor require little to no explanation. They also require a significant amount of attention and brain power to comprehend and map onto your talk topic. A sufficiently high number of complex equations and/or proof statements will often cause a portion of your audience to give up and just assume that you know what you are talking about.

Graphs are very useful in belittling an audience. However, they are such an important topic that they appear all on their own in Section 5.

8 Practice Talks

Giving practice talks to ensure minimal information conveyance is typically frowned on. Really if you have time to give a practice talk you should have been using that time to get better results and therefor not need this guide. However, if you are determined to practice your non-information giving skills there are a few things you can do to make your practice talk more successful.

Give your practice talk audience slide handouts to make notes on. Having something to write on makes the audience members feel more important. Don't worry, if you have picked your audience well most of them won't actually write anything. The handouts are also useful for you to go back through and read the notes. Comments like "where is this going?" or "??!!??" or better yet "impressive!" indicate that your talk is going well. While comments like "why are you using the XYZ methodology instead of the ZYX methodology?" indicate a dangerous level of methodology understanding in your audience.

Only invite people who already know your research. This way they don't actually have to listen to your talk but can give good advice at the end and explain to the other audience members what you actually did in far more glowing terms than you could because unlike you they can exadurate in ignorance. They are also less likely to ask hard questions or nit pick your slides.

Invite the people who came to your practice talk to your actual talk. That way if anyone complains about the talk they will be on hand to explain just how bad it used to be and how impressed they are with your improved presentation.

9 Concluding the talk

Writing a talk conclusion is actually extremely easy and nearly identical to an information imparting talk. Remember that outline slide we created in section 4? Take the same set of bullet points, toss out the Introduction and Conclusion and make each point into a partial sentence. The resulting set of bullet points should read something like this: I told you about this big important problem (Motivation) that no one else has adequately solved (Background), I did something brilliant (Methodology) and completely solved the problem (Results).

Remember to NEVER, EVER introduce new material in the conclusion. Besides the obvious problem of finding more material to add and the risk of an audience member calling you on something, you aren't even supposed to be putting material here. Why would you make life harder for yourself than it needs to be? The audience doesn't want new info, you want to tell the audience as little as possible, everything is good, don't rock the boat.

This is also a good place to remind you that one of the goals of a good information impeding talk is that the audience is happy and feels like you are competent and tried to give a good talk. The conclusion is your last chance to make it clear to the audience that you really did something useful, even if they didn't understand it, and that they haven't wasted the last hour of their lives. The best part is that you can do this without ever stating anything specific. In fact the exact words "I showed that our solution is vastly superior to existing solutions" can be safely said when reading the "results" bullet of the conclusion slide in the majority of talks.

What you proved or how well it worked need never even be mentioned.

10 Simple Tips and Tricks

The lessons above will give you a good start to creating an optimal information impeding talk but there are a few extra embellishments you can add that will make your talk run all that much rougher.

Page numbers Leave them out! Page numbers allow your audience to make effective notes and to refer back to specific points in your talk during their questions. Without page numbers the audience is forced to write more vague notes or copy the titles from your sides which takes more time. Also referencing the slide again during the questions will take time which decreases the number of questions you will get.

Distracting computer messages Leave them on, thereby ensuring those audience members who have managed to follow your work are temporarily distracted and are unable to pick up the train of your talk again. Its best if the distracting messages are innocent looking and therefor not something you may have anticipated. Email notifications are good by they only pop up in the corner and vanish on their own. System messages such as failure to backup or out of battery power are much better as they cause Power Point to exit out of the slide presentation all together buying you several minutes of technological fixing during which you don't have to present. Also, the interruption is sufficient to cause your audience members to loose their train of thought.

Eye contact This is a tricky one. Effective eye contact is important during your talk. You need to use aggressive eye contact to stare down the opposition and challenge them to disagree with you. But be wary of too much eye contact though as you can inadvertently engage your audience and keep them from their mid afternoon nap. Be careful who you look at in the audience and avoid anyone who looks distracted or who might zone out or doze off if left alone.

Talking fast This is an excellent plan if this is how you plausibly normally talk. If you tend to be high strung and talk somewhere near the speed of light there is no reason to slow down for your talk. The faster you talk the less likely your audience will comprehend what you are talking about. If you talk calmly normally you can still safely speed up your speaking by a small margin and blame it on being nervous.

Um, like, uh Talking fast is hard, talking to slowly makes you look, well, slow, so what are you to do? The answer is to add filler words. Similar to the filler found in meat products, word filler has no value and has an unproven negative comprehension effect on the audience. If you are stuck, need time to think, forget what to say, or just need to waste more time on a slide feel free to add in liberal amounts of any of the "um," "like" or "uh" filler words. These words add nothing to your presentation but they do make it a bit harder for the audience to determine what you are saying and they ensure that you are saying something even if you don't have anything to say.

Cluttered slides Do this. While you don't want to assist your audience in understanding exactly what you did you also don't want to be seen as leaving out important information. The simple solution is to hide the needle of knowledge in a haystack of words. The more words on the slide the less likely your audience is to actually read anything. So bury your statistical significance test results in the middle of a paragraph of text and feel confident that you both provided them to your audience and that your audience will never find them.

Talking to the screen instead of the audience Don't do this. While the projection screen or even your computer screen may seem less intimidating than your audience you really shouldn't talk to them. It can make you look a bit crazy and imply that you are not the top notch researcher you are. The goal of your talk is to impress your audience with your supposed research skills, not convince them that you are crazy and think that projector screens can hear you and talk back. Take a deep breath and repeat the following litany against fear to yourself, turn around, and speak to the audience.

> I must not fear.
> Fear is the mind-killer.
> Fear is the little-death that brings total obliteration.
> I will face my fear.
> I will permit it to pass over me and through me.
> And when it has gone past I will turn the inner eye to see its path.
> Where the fear has gone there will be nothing.
> Only I will remain [1].

11 Conclusion

Every year thousands of students woefully discover that their research results fall into the unprestegious Bovik quadrant. Unlike Pasteur's quadrant or Bohr's quadrant, Bovik quadrant research creators would rather no one ever discovered that their research results are both not interesting and not useful.

Bovik quadrant members have been woefully neglected by research talk design experts who focus nearly all their energies on how to create information imparting talks for those lucky researchers with good results. In this work I have attempted to address this inadequacy by providing recommendations on how to create an information impeding talk. Using the skill set gleamed from this work the reader should now be able to 1) convince an audience that you did something worthwhile 2) make sure an audience never figures out what that important thing was.

12 Bibliography

References

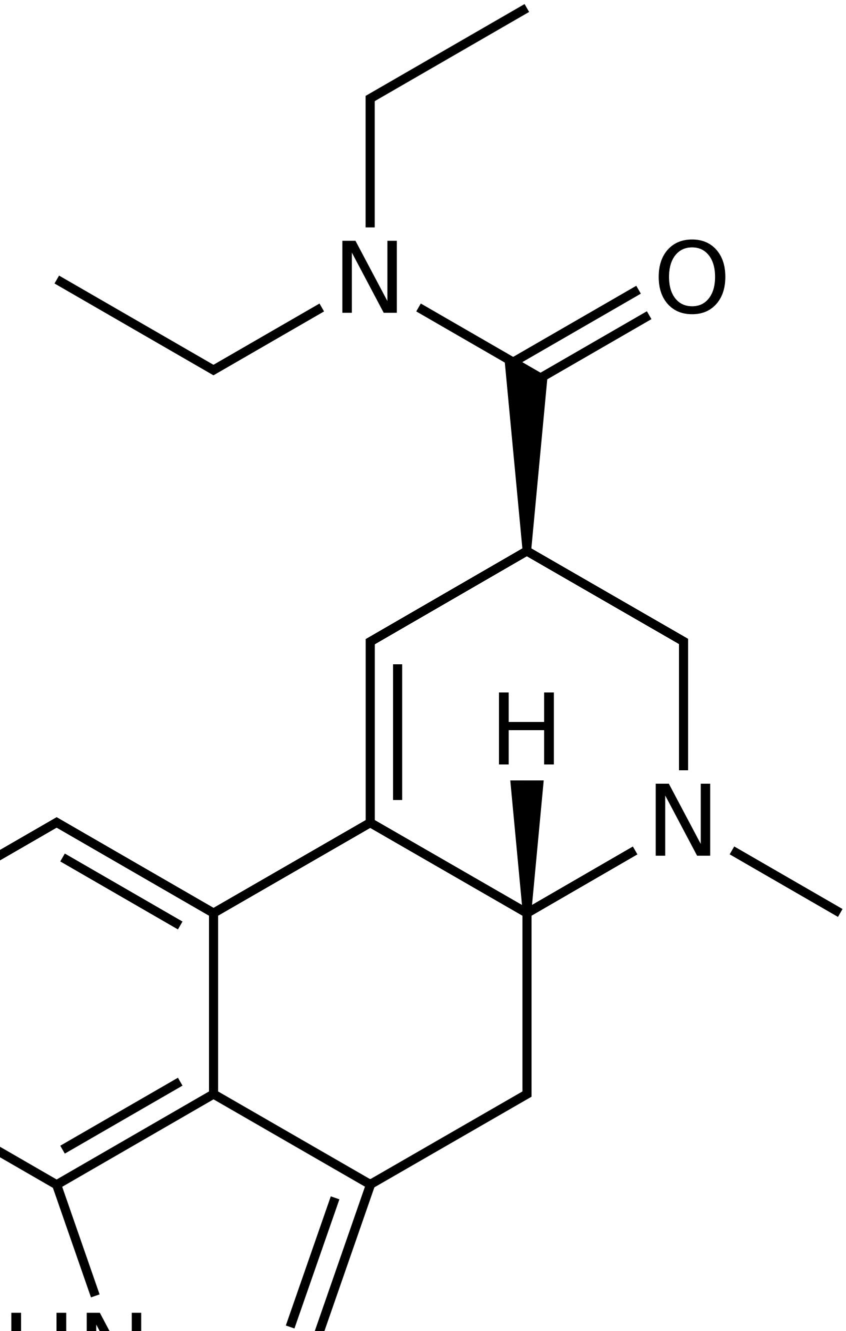
[1] F. Herbert. *Dune.* 1965.

[2] D. E. Stokes. *Pasteur's Quadrant: Basic Science and Technological Innovation.* Brookings Institution Press, 1997.

[3] E. R. Tufte. *The Visual Display of Quantitative Information, 2nd edition.* Graphics Press, 2001.

An Alternate Construction Method for the Surreal Numbers

Jim McCann[*]
Carnegie Mellon University



(Image created by wikimedia commons user Acdx.)

[*] e-mail: jmccann@cs.cmu.edu

The first paper on
SAMIR – ANVESH

SAMIR Sapra and ANVESH Komuravelli
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{ssapra,anvesh}@cs.cmu.edu

Abstract

We introduce *A Novel Verification Environment for Software and Hardware* (ANVESH), which employs *Stateflow Abstraction Modeling for Information Retrieval* (SAMIR) to retrieve information from the Internet, make state machines, and model check them. We employ two novel approaches to achieve this. In this way, astonishingly, we verified everything reachable via the Internet. Results are presented on a 42 Hz Linux box (we were simulating a machine using Simics, and our license didnt let us go any faster). We thus, also, propose the end of the Verification era. Thank you very much, Model Checking, for leading us this far! You will be remembered for ever!

1 Introduction

We present the last and the greatest paper on Model Checking [5]. We describe our novel mega-tool, *Stateflow Abstraction Modeling for Information Retrieval* (SAMIR), which uses novel approaches of lurking and larceny (as part of *A Novel Verification Environment for Software and Hardware* (ANVESH)) to acquire and verify everything reachable via Internet thus satisfying the ultimate and long-cherished goals of Model Checking. This is the long awaited victory of Model Checking. Hereafter, we refer to our tool as the *novel mega-tool.*

A critical feature is the ability to embed novel free-trace webcams into the existing Cloud[1]-computing infrastructure using "Jack and the *i*Beanstalk" technology.

We guess, there is nothing much left to introduce. Without asking you to hold your breath for more time, let's dive into this great work.

2 Related Work

People have earlier worked for quite long on what they thought were the state explosion [1], cloud computing [2] and online model checking [3] problems. But these are completely artificial as we show in this ground-breaking work. To be very frank, no state exploded in state explosion, cloud computing was not on real clouds and online model checking was not online but done when Internet was not connected at all! The goal of this paper is to ignore all this prior "work" and *to explore the real meanings of these terms, to seek out new devices on the Internet and model check them, to boldly go where no model checker has gone before!*

3 What State Explosion actually is . . .

One major difference we have in real world model checking with the prior work is that state explosion in the prior work was completely artificial, bah! We tackle much more realistic and terrifying (and terrific

[1] someone may need a reminding that real Clouds still exist

Figure 1: Poor battery. We feel so sad for it ☹. Hey battery! We know, you are not the culprit! Anyway, the important thing here is that the firmware died a brave death. Long live firmware!

☺) scenarios. For example, we once had an intelligence information[2] in 2005 that our novel mega-tool sneaked into a switched on Nodia 2600 firmware and started model checking! The information also said it *promptly* found a serious bug in the firmware and was ready to fax a counter example to our headquarters at the Mariana Trench using the inherent satellite communication component of the firmware of the device (which was, by the way, captured and model checked by our novel mega-tool before to be on the safe side). But the firmware was smart enough to realise it was about to fall into disrepute and rather than face the inevitable shame, decided to sacrifice itself by sending the battery the following code fragment

```
begin
  panic
  xplod$_{\text{please,pls,pls,...}}$
end
```

This, apparently, resulted in the situation shown in Figure 1[3]. (If you are wondering how we got this pic, we were lucky enough to have our novel mega-tool already sneaked into (and of course, verifying) the Closed-Circuit TeleVision which was present in that room. FYI, the CCTV was model checked successfully!)

As we see from this specific example, the world is not ready to face the shame of getting its bugs exposed. State explosion, is henceforth, redefined as the state of the *world* where an actual explosion takes place when our novel mega-tool is ready to report the bugs!

To give this *paper* its dignity of definition, we pause our discussion here and continue on the next *paper*.

[2] Due to the sincere work by SANIA (SAMIR-ANVESH Information Agency)

[3] Figure courtesy: `http://www.techgadgets.in/`

The second paper on
SAMIR – ANVESH

SAMIR Sapra and ANVESH Komuravelli
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{ssapra,anvesh}@cs.cmu.edu

Abstract

Please refer to the first paper on SAMIR-ANVESH.

But, some reader may say that, a part of the world may not care about its bugs. To make the world look uniform, our novel mega-tool exploits the bugs to sneakily elevate its privilege within the buggy devices in this shameless part of the world and induces a state explosion as implied by the following lemma.

Lemma 1 *Given a device in the shameless part of the world, our novel mega-tool can sneakily elevate its privilege which enables state explosion.*

Proof. Let d be a device in the shameless part of the world. Because d is shameless, it got to have some bug b for which it is not shameful. Given the novelty of our novel mega-tool, it will *promptly* find b. We stress, d is *shameless.* This implies d does not feel shame if its privileges can be stolen. And we stress, our novel mega-tool is novel! The rest of the proof is obvious. □

4 Model Checking using Clouds

In this section, we describe our infrastructure using Clouds (mind you, the approach is novel enough to use the real Clouds). We also present our two major novelties, namely, the *Lurk-based sneaking* (Section 4.1) and the *Theft-based sneaking* (Section 4.2) which exploit this infrastructure to model check everything reachable via Internet. We first prove a useful lemma.

Lemma 2 $\forall t : \texttt{time}(\texttt{survive}(Internet, t) \supset \exists c : \texttt{Cloud} \;\; \texttt{monkeying\_around}(c, t))$

Proof. Proof by complete structural induction on the graph of the world (technically we consider its MST rooted at Pittsburgh):

Base case: Pittsburgh

As long as there is Pittsburgh, there will be Clouds. So, we can easily find a cloud which monkeys around the world. And Pittsburgh isn't going anywhere, anyway! So, holds for all t.

Inductive step: Suppose the hypothesis is true for all graphs smaller than given graph $\mathcal{T}$. Want to show: claim holds for $\mathcal{T}$.

Note that the base case (Pittsburgh) isn't going anywhere. So we have plenty of clouds for everyone in $\mathcal{T}$. Rest of the proof same as the base case.

□

Now that, we do not to worry about the existence of Clouds, we explain our novel way of use of them. Inspired by the free-food webcam, generously donated by Luis von Ahn and operated by Dec/5 Inc., we introduce the novel concept of *free-trace webcam*. A free-trace webcam is installed using the fabled "Jack and the *i*Beanstalk" technology. Initially, a robo-Jack installs a free-trace webcam on top of the *i*Beanstalk when it just sprouts. As the *i*Beanstalk grows, the webcam inches more towards the Clouds (see Figure 1)[1] and when the *i*Beanstalk ultimately reaches a Cloud, the webcam glues itself to the Cloud. Whenever the Cloud makes a move, it gets detached away from the *i*Beanstalk. These special webcams are manufactured to be solar-powered to avoid exorbitant losses to Do-cane Light Company and equipped with wireless World Area Network cards to avoid exorbitant losses to (`cast`)Com. This webcam keeps recording the trace as the *i*Beanstalk grows.

Figure 1: Our webcam happily rising to the Cloud after a robo-Jack installed installed it on the *i*Beanstalk.

4.1 Major Novelty 1 - Lurk-based sneaking

One of the key components of our novel mega-tool is property inference via a *Lurk*[2]*-based sneaking*. The idea behind this approach is simple and yet elegant (not to mention, novel and to the best of our knowledge, a first attempt to think outside previous boxes; and our knowledge is the best, to the best of the our knowledge, lol ☺). To make the presentation (too!) simple and yet elegant, we quote below a message sent by our tool while using this novel approach.

> *Yo! Waz up? Well, I am just a-lurkin'. These brainless guys (traces, though) are runnin' and runnin' far too lon'. Looks like I don't have anythin' to report yet! Sorry buddy! I should wait till they are done and then, you know, I shud prolly get busy inferring the damn properties and only then, can I get into my actual business. So..I'll keep lurkin'. Cya.*

In a formal lingo, what this message is trying to say is, the tool lurks around a host, observing the execution traces (we call this novel mega-technique, *observing the observables*) generated while the host keeps running. Once the host is done, the tool uses learning techniques to infer properties that the host has to satisfy. And the business our tool was talking about is to lurk around more and wait for *observables* which refute these properties. We pause here to continue the discussion in the next *paper*.

[1] Figure courtesy: `http://www.teachershelp-teachingresources.co.uk/`

[2] Oh! which, by the way, means to lie in wait, lie in ambush, behave in a sneaky and secretive manner

The third and the last paper on SAMIR – ANVESH

SAMIR Sapra and ANVESH Komuravelli
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{ssapra,anvesh}@cs.cmu.edu

Abstract

Please refer to the first paper on SAMIR-ANVESH.

How does the tool know about the host? We use our novel free-trace webcams to eagle-eye for free execution traces generated by hosts on the Internet. As an immediate example of lurk-based sneaking, we recall the technology used for transporting the webcams on top of the Clouds. We noted before that, a webcam records the trace as the iBeanstalk grows. Now, once it gets glued to a Cloud, it hands over the information of this trace to our lurk-hungry tool which immediately detects that the iBeanstalk is going to generate a memory-out-of-bounds exception. An iBeanstalk is humble by construction. It gets ashamed of itself and creates a state explosion, killing itself. (By the way, this also removes a huge obstacle for the webcam's view!)

4.2 Major Novelty 2 - Theft-based sneaking

As a useful side-effect of our free-trace webcams, our tool gets the novel information about the sneakable devices reachable via Internet. Then, we use the novel technique of *theft-based sneaking* [4] and novelly apply a novel application of it by sneaking into the device and stealing some CPU cores for model checking. It also novelly elevates its privileges to be able to muck around with the firmware to build a Kripke structure (this is as good as it gets in Model Checking) and model check for various safety and liveness properties.

To show the applicability of this novel theft-based sneaking, we present the best example model checking could have ever achieved. This is a report from one of our faithful (and grateful) sufferer of our novel mega-tool, in her own sweet words (and also melodious, but unfortunately, we are unable to provide the voice memo),

One fine day, this awesome thing happened. I was checking my email on my iPhone and I suddenly get this message flashed on top of the screen - "Congratulations! Your smartphone is now completely verified! Don't worry about any further bugs either in the hardware or in the firmware or in any of the apps you might be using or even any of the updates you might possibly have in future! Your device is completely bug free. 100%. *Enjoy using your iPhone. SAMIR and ANVESH!" Simply brilliant! I don't need to worry about bugs in my device at all! I love Apple!*

She also sent us a proof-by-image (see Figure 3)[1].

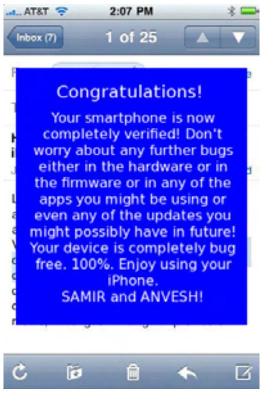
Unfortunately, the poor customer thinks this is a miracle of Apple, but we do not mind.

5 Side result

Theorem 1 *Eventually, the world will by completely bug-free. Indeed, we are currently in that time.*

Proof. *Drumroll...* Trivially true! □

[1] Figure courtesy: http://mactoids.com/



Congratulations!

Your smartphone is now completely verified! Don't worry about any further bugs either in the hardware or in the firmware or in any of the apps you might be using or even any of the updates you might possibly have in future! Your device is completely bug free. 100%. Enjoy using your iPhone.
SAMIR and ANVESH!

Figure 3: The happy message shown to the user by our mega-tool! This manner, we made many people happy. Yo SAMIR and ANVESH!

6 Future Work

Wth? No future work left! Go and sleep!

7 Conclusion

We successfully deployed our novel mega-tool to model check everything reachable via Internet after successfully bootstrapping on our dear machine. See abstract for details of the machine. Hats off to Model Checking! You did it!

Acknowledgement. We wish to acknowledge the organizers of SIGBOVIK for giving us this wonderful opportunity to check models on the Internet. We also want to acknowledge Jim Skees for giving us Facilities, and the SCS Help Desk for the dearest machines, Guy Blelloch for the fancy building where our ground-breaking research was conducted, and Prof Todd Mowry for sneaking ECEs Simics [6] license into the building (inspired by our novel application of *theft-based* sneaking). Refer to the rest of the doc for possible inlined acknowledgements.

References

[1] Peter Bovik and Harry Q. Bovik: *State explosion and beyond*, In the proceedings of the 1st International Conference on State Explosion, Bovik Publishers, Pinatubo, 15 June, 1991.

[2] Mr. Fredrickson: *My experiments with cloud computing*, In the proceedings of the −29th Atmospheric Conference on Clouds, Disney Publishers, South America, 1980.

[3] Ms. Confusee: *Checking Models as they are being generated*, In the proceedings of the 100th Offline Conference on Online Model Checking, Offline Publishers, A Confused Place, 2003.

[4] Jim McCann and Ronit Slyper: *A theft-based approach to 3D object acquisition*, In the proceedings of the 6th Binarennial Conference in honor of Harry Q. Bovik, 2007.

[5] `http://en.wikipedia.bovik.org/wiki/Model_checking/`

[6] `http://www.simics.net/`