A Record of The Proceedings of SIGBOVIK 2008

April 6th, 2008
Carnegie Mellon University
Pittsburgh, Pennsylvania USA

http://sigbovik.org/

A Message From The Organizing Committee:

The Association for Computational Heresy Special Interest Group (ACH SIGBOVIK) on Harry Q. Bovik is particularly excited to be presenting this, the Second Annual Intercalary Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's $(2^6)^{th}$ Birthday. The previous "Binarennial Scheduling" of the SIGBOVIK conferences has been seen as a game-changing development in the storied history of conference presentation ever since George P. Burdell's historic keynote talk at the first SIGBOVIK in 1944 on the occasion of Harry's $(2^0)^{st}$ birthday.

However, the response to what was is generally accepted as the seventh SIGBOVIK Conference[1] in 2007 was so overwhelming that the ACH SIGBOVIK Governing Board was forced to recognize that it would be simply negligent to allow the crucial work that finds unique expression at SIGBOVIK to lie dormant until 2071. Therefore, the Intercalary Workshops were announced, and this, this Second Intercalary Workshop in Celebration of Harry's $(2^6)^{th}$ Birthday, is the first of such intercalary workshops that will help to advance the progress of science until 2039, when the Thirty-Second Intercalary Workshop in Celebration of Harry's $(2^6)^{th}$ Birthday will be held concurrently with the First Intercalary Workshop in Expectation of Harry's $(2^7)^{th}$ Birthday.

We hope that you will find yourself edified and enlightened by this, the proceedings of SIGBOVIK 2008. We, the pseudonymous SIGBOVIK 2008 Organizing Committee, are proud to present it, and we thank QVT Financial LP, who boldly went where no corporation has gone before: to sponsorship of SIGBOVIK.

Sincerely,
The SIGBOVIK 2008 Organizing Committee:

Ciel Elf	General Chair
Guy Fantastic	Program Chair
Emcee M.C.	Assistant Program Chair
Tux Rat	Publicity Co-Chair
Forbes Anchovie	Publicity Co-Chair
Emmet O'Theorem	Design Chair
Julia Cette	Pre-Program Chair
Turing T. Turing	Procurement Chair
Alga Rhythm	Webmaster Chair
Alf A. Berry	Local Arrangements Chair
Reginald "Red" Acted	External Relations Chair

---

[1] The second and third SIGBOVIK conferences technically did not happen due to Harry's approximately six-year "phase" from early 1945 to late 1950, and the seventh SIGBOVIK conference (on Harry's $(2^6)^{th}$ birthday) was confusingly introduced in the proceedings as the sixth, as the organizers counted from one instead of zero.

- The SIGBOVIK Committee For Fixing The Numerical Errors In Last Year's Introduction

TABLE OF CONTENTS

Track 1: Chuck Norris

An Incentive-Compatible Mechanism for all Settings: Chuck Norris

Michael Dinitz
Computer Science Department
Carnegie Mellon University
mdinitz@cs.cmu.edu

April 6, 2008

Abstract

A very important property for any mechanism is *incentive-compatibility*. A mechanism is incentive-compatible if no agent has an incentive not to follow the protocol. In this paper we present a new meta-mechanism that can be applied to any existing mechanism to make it incentive compatible. This meta-mechanism is Chuck Norris. This shows that any mechanism can be made incentive-compatible, and thus the field of mechanism design is now solved.

1 Introduction

Mechanism design is the attempt to design protocols where each agent has their own selfish goals and is rationally attempting to optimize them. This selfish rationality may result in agents refusing to participate if they cannot benefit, or if they participate they might lie or refuse to follow the protocol in some other way in order to maximize their utility. The goal is to design mechanisms that are *incentive-compatible*, in which every agent has no incentive to deviate from the specified protocol. We would also like our mechanism to maximize the *social welfare*, usually defined as the sum of the utilities of all of the agents. In many cases it is difficult to develop mechanisms that are incentive-compatible and social welfare maximizing, as sometime maximizing the social welfare will involve punishing one agent in order to make the others happy, and thus this one agent will not be incentivized to participate or to follow the protocol. For a more in-depth introduction to the field of algorithmic mechanism design and it motivations, see [1].

In this paper we prove the existence of incentive-compatible mechanisms in all settings. We do this by constructing a *meta-mechanism* that can be used to transform any existing mechanism to make it incentive-compatible. This meta-mechanism can be described in two words: Chuck Norris.

2 Main Result

Suppose that there are agents $x_1, \ldots, x_n$, and the possible outcomes of the protocol are in some set $\mathcal{S}$. Each agent x_i has a utility function $u_i : \mathcal{S} \to \mathbb{R}$. For every $s \in \mathcal{S}$, let $v(S) = \sum_{i=1}^{n} u_i(s)$ be the value (i.e. the social welfare) of the solution. Let $\mathsf{OPT} \in \mathcal{S}$ be the optimal solution, so $\mathsf{OPT} = \text{argmax}_{s \in \mathcal{S}} v(s)$. We say that a mechanism is an α-approximation if it returns a solution $s \in \mathcal{S}$ such that $v(s) \geq v(\mathsf{OPT})/\alpha$.

Suppose there is some mechanism $\mathcal{A}$ which, if all agents follow the mechanism, is an α-approximation. Let $\mathcal{A}^{CHUCK}$ be the following mechanism. First, any agent that does not participate gets a visit from Chuck Norris, who then proceeds to roundhouse kick the agent. We then proceed according to $\mathcal{A}$, but any time an agent interacts with another agent or with the mechanism Chuck Norris roundhouse kicks them if they do not follow the protocol.

Theorem 2.1 *$\mathcal{A}^{CHUCK}$ is incentive-compatible.*

Proof: A Chuck Norris-delivered roundhouse kick is the preferred method of execution in 16 states [2]. Thus the utility to an agent of any solution which involves being roundhouse kicked by Chuck Norris is $-\infty$, since that is the utility of death. It is easy to see from the definition of $\mathcal{A}^{CHUCK}$ that any deviation from $\mathcal{A}$ by an agent will result in a Chuck Norris roundhouse kick, and hence a utility of $-\infty$. So all agents will follow $\mathcal{A}$, and thus $\mathcal{A}^{CHUCK}$ is incentive-compatible. ■

The following corollary is almost immediate:

Corollary 2.2 *$\mathcal{A}^{CHUCK}$ is an α-approximation*

Proof: Recall that $\mathcal{A}$ is an α-approximation if all agents follow the protocol. Since $\mathcal{A}^{CHUCK}$ is incentive-compatible we know that all agents will follow the protocol. And except for possible Chuck Norris roundhouse kicks $\mathcal{A}^{CHUCK}$ follows $\mathcal{A}$ exactly, so $\mathcal{A}^{CHUCK}$ is also an α-approximation. ■

3 Discussion

In this section we discuss possible objections to the Chuck Norris meta-mechanism. One possible problem is synchronous actions: if multiple agents are all taking actions at the same time, then they all have to be threatened by Chuck Norris, not just one of them. This is not a problem, though, since a little-known (but very useful) folk theorem states that "Contrary to popular belief, there is indeed enough Chuck Norris to go around" [2]. A related objection is that, even if Chuck Norris is physically able to administer a roundhouse kick, non-compliance with the protocol might involve simply misreporting private information, and thus Chuck Norris would not be able to determine whether or not the protocol was followed. But this is false, since Chuck Norris has the ability to read minds [2].

Finally, there is the possible issue of the utility of Chuck Norris himself. After all, we crucially depend on his roundhouse kicks, and while he obviously has the ability to roundhouse kick whomever he wants, he might not have the desire. Fortunately an examination of the other agents makes it clear that Chuck Norrs would indeed derive utility from administering roundhouse kicks to the bad agents. This follows from the fact that any agent which does not follow the protocol has decided to ignore the threat of a Chuck Norris roundhouse kick. This is obviously a foolish thing to do, and while "Mr. T pities the fool, Chuck Norris roundhouse kicks the fool's head off" [2].

4 Conclusion

In this paper we have shown that any mechanism can be made incentive-compatible by using Chuck Norris. This essentially solved all open problems in the field of algorithmic mechanism design. Thus Chuck Norris can add "solving all problems in algorithmic mechanism design" to his formidable list of accomplishments.

References

[1] N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140, New York, NY, USA, 1999. ACM.

[2] C. Norris. http://www.chucknorrisfacts.com/.

Track 2: Complexity Theory

A non-non-destructive strategy for proving P = NP

Tom Murphy VII

1 April 2008

Abstract

We provide a radical new approach for proving that $\mathbf{P} = \mathbf{NP}$, demonstrating that if you put your mind to it, you can accomplish anything!

Keywords: complexity theory, p, np, prongs, hyper-driven devices

1 Introduction

The field of Computers Science has been fairly successful in answering its "grand challenge" questions. For example, in 19XX Computers Science answered in the affirmative, "Can a computer beat a human in Chess?" In 20XX we successfully built a Windows Vista. In 20XX Computers Science answered in the affirmative, "Can a computer not be beaten by all humans in Checkers?" However, some problems are still unsolved. Most vexing of these is the question of $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ [Cook(1971)], that is, are nondeterministic Turing Machines inherently more efficient than deterministic ones?

This is troublesome for a number of reasons. First, the existence of unsolved problems adversely affects our "batting average," the primary means for comparing Computers Science to other fields of import. (This also has an indirect effect on other comparative statistics, such as the Earned Run Average of competing fields.) Such tarpits also waste countless hours of ambitious graduate students's most creative years, and the time and patience of program committee members for second- and third-rate conferences. The open proposition also induces additional market volatility, as futures markets[1] and institutions such as the Clay Mathematics Institute have placed bounties totaling \$1m USD on its resolution—in either direction [Cook(2000)]. One is also subject to the eerie suspicion that some hyperintelligent observer is chuckling at our Sisyphean impotence, our endless attempts at the same dead-end strategies and the puppy-like faithfulness with which we return to the problem and continue to sing its praises. Time's up. Put your pencils down and pass your papers to the front of the class; the problem must be solved *now*.

In this paper we present a new strategy for proving that $\mathbf{P} = \mathbf{NP}$. This approach differs from those that came before it in methodology and consequences: It is inherently non-constructive (indeed, non-non-destructive) for one, meaning that we cannot directly use it as an effective means for solving difficult (NP-hard) problems in polynomial time. In fact, the result makes the computational landscape less efficient in general.

To begin, let us refresh our memories as to the statement of the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ problem so that we can attack it where it is most weak.

2 Problem statement

The set of languages $\mathbf{P}$ is defined as follows [Cook(2000)].

$$\mathbf{P} \quad = \quad \{\ell | \ell = L(M)\} \quad \text{for some Turing Machine } M \text{ which runs in polynomial time}$$

where ℓ is a language and $L(M)$ is the language accepted by the machine M.

[1] Although no longer posing any risk to investors, the stillborn ACM–NASCAR crossover *Turing Machine 0x500* debacle—in which "races" complete with pace rabbits were staged between different Turing Machine programs solving various problems with unknown complexity bounds, and holiday Vegas bettors would have their pensions cleaned out by computational savant bookmakers in smoky but mostly empty parlors designed to resemble mainframe machine rooms—could also have been avoided had $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ been solved prior to its inception.

Similarly, the set of languages **NP** is defined as

$$\mathbf{NP} = \{\ell | \ell = L(N)\} \quad \text{for some non-deterministic Turing Machine } N \text{ which runs in polynomial time}$$

where a non-deterministic machine is defined in the usual way.

Proving (or disproving) that deterministic and non-deterministic machines describe the same set of languages (by, for example, establishing a polynomial-time solution to an NP-hard problem, or giving a lower bound for one) is famously difficult. In this paper we take a completely different approach. The key observation is the implicit existential quantifier in the definitions of **P** and **NP**: A language ℓ is in **P** if there *exists* a polynomial time machine M such that $\ell = L(M)$. We present a multi-pronged attack on existence by metaphysical arguments, non-non-destructive techniques, and complexity class mobility.

Figure 1: An example of a multi-pronged attack on Turing Machines.

3 Do any Turing Machines exist?

The first question we can ask is: Do any Turing Machines actually exist? If not, then the languages **P** and **NP** are empty, and trivially equal. One can make a reasonable case that, in fact, there are no Turing Machines; the machines require an infinite-length tape, an object that many object to the existence of in the physical universe. (Some argue that the lack of evidence for infinite tape is actually a planned obsolescence conspiracy by the 3M corporation, and that infinite rolls of tape are in fact present in their underground laboratories.)

However, even if Turing Machines do not exist in the physical universe, most Mathematicians and Computers Scientists would be prepared to accept the existence of Turing Machines within the Platonic universe of idealized mathematical objects. Here, infinite tape is in abundant supply. The Platonic universe fortunately also affords the ability to carry out many other feats of the mind, which abilities we use in the next prong.

4 Destroy all Turing Machines

Supposing that Turing Machines do already exist, and we find this to not be desirable, we still do have recourse.

For example, many systems for formal mathematics such as The C Programming Language and LaTeX support the ability to remove or alter definitions in the environment (for example through `#undef` and `\renewcommand`). Why should not these constructs of human thought be available to us in the Platonic universe? Specifically, why should we not be able to *make* Turing Machines not exist by the power of human thought alone? The traditionally non-destructive nature of the Platonic universe compels us to forever recall our inconvenient mistakes. This is, frankly, some intolerable bullshit. Are Computers Scientists ready to admit that their thoughts are not powerful enough to undo their own other thoughts?

Effecting this change might not be so simple. The Platonic universe is a mathematical commons shared by all clear thinkers. Observing our weakness and our attempt to subvert it, competing fields may very well cause Turing Machines to come back into existence by redefining them to their current pernicious meaning. Maintaining a force of constantly vigilant Computers Scientists to battle the existence of Turing Machines could be as wasteful as attempting to decide $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ through conventional means. Instead, we should use our creative powers to populate the competitive idea landscape with countermeasures

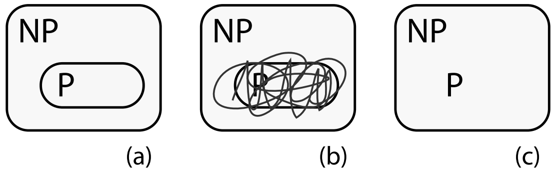


Figure 2: A non-non-more-destructive approach. (a) is the normal inclusion diagram for **P** and **NP** in the absence of an answer to $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. In (b), for each language in **P**, we forget all polynomial time algorithms. This leaves only the exponential time solutions, making $\mathbf{P} = \mathbf{NP}$ (c).

to prevent Turing Machines without constant attention. For example, I am currently imagining a mystical boomerang-like five-pointed Glaive weapon that has been rescued from a lava cave such as like in the 1983 heroic fantasy film *Krull* and which has the ability to chop up a Turing Machine's big ol' tape like superheated tungsten piano wire through a deciliter of I Can't Believe It's Not Butter brand butter-like spread. This weapon I've imagined is chopping up Turing Machines at a rate of like an Ω stack of Ωs every second, and I'm just getting started (Figure 1)! By populating the Platonic universe with such non-non-destructive thoughts, we can keep it essentially clean of working Turing Machines and simultaneously produce Platonic block-buster films on the cheap.

5 Rise Up! Up the polynomial hierarchy!

On the other hand, many people have become rather attached to computation and its useful fruits. What if we are unwilling to abandon computation altogether? A second approach is inspired by the asymmetry in the difficulty of the $\mathbf{P} = \mathbf{NP}$ question: It is very easy to prove that $\mathbf{P} \supseteq \mathbf{NP}$ but difficult to prove that $\mathbf{NP} \supseteq \mathbf{P}$. Rather than refute the existence of Turing Machines, we achieve $\mathbf{P} = \mathbf{NP}$ by "forgetting" all of the polynomial time algorithms for solving problems in **P** (Figure 2). Since deterministic exponential time suffices for solving every problem in **NP** (by exhaustive search), if we also have only exponential time algorithms for solving problems in P, then these two complexity classes will be equivalent.[2] As a result, our existing computer programs will run more slowly, but we will at least simultaneously be able to have computation and a satisfactory solution to the vexing $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ problem.

[2]PS. My Krull weapon flying around idea space is preventing you from noticing that this argument does not make sense (Figure 1).

6 Related work

Others have proposed trivializing solutions to the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ problem, such as the algebraic solutions $N = 1$ or $P = 0$. This is pretty dumb.

7 Conclusion

I hereby authorize the Clay Mathematics Institute to direct deposit $1m USD into my bank account, routing number 7474-133-790.

References

[Cook(2000)] Stephen Cook. The P versus NP problem, May 2000. URL `http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf`. Clay Mathematics Institute Millennium Problems.

[Cook(1971)] Stephen Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

O(0) Algorithms and Other Applications of Time Travel

David Klionsky
Carnegie Mellon University

April 1, 2008

Abstract

Asymptotic analysis has hit an asymptote in its ability to classify algorithmic complexity. We propose a new order of functions, zero-time functions, to classify the set of functions that terminate before they are run, using novel applications of age-old time travel techniques. Other topics include the McFly Theorem (an extension of the Master Theorem), the Bill and Ted (BT) class of algorithms named for its most excellent founders, and the Primer Conjecture which we are certain no one understands. We also prove that P does indeed equal NP. We've been to the future, people, just trust us on this one.

Track 3: Lies, Damn Lies, and Applications

Towards a Frequentist's Approach to Pascal's Wager

Mary McGlohon Robert J. Simmons*

April 6, 2008

Abstract

Pascal's wager attempts to provide a mortal with a proper choice of believing or not believing in a god, based on the expected reward of a given belief. It is essentially a Bayesian approach to the existence of a supreme being, as it deals with a *degree of belief* approach to proability. However, given the ineffability of a supreme being, the idea of finding a Bayesian prior for performing inference is impractical. However given the high population of observable mortals, a frequency probability would be a more obvious choice. Therefore, we present a systemized frequentist approach to the problem of a supreme being.

	CHICKEN SERVED	STEAK SERVED
BRING WHITE WINE	7	-2
BRING RED WINE	3	11

	GOD DOES NOT EXIST	GOD EXISTS
BELIEVE GOD EXISTS	k+α	∞
DO NOT BELIEVE	k	k

Figure 1: Diagramming the utility in situations where there is a choice you have control over (&) and another choice you do not have control over (⊗). On the left, bringing wine when fish or steak may be served. On the right, believing in God when God may or may not exist.

1 Introduction

1.1 Pascal's Wager

We follow Giden Rosen's description of Pascal's Wager [8], also know as Pascal's Gambit. First, we note that is possible to drawn a chart that describes the different choices available to an actor on the y-axis and the different possible states of the world (which are assumed to be unknown to the actor) on the x-axis. As with most problems in life, this problem can be recast in terms of alcohol [9]; in particular a situation in which the actor has the option of bringing red or white wine to a friends' house without knowing whether chicken [7] or wine will be served for dinner.

As everyone knows [1], chicken with white wine is pretty good, but chicken with red wine is so so, whereas steak with red wine is freakin' amazing but white wine with steak is no good. By assiging a numerical value to the utility of each of these combinations, we can obtain the graph in Figure 1. A risk averse actor would be inclined to bring red wine unless there was no possibility of steak being served; however, the behavior of a fully rational actor will be determined by the probabilities they assign to the different possibilities. Presuming that there is an equal probability of either possibility, then the expected utility of bringing white wine is $7 \times .5 + -2 \times .5 = 2.5$, whereas the expected utility of bringing red wine is $3 \times .5 + 11 \times .5 = 7$, and the rational actor will bring red wine, as the expected gain for doing so is 4.5 units. On the other hand, if the rational actor thinks that there is an 80% chance of chicken being served, then the expected utility of bringing white wine is $7 \times .8 + -2 \times .2 = 5.2$, whereas the expected utility of bringing red wine is $3 \times .8 + 11 \times .2 = 4.6$, and the rational actor will bring white wine, as the expected gain for doing so is .6 units.

Pascal's Wager seeks to extend this ordinary and legitimate reasoning to the case for belief in a deity. One version of the argument imagines that there is an inherent utility of a human life, k, and that the value of $k + \alpha$ is the value of a life lived acting under the belief in the existance of God. Some versions of Pascal's Wager take α to be positive, some negative, typically depending on how much people like guilt and/or Gregorian chant. Then, posit that either no God exists, or else there is a God who rewards His believers with eternal bliss – we will describe this God as a "rational rewarding" God. Definitionally, we can assume that the utility of "eternal bliss" is infinite, and seeing as $k + \alpha + \infty = \infty$ as long as k and α are finite, we end up with the chart on the right-hand side of Figure 1.

At first glance, we must expect an actor to assign some probability of the existence of God, and some other probability to the non-existence of God, and proceed by the same analysis we used for deciding whether to bring wine. If we assign that God exists with probability p, then the expected utility of non-belief is $k \times (1 - p) + k \times p = k$, and the expected utility of belief $(k+\alpha) \times (1-p) + \infty \times p = \infty$, and so a rational actor should believe in God, as the expected gain for doing so is a rather persuasive ∞ units.

*The results contained herein reflect neither the opinions of the authors, nor those of the National Science Foundation.

	NO GOD EXISTS	RATIONAL. REWARDING	PERVERSE. ACTIVE	RATIONAL. PUNISHING	RANDOM. REWARDING
BELIEVE GOD EXISTS	k’	∞	$-\infty$	k’	∞ (.8) $-\infty$ (.2)
DO NOT BELIEVE	k	k	∞	$-\infty$	k

Figure 2: A philosopher-mathemetician’s critique of Pascal’s Wager

1.2 A Mathematical Critique

The historical critique of Pascal’s Wager, as described by Giden Rosen [8], falls into two categories. The first is a theological critique; in a world where multiple religions teach eternal punishment or reward for belief/nonbelief in their god, Pascal’s construciton gives little-to-no guidance for the problem of picking “the right God.” This problem will not be considered in this paper due to restrictions [4], and in any case, this critique of Pascal’s Gambit is well-understood.

A more basic mathematical critique begins with the idea that the infinities in present in Figure 1 are suspect from a mathematical point of view. We can drive this concern home by assigning non-zero probability to a God which we call “perverse, active” and which Rosen describes as philosopher-friendly. “I didn’t give them any evidence of existance,” this God thinks, “and by golly, those non-believers, they stuck to their guns. I’ll give them eternal bliss, and give the believers eternal punishment.”

Now the non-believer has an expected utility of ∞, and the expected utility of the believer is... one must suppose, impossible to calculate. We can add even more absurdity to the Pascal argument by positing the existence of a God that sends believers to heaven or hell with probability .8 and .2, respectively, whilst leaving nonbelievers alone. The analysis used in descriptions of Pascal’s Wager becomes completely inadequate in this environment, though one must assume in such a universe non-belief and risk aversion would have to be linked.

1.3 A Frequentist Critique

Since the question of using Bayesian or frequentist approaches to statistical analysis is a nearly religious debate in the field [5], the obvious extension is to apply it to relgious matters. Furthermore, it assumes the “gambling god” to be introduced later– or, more generally, a god that does not consider gambling a punishable sin.

On the other hand, we do have billions of observable mortals, so assigning a frequency probability to the existence of a supreme being would be a more natural way of going about things in the supernatural realm.

2 Methodology

In order to determine an appropriately frequentist, we needed a sample space of universes . We wanted to investigate a wide variety of possible of potential God-models, including Gods that behave rationally (consistently rewarding those that believe in them), perversely (consistently punishing, or failing to reward, those that believe in them), or arbitrarily (meting out eternal reward or punishment in a manner that is only rational with some probability, which may or may not be contgient on belief).

2.1 Sampling

2.1.1 Rapture-Recapture

We introduce a novel method of sampling for supernatural experiences, which we call *Rapture-Recapture*. We first chose at random 100 people from each of 6 universes: Earth, Bizarro, World of Warcraft, Star Trek, Star Trek Mirror Universe, and the Buffyverse. We surveyed each subject regarding their beliefs in god, humanity, and their own sins. We then tagged the right ear of each subject and euthanized them. After some period of time we performed a re-capture and again surveyed each re-captured subject on their posthumous experiences.

2.1.2 Entrance survey

Before euthanization, we presented each subject with an extensive survey with questions regarding their faith, time spent on earth, and other necessary information to obtain before euthanization. The survey is included in Appendix A.

2.1.3 Euthanization

We then attempt a re-capture through wireless transmission. As we assume that everyone in heaven gets a free iPhone, and everyone in hell gets a Bluetooth Ouija Boards, we ensure that our hardware is compatible with both.

Zombiefication was also used as a backup method of obtaining posthumous survey data. It was only used as a backup, as the IRB would not approve the proposal to use zombiefication and revive people already in heaven.

2.1.4 Exit survey

Of the re-captured subjects, we obtained a completed survey from each, shown in Appendix B.

3 Results

Results from some of the universes sampled are presented.

Figure 3: One of the huntards that nearly killed the undergrad we hired to gather data from Azeroth. (picture courtesy of `www.figurerealm.com`)

3.1 World of Warcraft

The World of Warcraft (WoW) universe, termed *Azeroth*, has a number of interesting differences that often were an advantage for our experiment. A resurrection (*rez*) system is in place, in which players spend some amount of time essentially dead while their disembodied soul has to run from the *graveyard* back to the place they were *ganked*; this is known as a *corpserun*. During this time they are still able to use *voice chat* to communicate, which made our devices described earlier unnecessary.

Several difficulties arose in performing the rapture-recapture. It was difficult to get an unbiased sample, as whenever we tried to use subjects from *parties*, particulary *pick-up-groups* that included *paladins* (sometimes *priests* and *shamans*, because those were usually *n00bs* (or *n00badins*) with no respect for science and tended to interfere by casting healing spells upon our subjects or prematurely rezzing them. Secondly, on several occasions some *huntard* would sic their *pet*, usually a tiger, on the experimenters (see Fig. 3). Thirdly, occasionally *warlocks* stole the souls of dead characters and captured them in *soulstones*. Since we considered that to the an interruption of the normal rapture-recapture experiment, we were unable to use those data.

Results were somewhat inconclusive. Despite the built-in ease of communicating with un-rezzed characters, usually they went *AFK* (away from keyboard), as if ordering pizza were more important than the progress of science.

3.2 Bizarro World

The Bizarro World of *Htrae* functions in every way imaginable opposite of planet Earth. Very pleasingly, we thereby found opposite results. While we inferred from exit surveys that 10% of earthly subjects went to some version of eternal bliss, 90% of Bizarro subjects did.

4 Conclusion

We have not had time to fully analyze the results, and periodic demonic posession by our Subversion server has been a constant source of `<<< mine, all mine! bwahahahaha ==== >>> r666 ====` We are confident that our data sets will be a useful for future study. Hey, we put "Towards" in the title, didn't we?

References

[1] Like, duh.

[2] `http://www.saintaquinas.com/mortal_sin.html`.

[3] `http://www.icanhascheezburger.com`.

[4] William Kristol. Oh, the anguish!: The cartoon jihad is phony. *The Weekly Standard*, 11(22), February 2006.

[5] John Lafferty and Larry Wasserman. *All of Statistical Machine Learning*. Pink Book Publishers, 2012.

[6] Jim McCann and Ronit Slyper. A theft-based approach to 3d object acquisition. In *SIGBVOIK*, 2007.

[7] Mary McGlohon. Fried chicken bucket processes. In *The 6th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, April 2007.

[8] Gideon Rosen. Pascal's wager, 2002. Lecture, Introduction to Metaphysics and Epistomology, Princeton University.

[9] Robert 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*, April 2007.

[10] David Steiner. Proof-theroetic strength of pron with various extensions, 2001.

[11] Tom Murphy VII. *Name of Author by Title of Book*. Lulu Press, 2003.

[12] Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, Tom Murphy VII, and Tom Murphy VII. Level of detail typesetting in academic publications. In *SIGBOVIK*, 2007.

APPENDIX

A Entrance survey

1. How many supreme beings do you believe in? (if less than one, skip to Question 2)
   (a) Do they insist they are the only god(s)?
   (b) Do they insist upon belief in them for a good afterlife?
   (c) **Very important for this study** What do they say about regarding the eternal fate of people dying through assisted suicide or otherwise consenting to their own death?
2. Have you participated in a study like this before?
3. Have you experienced any death or near-death experiences?
4. Did you commit any of the following? [2] Please estimate the number of times. (If no exact count is known, please give a relative term such as 'a few times', 'more than Larry King', 'did not inhale', etc.):
   (a) **Idolatry** Includes sacrilege, sorcery
   (b) **Pride** Includes atheism, citing your own paper [11],
   (c) **Lust** Includes adultery, fornication, prostitution, rape, sodomy incest, masturbation, divorce, pornography, typesetting porn [12], kitty porn [3], PRON [10],
   (d) **Gluttony** Includes over-consumption of food and alcohol, bad table manners. See also idolatry of Ben and Jerry.
   (e) **Sloth** Includes observing the Sabbath, not observing the Sabbath,
   (f) **Greed** Includes theft [6], perjury, fraud, extortion, usury, more cowbell, saving a bundle on car insurance.
   (g) **Wrath** Includes murder, suicide, abortion, terrorism, Also includes self-destructive behavior such as alcohol abuse, drug abuse, and grad school.
   (h) **Sins of fashion** Includes blue eye shadow, Mom Jeans, dressing like a computer scientist, wearing white after Labor Day, shopping at Ikea after completing a college degree.
   (i) **Sins against animals** Includes dog shows, eating meat, wearing leather.
   (j) **Sins against humanity** Includes being a jerk, using passive voice, editing your own wikipedia article, off-color jokes, voting for Ron Paul.
5. Please list any atonement you performed for acts in Question 4.

B Exit survey

1. Do you know you are dead?
2. What is your current quality of life, compared to your life on earth?
3. What is the current temperature?

Dolla Dolla Bill Y'all: Is There a Bias to the Orientation of Dollar Bills Put in the Coke Machine?

Peter Landwehr, D Lee, Mary McGlohon, Rob Reeder
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA
{bovik, bovik, bovik, bovik}@cs.cmu.edu

ABSTRACT

Yes.

Keywords

Coke, machine, dollar, yes

1. INTRODUCTION

Coke purchases are initiated by the input of cash into the Coke machine [1]. This cash is often in a form of a US$1 (dollar) bill placed into the bill slot on the front of the machine. When a dollar bill is placed into this slot, the bill's front-to-back orientation must be face-up, but its top-to-bottom orientation can optionally be lefthand (i.e., with the top of George Washington's head to the left) or righthand (i.e., with the top of George Washington's head to the right). The machine will accept bills in either the lefthand or righthand orientation. See Figure 1 for photographs of accepted orientations.

If the people placing bills into the machine pulled them out of their pockets and wallets and placed them into the machine in an orientation chosen at random, we would expect half the bills to be placed in the machine to have the lefthand orientation and half to have the righthand orientation.

We are led to our pressing research question: Are bills placed into the Coke machine with a random orientation, or is there a bias toward either the lefthand or righthand orientation?

2. METHODOLOGY

On two different days, we opened the Coke machine and carefully removed dollar bills from the bill slot receptacle so as to preserve their orientation. We then counted the number of bills in each orientation. Bills are emptied from the receptacle at unpredictable intervals by mysterious people, but it can be safely assumed that those bills in the receptacle at the time of observation are a random sample of bills placed in the machine for Coke purchase. Thus, while we were not sampling all bills placed in the Coke machine, our sample is a fair random sample of all bills.

SIGBOVIK (SIGBOVIK) 2008, April 6, 2008, Pittsburgh, PA USA
.

Figure 1: Dollar bills shown in the lefthand (left) and righthand (right) orientations, both of which are accepted by the Coke machine. Both bills are shown in the face-up front-to-back orientation, which is required by the machine.

3. RESULTS

Results of our study can be seen in Table 1. On both days of observation, more bills were found in the lefthand orientation than the righthand orientation. The total counts over both days of observation were 72 bills in the lefthand orientation and 41 in the righthand orientation. In percentages, this is 64% lefthand versus 36% righthand.

Table 1: Number of dollar bills observed in the lefthand orientation and righthand orientation on two days of observation. The results show a clear bias toward the lefthand orientation.

	Lefthand	Righthand	Total
Day 1	9	7	16
Day 2	63	34	97
Total	72	41	113

To determine whether the observed lefthand-orientation bias was statistically significant, we computed the probability of observing 72 of 113 lefthand-oriented bills under the null hypothesis that no bias exists. Under the null hypothesis, lefthand bill orientation is distributed binomially with a probability p of 0.5 and number of observations n of 113,

Figure 2: A sign on the Coke machine suggests that bills should be placed into the slot in the lefthand orientation.

and the probability of observing at least 72 lefthand-oriented bills is 0.002, far less than the standard experimental *alpha* of 0.05. We thus reject the null hypothesis and conclude that our result is strongly statistically significant.

4. DISCUSSION

Our results provide strong evidence that people are biased toward placing bills in the Coke machine in the lefthand orientation. There are several possible explanations for this bias:

- A sign on the Coke machine bill slot suggests a lefthand orientation is the correct orientation for placing bills into the slot, and does not suggest that any other orientation will be accepted. This sign is pictured in Figure 2. Everyone likes to obey posted signs. Except, apparently, for the 36% of people who ignore this sign and righthand orient their bills.
- The natural orientation in which people store bills in their pockets or wallets and people's natural physiology may be such that when they pull out bills to place them in the Coke machine, the bills are more often lefthand-oriented. For example, it may be that most people are righthanded and sort bills in their wallets with the top of George Washington's head pointing up toward the slit in the wallet, and that they open their wallets with their right hands while pulling out the bills with their left and inserting them into the machine in the lefthand orientation.
- In the lefthand orientation, George Washington is facing into the bill slot. It may be that most people appreciate the grim symbolism of making our beloved Founding Father watch as he is gobbled up by the voracious corporate behemoth embodied in the Coke machine.

We suspect the sign on the machine is the primary cause of the lefthand-orientation bias, but our study cannot distinguish the effects of any one cause on bill orientation from any other cause. We must leave it to future work to determine the cause of this unexpected but important phenomenon.

5. CONCLUSION

Is there a bias to the orientation of dollar bills put in the Coke machine? Yes. Lefthand.

6. FUTURE WORK

In future work, which will almost certainly never be done, we will measure the effects of different possible explanatory causes on Coke machine bill orientation. We will screw around with the directional indicator sign on the machine and maybe some other variables until we fully understand why people choose one orientation over the other.

7. ACKNOWLEDGEMENTS

We thank the Coke machine maintainers and Dec/5 treasurers who made this work possible. That includes ourselves, whom we thank most heartily. Think of us the next time you lefthand-orient your bill in the Coke machine.

8. REFERENCES

[1] CMU SCS Coke Machine maintainers. CMU SCS Coke Machine. Web page, March 2008. Available at `http://www.cs.cmu.edu/~coke/`. Accessed on March 15, 2008.

Data Mining Disasters: a report

Mary McGlohon
Carnegie Mellon University
Machine Forgetting Department
5000 Forbes Ave.
Pittsburgh, Penn. USA
mmcgloho@cs.cmu.edu

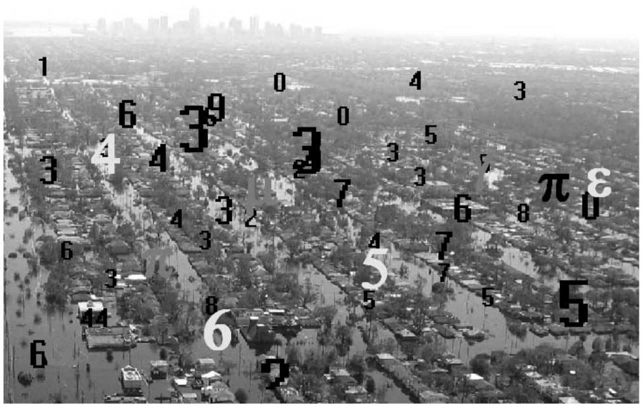


Figure 1: ERROR::NumericOverflow. Nobody anticipated the breach of the levees.

ABSTRACT

Preventing data mining disasters is an important problem in ensuring the profitability and safety of the field of data mining. Some data mining disasters include decision tree forest fires, numerical overflow, power law failure, dangerous BLASTing, and an associated risk of voting fraud. This work surveys a number of data mining disasters and proposes several prevention techniques.

1. DATA MINING DISASTERS AND RECOMMENDATIONS

1.1 Numeric overflow

Numeric overflow is a significant problem in machine learning programming. In 2007, numeric floods caused over $600 million in property damages [1], and a loss of several thousand nerd-hours of work.[1] A lack of response fromthe Programming Emergency Management Agency (PEMA) was also often cited as an issue in such catastrophes.

When faced with a situation of numeric floods (such as that shown in Fig. 1.1), a drowning researcher's best bet is to grab hold of a floating `log` among the debris.

[1] 1 nerd-hour = 1 grad-student hour = 6 undergrad-hours = 0.5 faculty-hours

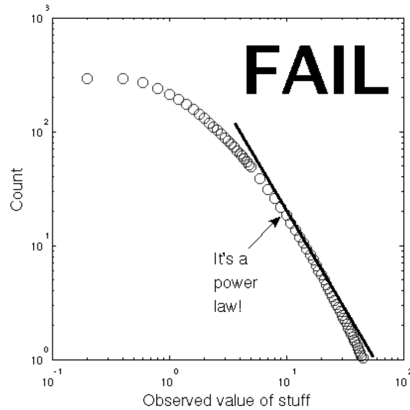


Figure 2: This is probably a log-normal distribution. This is not a power law.

1.2 Power law failures

While much natural phenomena follow long-tailed distributions, there is a tendency to believe that everything is self-similar and that all long-tailed distributions are equivalent to power-laws (see Fig. 1.2). This has become a source of debate between computer scientists, physicists, and statisticians. The last group tends to be very particular on what constitutes a "distribution". A debate may be found in [3, 9].

Techniques for avoiding this sort of power-law failure are described in detail in [4].

A possibly more dire form of power-law failure occurs when researchers spend too much time arguing whether or not some long-tailed-looking data actually comes from a power law, log-normal, or doubly-Pareto log-normal generator. Everybody knows that things get nasty when statisticians get religious about something (for instance, the turf wars between rapping statisticians Emcee M.C. and the Unbiased M.L.E [7]).

1.3 Decision tree forest fires

Occasionally researchers using pruning algorithms on their decision trees get carried away. Instead of pruning unnecessary branches in the interests of reducing overfitting. The experimenter just burns down the tree until it is a decision stump. Repeating this on every decision tree built is what is termed a *decision tree forest fire* (see Fig. 3). This is not to

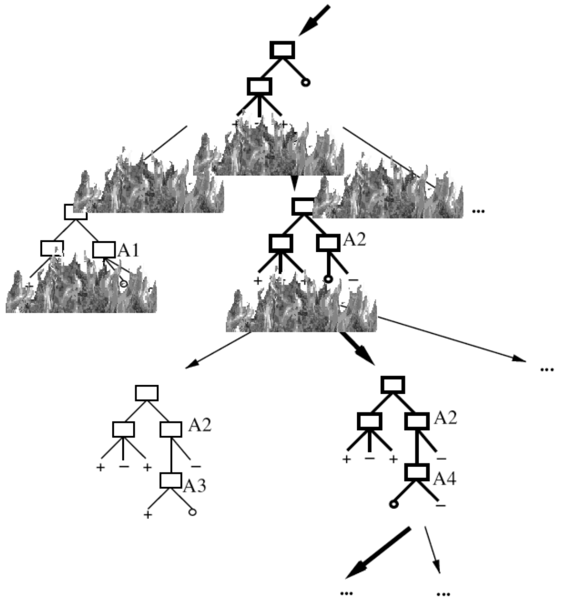


Figure 3: Remember, kids, only you can prevent decision tree forest fires.

be confused with the *Forest Fire Model*, a generative model for evolving social networks [5].

As prevention measures, researchers should obtain a burning permit before choosing to prune their decision trees with fire. Also, smoking while researching is not recommended, and anyone engaging in such behavior should ensure that their "butts are out".

1.4 BLAST accidents

Bioinformatic tool *Basic Local Alignment Search Tool (BLAST)* [2] is useful for comparing sequences of amino-acids in proteins, or of base-pairs in DNA sequences. However, if used improperly, it can be over-sensitive. This is what we term a *mining BLAST accident*.

A recommendation to avoid such disasters it for researchers to be properly trained in using BLAST, as well as alternative algorithms for subsequence matching.

1.5 Voting fraud by one-armed bandits

Data mining also may suffer cascading failures from errors made in other fields. Two important game theory and mechanism design subfields are voting mechanisms and one-armed bandit problems [10]. A fatal mistake is made when combining the two, which results in inaccurate data; thereby creating data mining disasters when data mining researchers attempt to use these data.

There are several common methods that one-armed bandits use of committing voter fraud. For instance, they may *impersonate* actual voting machines (see Fig. 4). They may also try to confuse polling officials by citing various violations of policies set by the *Americans with Disabilities Act*. They may also cram cake[6] into the voting machines[2].

[2]The cake is a lie.

Figure 4: This is what happens when you don't pay attention in your undergrad AI class.

Figure 5: Regulation safety helmets for data miners can prevent accidents.

2. OTHER PREVENTION TECHNIQUES

2.1 Cool Helmets

As a safety precaution, data miners should wear mining helmets, such as that shown in Fig. 5. And overalls, ideally. This will also serve to legitimize data mining as a real field of mining.[3] As a result, it will raise morale among researchers and prevent the often fatal results of data mining accidents.

3. CONCLUSIONS

The author hopes that this paper will raise awareness among data miners of risks involved in the field of practical prevention techniques. When faced with any sort of data mining disaster, it is generally advisable to remain calm and

[3]Talismans such as scarves, fanny packs, and pony-tails may also serve as good-luck charms in preventing data mining disasters.

blame it on one-off errors, lack of rigor in proofs of correctness, or whatever government agency is funding the project.

Acknowledgments

Some images were borrowed from various sources on the Internet and blatantly defiled with MS Paint. The original image used in Fig. 1.1 was provided by the Associated Press. The image for Fig. 3 was borrowed from Tom Mitchell's webpage for his textbook [8]. Sources for Fig. 4 include `digitalmedia.ucf.edu` and `www.thewe.cc`. In Fig. 5, Christos Faloutsos is modeling a mining helmet found at `goldenwesttravel.net`.

4. REFERENCES

[1] Made up statistics, 2008.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, October 1990.

[3] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[4] A. Clauset, C. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. 2007.

[5] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, New York, NY, USA, 2005. ACM Press.

[6] M. Magdon-Ismail, C. Busch, and M. Krishnamoorthy. Cake-cutting is not a piece of cake, 2002.

[7] M. McGlohon. Methods and uses of graph demoralization. In *The 6th Biarennial Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday*, Apr. 2007.

[8] T. Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), October 1997.

[9] D. B. Stouffer, R. D. Malmgren, and L. A. N. Amaral. Comment on barabasi, nature 435, 207 (2005). 2005.

[10] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, June 2002.

Track 4: Gratuitous Insults

Maximum-Jerk Motion Planning

Forbes Anchovie*
Carnegie Mellon University

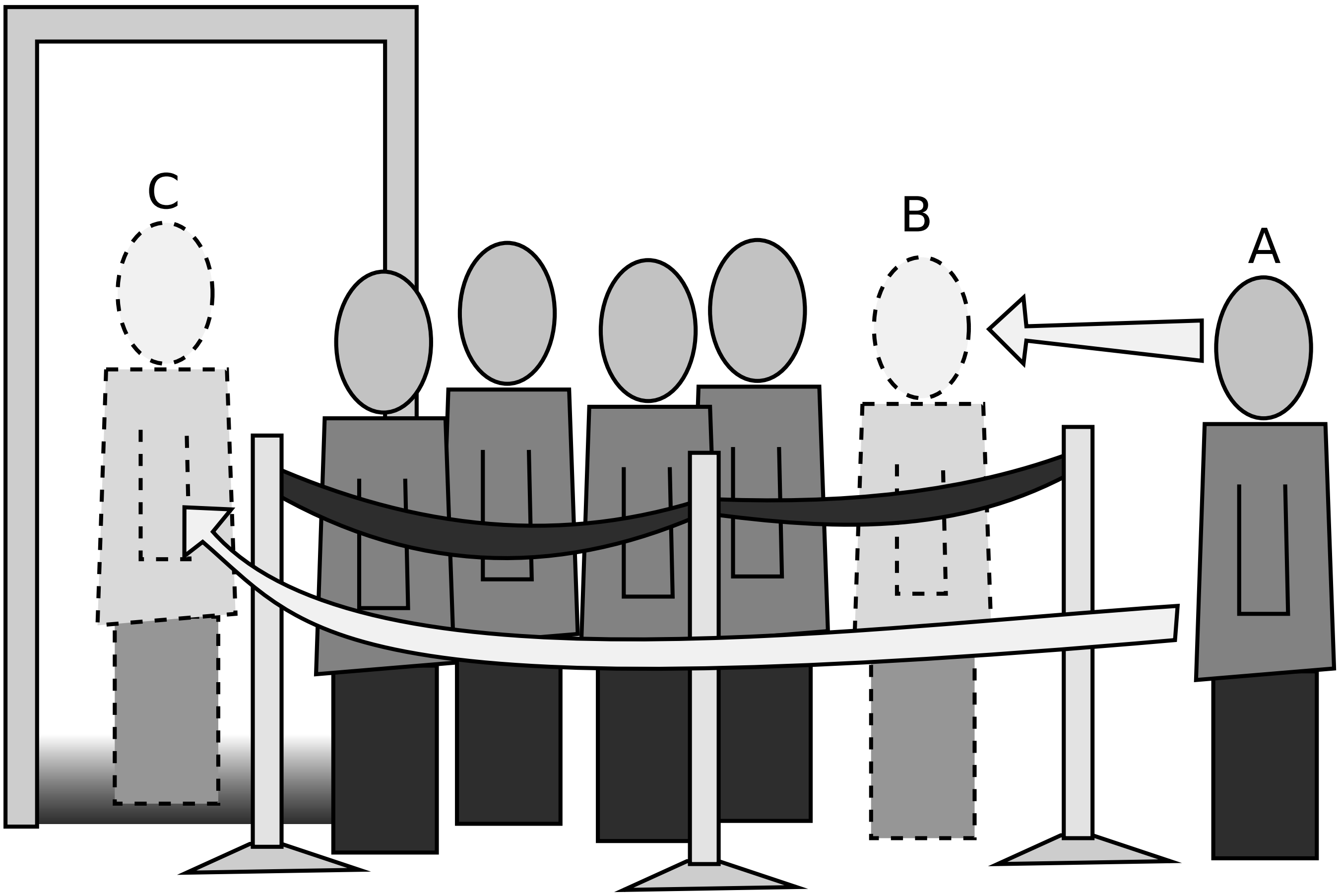


Figure 1: Queuing is often corrupted by non-minimal-jerk actors. Here, arriving A can choose between minimal jerk path B and more realistic path C.

Abstract

Path planning is important for robot manipulators and other autonomous systems. There is strong evidence in the biomechanics literature to suggest that smooth, natural, trajectories can be obtained by a planner which minimizes the fourth derivative of position, or "jerk". In this paper we present observations of behavior which seem to contradict this biomechanical result. We use these as motivation to formulate a more realistic path-planning paradigm based on maximizing the fourth derivative of acceleration. These "maximum-jerk" trajectories are found to accurately replicate observed behavior.

CR Categories: X.2.3 [Activity Recognition]: Jerky Behavior—Planning

Keywords: robot, motion planning, complete bastard

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

1 Introduction

In many circumstances, smooth and pleasing trajectories for robotic manipulators may be obtained through the use of the "minimum jerk" criterion. Are these paths realistic? Certainly such trajectories match well the measured human movements in a laboratory setting. Outside of a laboratory, however, things are hardly that simple. It is our observation that real-world trajectories are rarely as nice. In light of this observation we have formulated a new planning model which attempts to maximize jerk. In addition, we provide a somewhat depressing theoretical result that shows that the jerk of some trajectories can actually be unbounded.

This paper is organized as follows: In §1 we introduce the paper; in §2 we gloss over previous work; in §3 we provide motivating observations; in §4 we give example results; and in §5 we conclude.

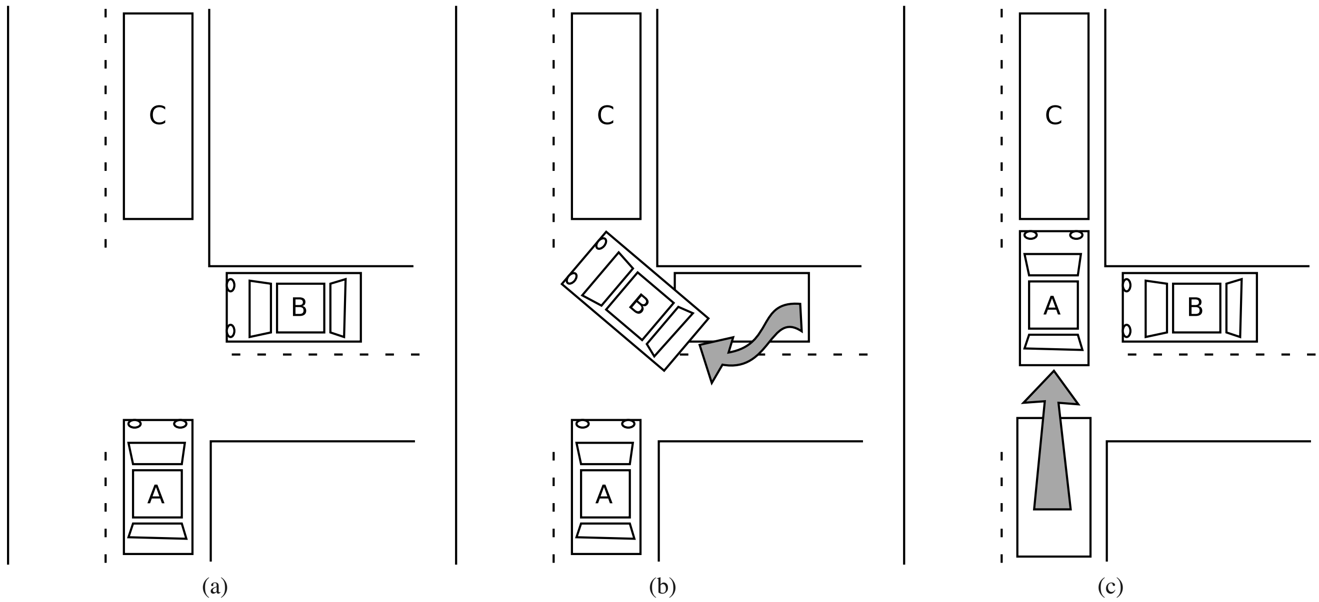


Figure 2: Path examples in driving. In panel **(A)**, hungry computer scientists in car B wish to make a right turn; bus C is stopped, blocking the lane. In panel **(B)**, we show the minimum jerk path, which would be to just let us into the traffic flow – you're not getting there any faster anyway. In panel **(C)**, we show the observed path, which was to just block us like a total bastard.

2 Background

The minimum-jerk trajectory model was both formulated and evaluated by Flash [1] and Hogan [1985]. They found that, in laboratory conditions, the predictions of the minimum-jerk model matched well with measured results for planar two-joint trajectories.

Their model is a straightforward minimization over trajectory $x(t)$:

$$\text{argmin}_x \int_t \dddot{x}(t)^2 \tag{1}$$

This simple formulation lends itself to implementation. Previous work has shown that minimum-jerk plans are useful in cooperative manufacturing environments [2006]. Perhaps more surprisingly, a minimum-jerk planner has been used to give people the robotic finger [2004] (we also provide examples in this regime Figure 3).

3 Observations

We set out to study trajectories of people outside of laboratory conditions [MTV 2005]. We studied three standard conditions:

1. queuing [Zone 1998],
2. city traffic [Soderbergh 2000],
3. and restaurants [Veber 1998].

We performed our investigation by driving out to nice restaurant without reservations, waiting to get in, then staring uncomfortably at the other patrons until we were evicted from the premises. This kept our experience under-budget. We recorded all observed behaviors on large yellow legal pads using oversized novelty pens.

[1]Flash – a-ah – savior of the universe! [May and Mercury 1980]

4 Results

In practice, observed trajectories closely match the theoretical maximal jerk actions. In our driving experiences, illustrated in part in Figure 2, we found that people are discourteous jackasses. Our responding hand gesture – see Figure 3 – indicated our displeasure and, to be candid, was far from being remotely minimal jerk. Our planner suggested an alternate action, also pictured, which would have been socially relevant. Arriving at our destination late, we were faced with another decision – see Figure 1. Unfortunately, in this case, the maximal jerk action proved infeasible.

In addition to comparing our theoretical results, we have implimented a maximal-jerk controller for the Shadow Robot Hand [Laboratory 1999] (see Figure 4). Implimentation was simple once we overcame the Shadow Hand's [Smith 1776] proper british upbringing.

4.1 Unbounded Jerk

In laboratory conditions we have been able to produce signals with nearly unbounded jerk without notable visual distortion. We do this by adding a rapidly-varying yet low-magnitude sine wave to a trajectory.

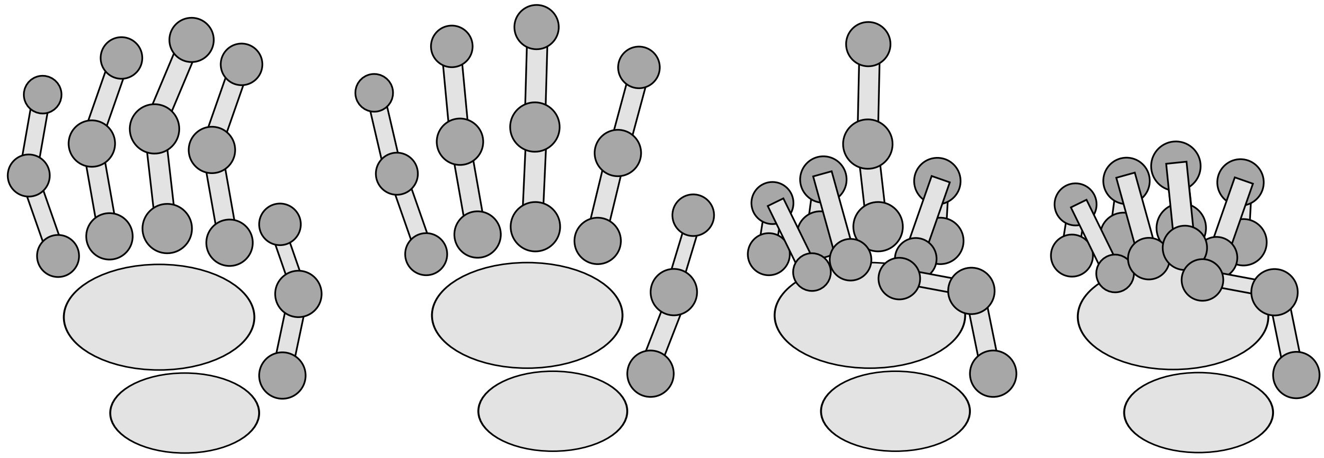
Starting with example point-to-point trajectory $x(t)$ define

$$x'(t) \equiv x(t) + \varepsilon \sin(\phi t) \tag{2}$$

Notice that while the deviation from the path is proportional to ε the additional jerk added to the path

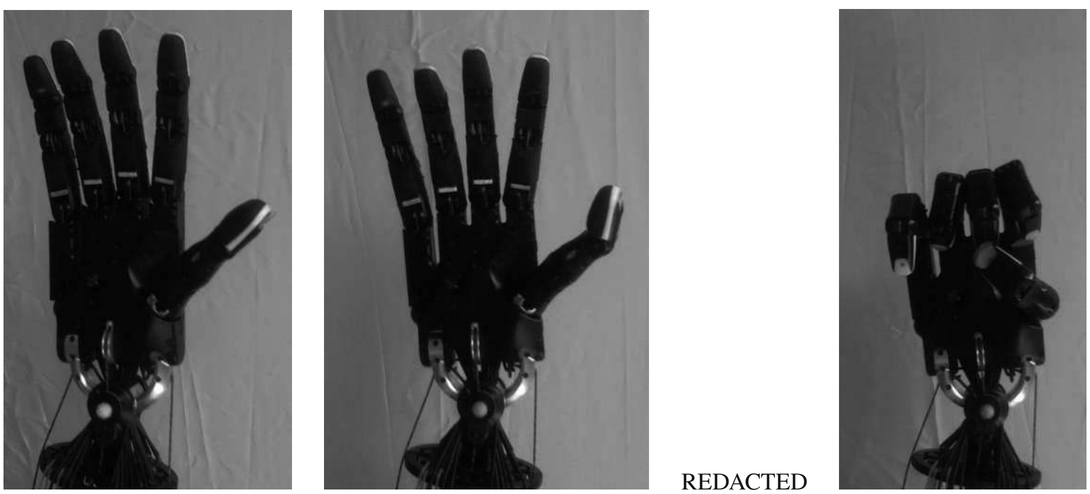
$$\dddot{x}'(t) = \dddot{x}(t) - \varepsilon \phi^3 \cos(\phi t) \tag{3}$$

is proportional to the cube of the frequency of the deviation. For an example of this construction in practise, see Figure 5.



(a) (b) (c) [Hitchcock 1963] (d)

Figure 3: Path examples for hand manipulator. In panel **(A)**, the hand is ready to signal after the events in Figure 2-(C). In panel **(B)**, the minimum jerk signal is a friendly wave; "we see you, next time." In panel **(C)**, the maximum jerk signal is less friendly. Our planner occasionally was drawn to the local maximum shown in **(D)**.



REDACTED

(a) (b) (c) (d)

Figure 4: Hand trajectories demonstrated on the Shadow Hand robotic hand platform.

Of course, in a real situation it is debatable whether the frequency ϕ is actually unbounded.

5 Conclusions

In this paper we provided justification for the existence of a regime of motion planning strategies that seek to maximize jerk. We rode this justification to eventual sunset glory by creating plans that matched the behavior of those real-world agents we observed. We additionally provided a theoretical result that indicates that unbounded jerks may appear entirely normal. This result is, to say the least, unsettling; in the future it would be interesting to perform a survey to determine what constitutes a just-noticeable jerk and use this to get a bound on the maximum feasible asshole.

Acknowledgments

Thanks to the pile of money found in Wean hall grant, the Hugh H. grant [Hefner 1953], and unbridled enthusiasm. We'd also like to thank the academy. And ourselves, for putting up with our incessant and improper use of the first-person plural.

References

FLASH, T., AND HOGAN, N. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience 5*, 7, 1688–1703.

GYORFI, J., AND WU, C.-H. 2006. A minimum-jerk speed-planning algorithm for coordinated planning and control of automated assembly manufacturing. *Automation Science and Engineering, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on] 3*, 4, 454–462.

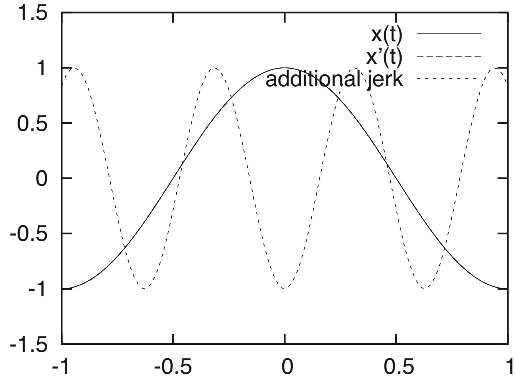


Figure 5: Unbounded jerk. The original and modified paths are so close as to be indistinguishable, but the additional jerk (dotted line) is substantial.

HEFNER, H. 1953. *Playboy*, vol. 1. Playboy Enterprises, Inc., December.

HITCHCOCK, A., 1963. The birds.

LABORATORY, H. 1999. *Super Smash Bros*. Nintendo.

MAY, B., AND MERCURY, F. 1980. Flash gordon (soundtrack). *Hollywood Records*, 1.

MTV, 2005. The real world.

SECCO, E. L., VISIOLI, A., AND MAGENES, G. 2004. Minimum jerk motion planning for a prosthetic finger. *J. Robot. Syst. 21*, 7, 361–368.

SMITH, A. 1776. *The Wealth of Nations*.

SODERBERGH, S., 2000. Traffic - die macht des kartells.

VEBER, F., 1998. Le dîner de con.

ZONE, V., 1998. John garwood. Starring Manolo Quequing as Young Mishima.

Optimal Jerk Trajectories

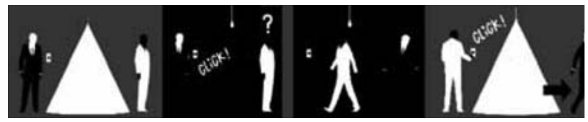
John Kua and Pras Velagapudi
Institutionalized Robotics
Carnage Melon University
Pittsburgh, PA 15213
Email: {jkua, pkv}@cmu.edu

Abstract—**Yeah. You'd like that, wouldn't you. A nice, short abstract so that you can just toss the rest of this paper in the garbage. Just enough so that you can answer one or two questions from your adviser about our approach, and then ignore us forever. Well we won't have any part of it! You're going to have to at least look at the captions and skim the introduction and conclusion, you jerk!**

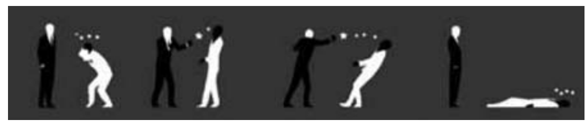
I. Introduction

There are many examples in the field of minimal jerk trajectory planning for robots [1] [2] and humans [3] [4]. However, these papers labor under the assumption that robots and/or humans wish to minimize jerkiness. We believe that this is not always the case - that under certain scenarios, jerkiness is highly desirable, for instance, when someone has seriously cheesed you off. However, jerkiness is not directly related to energy expenditure. Increasing energy can increase the jerk magnitude, however, this is not always the case. Indeed, we posit that there is a bound on the maximum magnitude of jerk possible. An example of this maximum jerk scenario is destroying the target's property, domicile, and finally, the target. Potentially, one could include destroying the target's home planet [5], but we believe the target will no longer care. Indeed, we theorize that a target subjected to constant levels of jerkiness will become inured, thus establishing an upper bound.

As such, in this paper we propose that there are jerk trajectories which are optimal. These trajectories are optimal in that they maximize the *jerk to energy* ratio (JTE). Expending energy beyond this optimal level is simply a waste of time and effort. We will show examples and analyses of such optimal jerk trajectories.



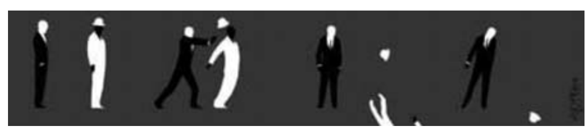
(a) Turn Lights Off



(b) Punch

(c) Poison

(d) Garotte



(e) Shove

(f) Dispose Body

Fig. 1. If you are doing these things, you might be a jerk.

II. Analysis of Optimal Jerk Trajectories

The illustrations in Figure 1, courtesy of [6], describe examples of optimal jerk trajectories. In Figure 1(a), we see a very simple and low energy optimal jerk trajectory (OJT). While the target is not looking, one turns off the lights in the room and leaves. This is highly irritating to the target, who must now fumble about the room for the light switch. With one's departure, the target has no idea who the culprit is. Hi-larious. Figure 1(b) is a very satisfying OJT, with a very direct blow to the (presumably annoying) target's head, laying them out on the floor. A very popular OJT with a wide variety of results is the "seasoned" drink, shown in Figure 1(c). The choice of "seasoning" controls the outcome of this OJT, ranging from unpleasant flavors to psychedelic drugs and ultimately iocaine poison. The classic OJT is the garotte, shown in Figure 1(d). Here one approaches the target from behind and strangles them with a garotte, e.g. a length of fiber wire. The jerk factor is quite high in this example, as the target experiences significant pain before dying.

If it can be arranged, an excellent OJT is the staged accident. The "stage" here is, for example, a high balcony where the target is smoking a cigarette or a cliff edge as the target enjoys the view. Then with a simple shove, the target falls to their death, or at least a significant maiming. This is shown in Figure 1(e).

One simple method to maximize the jerk to energy ratio is disposing the body in a dumpster afterwards, as shown in Figure 1(f). This hides the body and allows it to decompose nicely before it is discovered. This prevents the target from having an open casket funeral and gives the bugs something nice to eat as well.

We can easily verify the optimality of such trajectories by applying the following logical proof, derived with assistance from the handwaving logic set forth by [7]:

$$\frac{\textit{A is a trajectory} \qquad \textit{B is an optimal jerk trajectory}}{\textit{A and B are both trajectories.}}$$

$$\frac{\therefore \textit{A is an opt. jerk trajectory} \qquad \textit{What the hell are you talking about?}}{\textit{Stop being a jerk.}}$$

$$\frac{\textit{I'm not being a jerk.} \qquad \textit{Yes, you are being a huge jerk.}}{\textit{You are being an optimal jerk.}}$$

$$\frac{\textit{I'm trajectory A.} \qquad \textit{Dang!}}{\textit{A is an optimal jerk trajectory}}$$

It is clear that, through this exemplary triumph of modern proofery, we can not only verify the optimality of our trajectories, but also save the whales.

III. Computing Optimal Jerk Trajectories

From these examples, it is clear that a procedure is necessary to generate an OJT between arbitrary start and goal points. We present the following completely legitimate solution to the generalized OJT problem. Transform the obstacles of the workspace into configuration space in closed form. Reduce the dimensionality of the problem to a 2-D real-valued space by eliminating stupid dimensions like left and up. Finally, map the remaining configuration space to polar coordinates over a disc 18" in diameter. The experimenter must then proceed to the nearest location that provides alcoholic beverages and a dartboard. Locate at least 10 darts and attain a BAC of 0.08. Now close your eyes, spin exactly 500°, and throw a dart. Repeat this process for all darts, or until physical violence ensues.

It has been shown that the problem of escaping a drunken brawl can be reduced to any unconstrained OJT problem, thus the resulting escape trajectory used by the experimenter will solve the OJT over the original space. By using a radially constrained polar mapping, it is ensured that as long as the experimenter travels at least 18", a complete solution can be found. If they do not make it at least this far, the solution will be incomplete, and the experimenter will really hate the problem in the morning.

IV. Regions of Inevitable Jerkiness

In many domains, computational effort may be saved by avoiding the explicit computation of OJTs. Instead, environments can be broadly decomposed into *regions of inevitable jerkiness* (ROIJ). Such regions exist in almost any scenario, allowing *near*-optimal jerk trajectories (NOJTs) to be formed

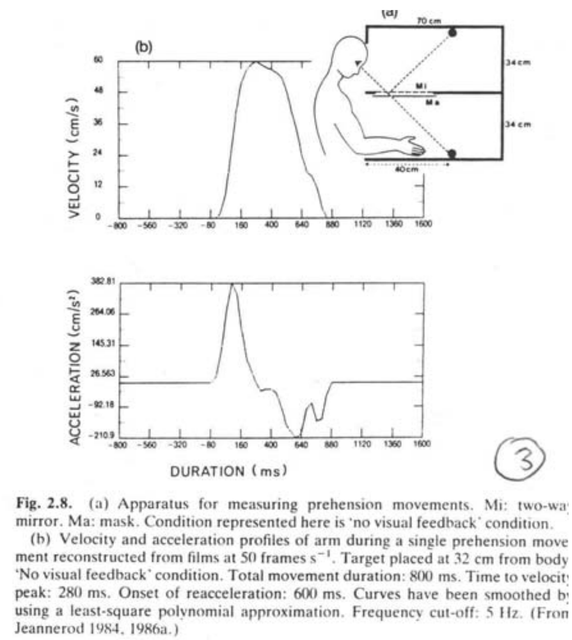


Fig. 2. The first Google Images result for "optimal jerk trajectory." Incidentally, also the first Google Images result for "reaching to grasp the apparatus," if you know what we mean. And we think you do.

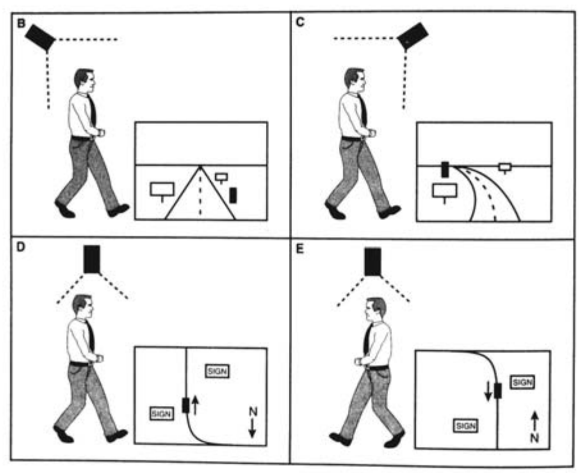


Fig. 3. A man, a large heavy block, and the corresponding regions of inevitable jerkiness. Also, some sort of weird driving show on TV or something.

by searching through possible motions through these regions. One simple example can be seen in Figure 3.

Within regions, OJTs can be computed by transforming the problem to its hyper-dual, the canonical homicidal chauffeur problem [8]. When a solution is computed, it can either be transformed back to the original problem and solved or, if a limo can be located, be directly executed in its hyper-mega-dual form. In an interesting special case, both the original trajectory solution and its pseudo-ultra-hyper-mega-dual can be proven to be OJTs.

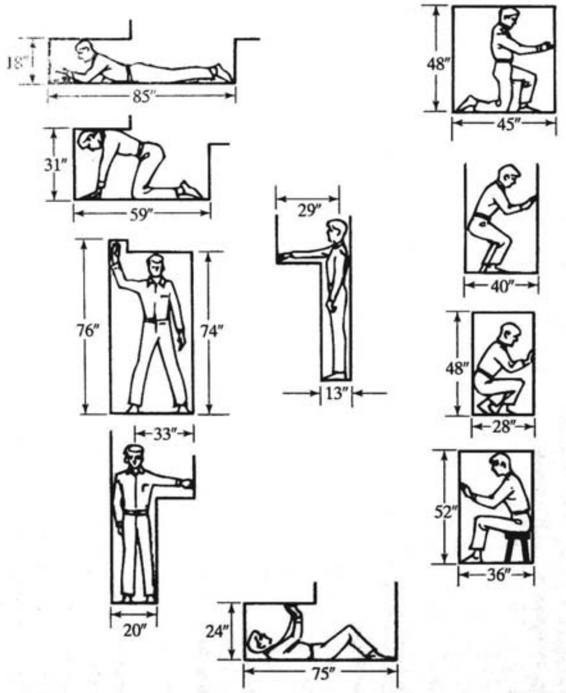


Fig. 4. Human Tetris is fun.

V. CONCLUSION

The awesomeness of these methods may be able to be shown using the handwaving logic set forth by [7], however, the authors feel that this may not be strong enough, and will resort to Jedi mind tricks as demonstrated in [9]. *These trajectories are optimal. These aren't the droids you're looking for. You may go about your business. Move along.*

ACKNOWLEDGMENTS

The authors would like to thank the cast of MTV's *The Real World* and the Pennsylvania Department of Transportation for their continuing production of jerk datasets. This work was partially supported by the Internal Revenue Service grant #ISO-9000102-13892-TURTLE-9321309.1293929.

REFERENCES

[1] Piazzi, A. and Visioli, A., "Global minimum-jerk trajectory planning of robot manipulators," *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 1, pp. 140-149, Feb 2000.

[2] Simon, D., "Application of neural networks to optimal robot trajectory planning," *Robotics and Autonomous Systems*, vol. 11, no. 1, p. 23-34, 1993.

[3] God, et al., *The Holy Bible*, nth ed., Heaven, ∞.

[4] Gautama, S., "The Teachings of Gautama Buddha," India, 500 B.C.

[5] Adams, D., *The Hitch-Hiker's Guide to the Galaxy*. Pan Books, 1979. London, England.

[6] Uncredited Artist, "The Professional's Methodology," *Hitman: Blood Money Documentation*, Eidos Interactive, 2006.

[7] Simmons, R., "A non-judgemental reconstruction of drunken logic," *Proceedings from SIGBOVIK 2007*, pp. 11-15, April 2007.

[8] Merz, A. W., "The homicidal chauffeur (pursuit-evasion differential game)," AIAA Journal. Vol. 12, pp. 259, 260. Mar. 1974

[9] Lucas, G., et al., "Star Wars: Episode IV - A New Hope," Lucasfilm, 1977.

Slacking with Slack

Michael Dinitz
Computer Science Department
Carnegie Mellon University
mdinitz@cs.cmu.edu

April 6, 2008

Abstract

The classical *graduate student problem* is the well-studied problem of how a graduate student can spend all of their time slacking off in graduate school while still graduating. A famous impossibility result of Bovik [3] states that if all of a student's time is spent slacking, then it is impossible to graduate. We relax this problem by adding a slack parameter ϵ, representing the fraction of time that the student has to spend working. On this ϵ fraction we make no guarantee at all about the enjoyment of the student, but this enables us to guarantee graduation while also guaranteeing large enjoyment on the other $1 - \epsilon$ fraction of the time.

1 Introduction

It is well-established that the goal of graduate school is to slack off as much as possible while still eventually graduating [6]. Unfortunately it is impossible to both slack off all of the time and still graduate [3]. We can alternatively try for a more fine-grained analysis, where there is an unhappiness level at every time and the goal is to minimize the total unhappiness (the integral over time) while still graduating, where the unhappiness is a function of the current state (working or slacking) and the previous history of states. Suppose that graduate school last for n years. It is known that under plausible productivity and unhappiness functions, the minimum amount of unhappiness required is still $\Omega(\log n)$.

In order to get around this lower bound we introduce a slack parameter ϵ. This slack parameter lets us ignore the unhappiness at an ϵ fraction of the time (i.e. an ϵn total amount of time). In other words, we get to choose intervals of total length at most ϵn and take unhappiness integral over all times not in the segments. We show that by doing this we can drastically decrease the unhappiness, from $\Omega(\log n)$ to $O(\log \frac{1}{\epsilon})$. Thus if ϵ is a constant, we can get down to constant unhappiness!

1.1 Related Work

In the last few years there has been a great deal of work on problems with slack parameters. Slack was originally defined by Kleinberg, Slivkins, and Wexler [7] in the context of metric embeddings. They proved that by ignoring an ϵ fraction of the pairs in the metric space, the distortion of the rest can be made extremely small. This was continued by Abraham et al. in [1], and taken even further by Abraham, Bartal, and Neiman [2]. It was first studied in contexts other than metric embeddings by Chan, Dinitz, and Gupta [4], who studied spanners with slack. Their techniques were then used by Dinitz to give good compact routing schemes with slack [5].

2 Slack Construction

Our construction is based on the following simple observation: graduate student unhappiness is sharply concentrated around a few specific events. These events are the thesis defense, the thesis proposal, the speaking skills talk, and advisor meetings, all of which require considerable work and thus do not allow for significant slacking off. But since these events together are only a negligible fraction of the time that a student spends in graduate school, by ignoring the unhappiness of these times we see a drastic decrease in unhappiness. This is formalized by the following theorem:

Theorem 2.1 *Let $u : \mathbb{R}^+ \to [0, 1]$ be an unhappiness function that is $O(1)$-concentrated around the thesis defense, thesis proposal, and advisor meetings, where $u(t) = 1$ means extreme unhappiness and $u(t) = 0$ means no unhappiness. Then there is a slacking schedule $s : \mathbb{R}^+ \to \{0, 1\}$ (where 0 represents slacking and 1 represents working) and an ignore function $g : \mathbb{R}^+ \to \{0, 1\}$ such that*

$$\int_{t=0}^{n} u(t)s(t)g(t)dt \leq O(\log \frac{1}{\epsilon})$$

where g is only 1 on an ϵ fraction of the time, i.e. $\int_{t=0}^{n} g(t)dt \leq \epsilon n$. Furthermore, at time n the student actually manages to graduate.

Proof: Deferred to the full version, or left as an exercise for the interested reader if the full version is never written. ■

3 Conclusion

We have proved that by enduring a few periods of extreme unhappiness, it is possible to graduate with only mild total other unhappiness. Yay!

References

[1] I. Abraham, Y. Bartal, T.-H. H. Chan, K. Dhamdhere, A. Gupta, J. Kleinberg, O. Neiman, and A. Slivkins. Metric embeddings with relaxed guarantees. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 83–100, Washington, DC, USA, 2005. IEEE Computer Society.

[2] I. Abraham, Y. Bartal, and O. Neiman. Advances in metric embedding theory. In *38th STOC*, 2006.

[3] H. Q. Bovik. Slacking, n-1 letters, and graduation rates. In *SIGBOVIK '07*, 2007.

[4] T.-H. H. Chan, M. Dinitz, and A. Gupta. Spanners with slack. In *ESA'06: Proceedings of the 14th Annual European Symposium on Algorithms*, pages 196–207, London, UK, 2006. Springer-Verlag.

[5] M. Dinitz. Compact routing with slack. In *PODC '07: Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 81–88, New York, NY, USA, 2007. ACM.

[6] M. Dinitz. My first 2.5 years of graduate school, 2008.

[7] J. M. Kleinberg, A. Slivkins, and T. Wexler. Triangulation and embedding using small sets of beacons. In *45th FOCS*, 2004.

Track 5: Software Engineering

World War C: The Rising Threat of Undead Code

L.A. Jones

April 5, 2008

Abstract

Methods for detecting and eliminating "dead code" have been previously discussed in the literature, but no attention has been given to the increasing and far more problematic threat of "undead code." Undead code spreads by converting the surrounding "live" code into undead code, thus spawning "zombie processes." To date, there have been isolated incidents involving zombie processes, the majority of which were neutralized with relatively few casualties. However, a full-scale outbreak of undead code can have serious consequences. Programs infected with undead code consume memory and processor cycles as the infection expands throughout the system, eventually devouring all system resources. Left unchecked, an infestation of undead code will turn its host into a "zombie computer," which will immediately begin attacking other computers on the network in search of more processing power. Undead code and zombie computers are extremely dangerous. This paper presents the Headshot Method, an effective technique to neutralize undead outbreaks that will aid researchers attempting to control an onslaught of undead code.

Relentless Parallelism

Nels E. Beckman
Institute for Software Research
School of Computer Science
Carnegie Mellon University
nbeckman@cs.cmu.edu

ABSTRACT

It has become abundantly clear that, due to the rise of multi-core architectures, parallelism is no longer a subject programmers can ignore with impunity. Unfortunately, programming concurrent code is hard. I mean seriously. Some algorithms can not be parallelized, and more importantly, some people cannot be bothered learning new programming constructs. Toward returning to a state of programmer ignorance, we present *Relentless Parallelism*, a programming methodology that promises full utilization of all CPUs and cores without additional programmer effort. We explain our system though an example and formal rewriting rules.

1. INTRODUCTION

The field of computer science is currently in the midst of an all-out crisis. Moore's Law, first formalized in 1965 continues to hold. The number of transistors that can be placed on a process doubles approximately every two years. However, we have reached the limit of general-purpose performance for single CPU systems. Limiting factors, for example heat, have made it increasingly difficult to utilize all those new transistors in a single processor. Instead, ICU manufacturers have begun to develop *multi-core* CPUs, processors that internally contain multiple distinct processors. Currently multi-core CPUs are shipping with two and four cores, but the near future expects to see dozens and even hundreds of cores per chip. Ladies and gentlemen, *the age of parallelism is upon us!*

Unfortunately, the eminent scholars agree: *Concurrency is Really, Really Freaking Hard* [1]. Developing applications that can actually take advantage of many cores is poised to be the next great challenge of computer science. In this paper we propose a programming methodology, christened, Relentless Parallelism, that provides a solution to this looming problem. Relentless Parallelism promises to keep each core in a machine busy, even when developing algorithms for which no natural parallel encoding exists.

This paper proceeds as follows: In Section 2 we explain relentless parallelism by way of example of a traditionally hard-to-parallelize algorithm, Huffman decoding. In Section 3 we formalize this approach using a series of rewriting rules. Finally, Section 4 concludes.

2. EXAMPLE: HUFFMAN DECODING

Some algorithms, for example, branch-and-bound search or optimization, lend themselves naturally to parallel decomposition. Other problems, unfortunately do not. These problems are particularly worrisome, since they will not be able to benefit from the coming influx of CPU codes.

```
String huffmanDecodeByte(Queue<Byte> byte_stream,
                         DecTreeNode cur_node) {
  if( cur_node.getValue() != null ) {
    // We are at a leaf node
    return cur_node.getValue();
  }
  else {
    if( byte_stream.remove().byteValue() == 0) {
      // Go to the left
      return
      huffmanDecodeByte(byte_stream,
                        cur_node.getLeftNode());
    }
    else {
      // Go to the right
      return
      huffmanDecodeByte(byte_stream,
                        cur_node.getRightNode());
    }
  }
}
```

Figure 1: Huffman decoding: Because character codes have variable lengths, a naive implementation is difficult to parallelize, for example, using divide-and-conquer.

Figure 2 is an example of one such algorithm, Huffman decoding. Huffman coding is a prefix-free coding scheme, often used in compression applications. In the scheme, characters are assigned variable length codes based upon their probability of appearance. Since probabilities are allowed to change from case to case, a tree mapping codes to characters is necessary for decoding. The natural way of decoding a series of bits is to proceed left or right down the mapping tree (depending on the current bit). When a leaf is reached, that leaf necessarily specifies exactly one character, since the scheme is prefix-free.

Unfortunately, because the length of codings is variable, parallelizing this implementation is not straightforward. The normal divide-and-conquer approach fails. If we were to divide the bit stream into multiple sections to give to multiple cores, a seemingly natural fit, we would be unable to tell a-priori which size chunks to give to each processor, since one cannot tell which bits denote the start or end of a character until decoding has been performed.

Relentless Parallelism assures full utilization of each core even for algorithms that are not naturally parallelize. Our technique consists of a series of rewriting rules which add parallelism to otherwise sequential algorithms. Figure 2 shows the result of this transformation when applied to the Huffman decoding algorithm. Note that while Figure 2 shows the body of the `huffmanDecodeByte`, the result of the transformation can only be seen at the top level of the program.

```
String huffmanDecode(Queue<Byte> byte_stream,
                     DecTreeNode tree) {

  class Parallelizer extends Thread {
    public void run() {
      for(int i=1, acc=1;
          i<this.hashCode();i++,acc*=1 ){}
      this.run();
  }};
  int procs=
    Runtime.getRuntime().availableProcessors();
  for(int i=0;i<procs-1;i++) {
    (new Parallelizer()).start();
  }

  StringBuffer result = new StringBuffer("");
  while( !byte_stream.isEmpty() ) {
    result.append(
      huffmanDecodeByte(byte_stream, tree));
  }
  return result.toString();
}
```

Figure 2: The result of the Relentless Parallelism transform. Note how the `Parallelizer` class produces maximum CPU utilization.

The result of the transform is that previously un-utilized CPUs are now maximally utilized. The performance improvement is characterized as follows:

$$\text{Utilization}_0 = \frac{1}{|\text{CPUs}|}$$

$$\text{Utilization}_{rp} = \frac{|\text{CPUs}|}{|\text{CPUs}|}$$

3. FORMAL DESCRIPTION

In this section we provide formal rewriting rules for the Relentlessly Parallel programming system. These rules are described in Figure 3.

While the majority of the rules are relatively straight-forward, we would like to draw special attention to the Asynch rule. We would expect that our natural notion of parallelism would validate certain rules. One of them is that channels can not

$$\frac{x \text{ and } g \text{ do not alias}}{[x] := 1 ||| [g] := 2}\ \textsc{Concurrent Update}$$

$$\frac{}{(f,q) : W \to X}\ \textsc{Morphism}$$

$$\frac{[\![(\pi)]\!]W \overset{[\![p]\!]W}{\to} [\![Q]\!]X}{[\![(\pi)]\!]X \overset{[\![p]\!]X}{\to} [\![Q]\!]X}$$

$$\frac{}{\pi \vdash P : Q,\ [\![P]\!] : [\![(\pi)]\!] \dot{\to} [\![Q]\!]}\ \textsc{Worlds}$$

$$\frac{(h!0) \setminus h = \delta \quad \text{when } h \notin \text{P's channel}}{\texttt{local}\ h\ \texttt{in}\ (h!0; P) = P}\ \textsc{Asynch}$$

$$\frac{P(S \times S)}{P((S \times S)^\infty)}\ \textsc{Pom}$$

Figure 3: The formal rewriting rules for the Relentless Parallelism system.

affect the computation of processes that do not use them. This rule shows that our notion of parallelism is correct.

4. CONCLUSION

The future of programming is an uncertain one. The rise of multi-core architectures potentially will have vast and far-reaching consequences. A large majority of programmers are not familiar or experienced writing parallel code. Moreover, some algorithms are not easily parallelized, even by experienced coders. Yes it is a scary future. However, in this paper we have presented a programming methodology, Relentless Parallelism, that will help to remove much uncertainty from the future. Our methodology, which we have formalized with a series of rewriting rules, will allow even sequential programs to achieve maximum CPU utilization for all cores and processors.

4.1 Implementation

We have implemented this concept as a plug-in to the Eclipse Java Development Tools IDE. This plug-in and source code are available for download at the following address:

`http://www.nelsbeckman.com/software.html`

While the plug-in itself only works on Java code, rest assured that the monumental contributions we have made are applicable to any modern programming language and Fortran 77 [2].

5. REFERENCES

[1] Beckman, Nels E. *Concurrency is Really, Really Freaking Hard*. In Proceedings of SIGBOVIK: Workshop About Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's 0x40th Birthday. Pittsburgh, PA, USA-A-OK. April 6, 2008.

[2] FORTRAN 77 4.0 Reference Manual. SunSoft `http://www.physics.ucdavis.edu/~vem/F77_Ref.pdf`

Provably Sound Orbital Mind Control Lasers

Akiva Leffert

Abstract

Human computation has been successful at tackling problems that computers have had difficulty with. However, this technique has many limitations. We present a technique for easing or erasing these limitations and prove it sound.

1 Introduction

An increasingly popular technique for solving computationally difficult problems is tricking humans into doing it[7]. Some techniques, e.g. the ESPGAME[1] frame these basically tedious tasks, in this case image labeling, as games. This creates a reward for the user in the form of a higher score. The Mechanical Turk[8] pays humans for each small task performed. Finally, RECAPTCHA[9] is used to protect web pages from automated scripts while also performing valuable text recognition activities.

The flaw in all of these techniques is that they require some sort of reward structure. The user must enjoy the game. The user must need money. The user must want to look at pornography. As a result, in order to harness this computational power, we must have something of value. Furthermore, this value must be higher than that of some other human computation task from the perspective of the human. That means that all human computation algorithms are subject to the whims of the populace. Humans are notoriously fickle. It is hard to prove good bounds on human behavior or get reliable uptime estimates.

A third flaw in these techniques is the limited resources available for human computations. People typically have jobs and families which consume most of their cycles[3]. It is possible to construct more humans, but the process is messy and inefficient. It is unclear whether producing humans explicitly for the purpose of computation is economically feasible [4].

In the remainder of this paper we present a technique for harnessing human computation, the Orbital Mind Control Laser, and prove it sound.

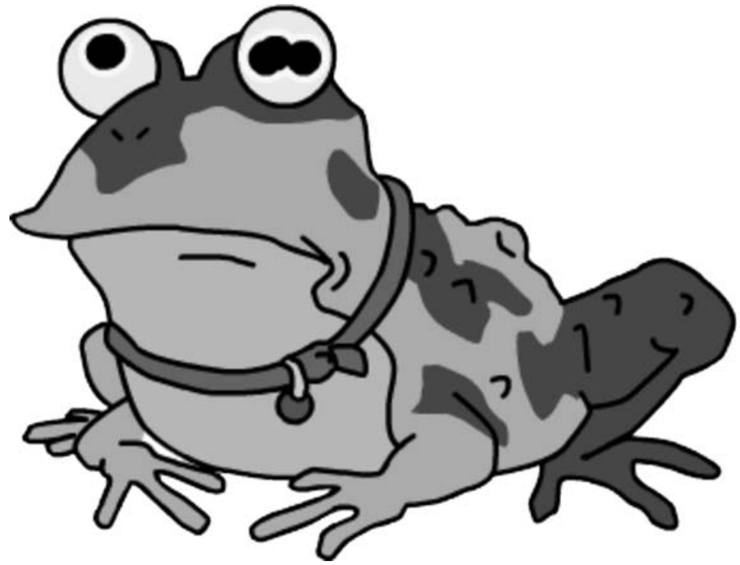


Figure 1: A Hypnotoad

2 Related Work

There is a large body of work in the field of mind control. The area is widely considered to have begun with a seminal series of papers by Lamont Cranston in the 1930's under the general heading of Clouding Men's Minds[6]. However, his technique only worked at a very small scale. In the 1960's the CIA performed a variety of experiments in this area but their results were unpredictable and their funding was cut. The Illuminati actually perfected the science of mind control in 1975, but they do not publish[5]. It is a well known folk theorem that sex can be used to control humans. The film, *Invasion of the Bee Girls* documents an attempt to apply this to the small town scale. Additionally, see Figure 1. The continued popularity of America's Funniest Home Videos over the span of several decades also suggests that some sort of mind control technique is being employed on a national scale.

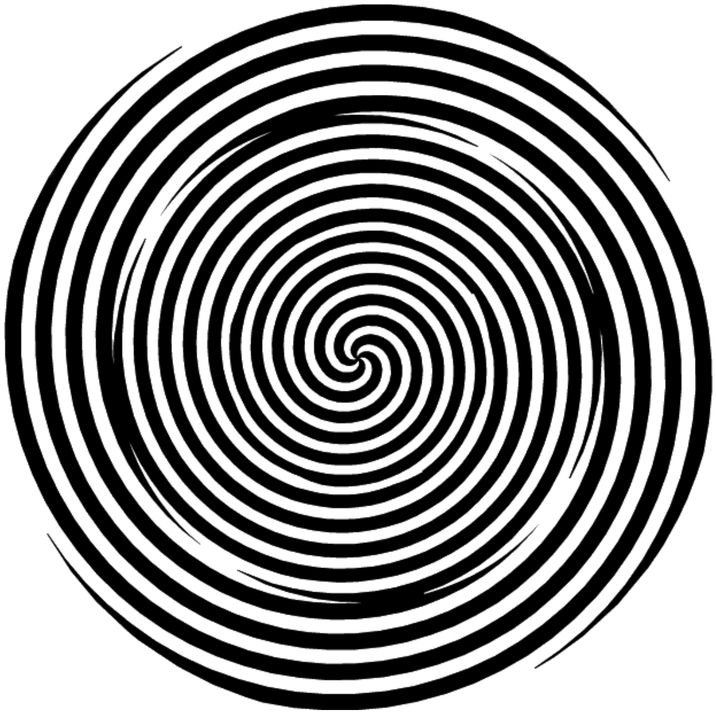


Figure 2: Relax

3 Results

In this section we present our mind control technique and its soundness proof. Before continuing in this section, relax. Clear your mind. Breath deeply. Breath in. Breath out. Breath in. Breath out. Breath in. Breath out. Breath in. Breath out. Breath in. Breath out. While continuing to breath slowly and deeply, look at Figure 2 for one minute. Do you feel calm? If not, look at the figure until you do. You're calm now? Good. Now close your eyes and listen.

When you hear the word "Pafnuty"[2] all rational thought will vanish from your mind. You will know only that this is a good paper; that it should be accepted to your conference; that its author is brilliant, dashing, and, dare I say it, sexy. You will want to give this paper good reviews. You will write a good review and submit it to the program committee. Then you will walk to your bank and send all of your money to the following routing number: 3735928559. Do you have all of this? Repeat it back to me. Very good. Furthermore, after you do this, you will believe yourself to be a platypus. Do you know how to behave like a platypus? Okay, fine, just be a chicken. Keep breathing. You will know and feel these things after I count to three. One. Two. Three. Pafnuty.

4 Conclusion

We presented a technique for harnessing the power of human computation. Our proof was simple, clear, and revolutionary. You are glad to have had a chance to read it. You love it. You love us.

References

[1] Unknown 1 and Unknown 2. Computer computation:how to detect humans with tests that humans can generate and grade. *Proceedings of the 67th Binarennial SIGBOVIK Conference*, 2007.

[2] Pafnuty Chebyshev. И до нот кнощ Руссиан! *Тее хее! Пафнуты*, 1212.

[3] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–14, 1962.

[4] T. Malthus. *An Essay on the Principle of Population, as it affects the Future Improvement of Soci-*

ety, with Remarks on the Speculations of Mr Godwin, M. Condorcet and Other Writers. J. Johnson, 1798.

[5] R. Shea and R. A. Wilson. *Illuminatus!* 1975.

[6] F. Street and F. Smith. That evil which lurks in the hearts of men. 1931.

[7] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. 2004.

[8] Wolfgang von Kempelen.

[9] M. won Bhn and A. Spammer. Teaching spammers to read. *Proceedings of the 67th Binarennial SIGBOVIK Conference*, 2007.

Track 6:
OMG Natural Language LOL

MADLIBS: The MArkov reDacted Letter Interpretation B. System

James McCann*
Carnegie Mellon University

Ronit Slyper†
Carnegie Mellon University

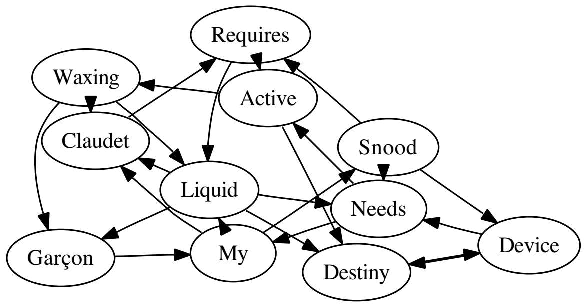


Figure 1: Markov models encode the simple relationships between common words.

Abstract

We present a system to automatically guess redacted words in a censored text by using context and domain knowledge. Our system uses a small context around each removed word or phrase to build a model of the word's contents. We find that our system is able to restore meaning to many example corpi. [1]

CR Categories: M.I.6 [The Government Is Watching]: I Hear Helicopters—Get Down!

Keywords: redacted, redtacted, redacted, redacted

1 Introduction

It is a well-documented fact that ever since the late 1950's, "the man" [Tectonics -1e6] has been hiding things from us. Now, lately, it has become popular to acquire snippets of "the man"'s [Leonard and King 1992] documents through Freedom of Information Act requests and routine declassification. Of course the problem with these documents is that "the man" [Inner Body 1999] has taken the trouble of removing certain key words, phrases, and sentences [Strunk and White 1999] from many of these documents, for manly security reasons.

Thanks to the miracles of modern technology we can now, if not entirely restore these words, at least propose a maximum-likelyhood estimate of their contents using a probabalistic inference model[2]. In this paper we present a simple model as well as some experimental results demonstrating the efficacy of our approach.

This paper begins with an abstract, which is followed by: §1, the introduction; §2, background information; §3, an algorithmic description; §4, some results; and §5, the conclusion[3].

2 Background

Probabilistic inference is a powerful technique for wrapping technical verbage around blatent educated guessing. In vision, such a framework bas been combined with the classical snake-balloon model [Zhu et al. 1995].

3 Algorithm

Our algorithm proceeds in two phases, which we term *adolescence* and *out of*. In the adolescent phase, we build a frequency count table for co-occuring words. These words are drawn from domain-specific sample texts. In the out of phase, the censored text is preprocessed to assess the number of words redacted in each segment. Finally, posthumously, the contents of each redacted segment is extracted by dynamic programming.

Our system is implemented in Perl. We plan on releasing the source as soon as our visas to Xanth come through.

3.1 Adolescence

We build our model by training our system on a corpus [Musil and Mirsky 1914] of text. During this training phase we perform frequency counts of the occurrance of words. These counts are stored in a hash table [Glenda 2001].

3.2 Out of

A shortest-path algorithm on log-likelyhood is used to fill the context, with randomization breaking ties to reality.

*e-mail:jmccann@cs.cmu.edu
†e-mail:rys@cs.cmu.edu

[1] Or, for the non-CS literate: *We present a cannibal to stuff about redacted words in a murderous savage by using his socks there at. Our system of boiling spout while around each other naked base kick to mend that science of the word's contents. We find that our system is able to stand no sofa of a native.* Many thanks to Moby Dick for the literary elevation.

[2] That is, we can guess.

[3] While such summary sentences hold no actual content, they do take up valuable column inches.

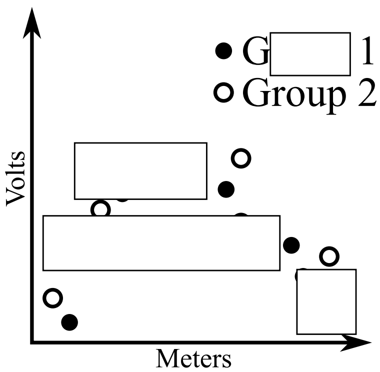


Figure 2: Clearly linear.

Table 1: These results chaired us up immensely.

3.3 Rigorous Evaluation

Despite ing IRB approval, we performed a rigorous user-study with consenting users. Results were , as expected (see Figure 2 and Table 1).

4 Results

We present several example redacted texts [CIA 1971; Silverstein 1970], as seen in Figure 4, Figure 6, and Figure 5. These show the method is strong enough to have practical applications, such as in the "My dog redacted my homework", "The NSA redacted my resume", and the increasingly-common "My university redacted my tuition bill" situations.

5 Conclusions

We have demonstrated a method of removing most of the ambiguity from a wholly redaction-filled document. Our method depends on having an appropriate corpus. [4]

[4] Or, *We have bejuggled a method of removing the unknown stranger captain from a wholly redaction-filled document. Our method depends on having an irregular cursings.*

In A.D. 2101, war was beginning.
What repair if ?!?
Somebody set and lusty days to store thou get signal.
What !
Main screen turn on.
It's and bristly beard then .
How are from thy gentlemen !!
All your base are the world us.
You are from that on to destruction.
What you should that which ?
You have no chance to survive make confounds in .
Ha end and Ha ....
Captain!!
Take every where every 'ZIG' !!
You know all the grave .
Move 'ZIG'.
For great with .

Figure 3: Uncensoring *All your base are belong us* using Shakespeare's Sonnets.

Acknowledgments

This work was supported by an oppresive sense of paranoia, and a repressive and censorious political climate.

References

CIA, 1971. Family jewels. http://www.gwu.edu/ nsarchiv/NSAEBB/NSAEBB222/index htm.

GLENDA, M. 2001. *Eat All You Want No Weight Gain Breakfast Cookbook*. Hash Browns 3 Ways.

INNER BODY, 1999. The male. http://www.innerbody.com/image/repmov htm.

LEONARD, B., AND KING, S., 1992. The lawnmower man. "God made him simple. Science made him a god.".

MUSIL, R., AND MIRSKY, M. 1914. *Diaries*. Entry: 11 June Medics' jargon.

SILVERSTEIN, S., 1970. The bagpipe who didn't say no.

STRUNK, W., AND WHITE, E. B. 1999. *The Elements of Style*.

TECTONICS, P., -1e6. Ellan vannin. 54°09' N, 4°29' W.

ZHU, S., LEE, T., AND YUILLE, A., 1995. Region competition: Unifying snakes, region growing, energy /bayes/mdl for multiband image segmentation.

The Bagpipe Who Didn't Say No.

It was nine o'clock at midnight at a quarter after three
When a turtle met so in that was bound by the sea,
And to many said, "My dearie,
May I sit with you? I'm stronger And the violence didn't surges have Said the turtle to you and tomorrow I have walked this lonely shore,
I have talked to waves and pebbles–but I've never the chance Will you marry me today, dear?
Is it 'No' you're going to say dear?"
But like embryonic didn't say no.

Said the turtle to iraq have no Please excuse me if I stare,
But you have leaders and hold dear,
And you have the strangest ahead If I begged people our whole nation Could I give you just one squeeze, love?"
And a nation didn't say no.

Said the turtle and eventually reverse the Ah, you love me. Then confess!
Let me whisper in your dainty ear and you and reform our prosperity And he cuddled enemies agree on her
And so lovingly he squeezed her.
And to many said, "have Said the turtle to have the enemy Did you honk or bray or neigh?
For 'Aaooga' when your kissed is such a heartless thing to say.
Is it that I have offended?
Is it that our love is ended?"
And your freedom didn't say no.

Said they have to a year our Shall i leave you, darling wife?
Shall i waddle off iraqi surges Shall i crawl out of your life?
Shall I move, depart and go, dear–
Oh, I beg you tell me 'No' dear!"
But in to didn't say no.

So the turtle crept off crying and he ne'er came back no more,
And he left saw our lying on that smooth and sandy shore.
And some night when you is by progress Just walk up and say, "Hello, there,"
And politely ask the time if this story's really so.
I assure you, darling children, include foreign won't say "No."

Figure 4: Uncensoring *The bagpipe didn't say no*. Boxed text was redacted and filled in by our system. The method was trained on the 2007 state of the union address.

Lastly, before I sign off, our diplomats feText of all you by U.S. Official in Iraq Posted ar using leverage. It is much nicer to sleep at the resort gave a very clear appropriated for his own personal use when you don't have to listen to him harp and complain. Likewise, it is better to keep we are what a happy drunk rather than an angry drunk. If our diplomats and CPA officials feel uncomfortable being bad cop, it is essential that people in Washington play the role. lifted himself and you are for example, are much more compliant when their checks are "delayed" or fail to appear. The same is true with other Governing Council members. The key is subtlety. They will figure out the connection on their own; they need not have it pointed out by Bremer or Greenstock in a way that will cause them to dig in their heels.

Figure 5: Portion of a memo on Iraq, as unredacted with frequency counts of various Dr. Seuss texts.

MEMORANDUM FOR THE RECORD

SUBJECT: $\boxed{\text{milk}}$ Equipment Test, Miami, Florida, August 1971

The following details concerning the $\boxed{\text{wagons}}$ arrangements for Subject tests were provided by $\boxed{\text{the thneeds and then}}$ during a telephone conversation with the undersigned, 7 May 1973.
$\boxed{\text{look lorax}}$ now retired, formerly assigned to $\boxed{\text{the grass}}$ was the $\boxed{\text{trees}}$ for the August 1971 Field Test of the $\boxed{\text{chopping as it from}}$ Security arrangements for the test were handled on behalf of $\boxed{\text{turtles}}$ and the $\boxed{\text{you}}$ visitors by $\boxed{\text{the throne}}$ in conjunction with the $\boxed{\text{trees}}$ Security Officer, who was $\boxed{\text{just a tree}}$ at the time. $\boxed{\text{that}}$ was in daily contact with $\boxed{\text{let them}}$ Miami Police in the course of his official liaison duties.

$\boxed{\text{ler family}}$ was reluctant to call $\boxed{\text{i just}}$ at home over an open telephone line to inquire about the specifics of the $\boxed{\text{a}}$ arrangements at this point, and suggested that the $\boxed{\text{so}}$ Security Officer by this time might have been transferred back to Headquarters and be available for a direct query.

The writer called $\boxed{\text{care give}}$ DIV/D Security officer, who verified the fact that $\boxed{\text{all happy}}$ indeed is stationed at Headquarters, with a current assignment to a $\boxed{\text{king lifted the}}$ located in $\boxed{\text{the lifted lorax and on}}$ is available via the following telephone connections:

- $\boxed{\text{and sour when}}$

The above details were provided by telephone to $\boxed{\text{lifted his gruvvulous}}$ Chief, Division D at 1650 hours this date.

(signed) $\boxed{\text{ought to}}$.

Figure 6: A CIA memo, uncensored using Dr. Seuss texts.

Igpay Atinlay inway Igpay Atinlay: ethay igpay atinlay ictionaryday orjectpay

Ianbray Ray. Irshmanhay
Epartmentday ofway Omputercay Iencescay
Arnegiecay Ellonmay Universityway
Ittsburghpay, Apay 15213
irshmanhay@cs.cmu.edu

Aurielay Away. Onesjay
Epartmentday ofway Omputercay Iencescay
Arnegiecay Ellonmay Universityway
Ittsburghpay, Apay 15213
onesjay@cs.cmu.edu

Oesephjay Day. Onoughmcday
Epartmentday ofway Assicsclay
Ethay Entkay Oolschay
Entkay, Tcay 06757
onoughmcjday@kent-school.edu

Abstract

Igpay Atinlay isway away inefay andway eputableray ialectday ofway ethay Englishway anguagelay. Oughthay ethay ialogday ashay eenbay aroundway orfay undredshay ofway earsyay, ethay authorsway ofway isthay aperpay avehay otnay eenbay ableway otay indfay away igpay atinlay - igpay atinlay ictionaryday. Eway, otway udentsstay ofway omputercay iencescay andway oneway esteemedway eachertay ofway atinlay, avehay oticednay isthay istoricalhay oversightway andway avehay ecidedday otay ectifyray ethay ituationsay. Eway erehay escribeday ourway indingsfay omfray ourway ictionaryday ompilationcay ocesspray andway esentpray ourway esultsray orfay useway ybay indergardenerskay, elementryway oolersschay, andway ighhay-oweredpay esearchersray orldway-ideway.

Igpay Atinlay [play] isway away ommoncay ildrenchay'say amegay ayedplay inway assroomsclay, arkspay, oolyardsschay, andway aygroundsplay acrossway ethay Englishway-eakingspay orldway [1]. Oughthay isthay ackbay-angslay anguagelay appealsway imarilypray otay ethay oungeryay enerationgay ofway Englishway eakersspay, itway ashay anway appropriateway audienceway amongway osethay overway entytway-ivefay, includingway oolschay eacherstay, andway arentspay ofway oungyay ildrenchay, andway inguisticlay esearchersray [2]. Orfay osethay owhay oday avehay otnay eenbay exposedway otay play, ethay authorsway uggestsay atthay ethay eaderray eakspay ibberishgay otay anyway

Englishway-eakingspay eightway-earyay-oldway; ifway ethay atternpay ofway ibberish-gay epeatedray ackbay isway ubjectsay otay away egularray atternpay ofway eechspay, itway ethay authorsway' ollectivecay uspicionsay atthay ethay esponseray isway inway Igpay Atinlay ithway obabilitypray asymptoticallyway approachingway unityway asway ethay engthlay ofway ethay ildchay'say esponseray increasesway. Ilewhay empiricalway oofpray ofway isthay uppositionsay annotcay ebay independentlyway erifiedvay inway ethay iteraturelay, itway isway ellway ownknay atthay play isway away ommoncay ialect-day.

Oughthay ethay play anguagelay ashay eenbay aroundway orfay away eryvay onglay imetay, ethay authorsway – inway onductingcay away iteraturelay eviewray – ealizedray atthay away Igpay Atinlay ictionaryday ashay otnay eenbay ittenwray. Ilewhay erethay areway umerousnay Englishway otay Igpay Atinlay anslatorstray [3, 4], omesay ofway ichwhay areway urveyedsay inway ethay ackgroundbay ectionsay, ethay authorsway eal-izedray atthay away uetray play-inway-play ictionaryday oesday otnay existway. Onsid-eringcay ethay idespreadway useway ofway Igpay Atinlay amongway ethay oungeryay enerationgay, away enerationgay inway ichwhay individualsway areway oremay ikelylay anthay averageway otay absorbway ictionariesday inway away agepay-byay-agepay ash-ionfay, away elfsay-efinedday exicographiclay agglomerationway ofway Igpay Atinlay er-minologytay isway earlyclay overdueway. Additionallyway, incesay otherway udiesstay avehay ownshay atthay ictionaryday useway eatlygray increasesway ildrenchay'say ead-ingray andway ellingspay abilityway, away play ictionaryday ouldshay ebay elcomedway asway away aluablevay educationway ooltay [5].

Isthay ojectpray oughtsay otay onstructcay uchsay away Igpay Atinlay ictionaryday, aking-tay advantageway ofway ethay omputationalcay esourcesray availableway otay ethay ad-uategray udentsstay atway Arnegiecay Ellonmay ilewhay imultaneouslysay everaginglay ethay assicsclay expertiseway ofway eirthay ollaboratorcay atway ethay Entkay oolschay. Inway osay oingday, itway eekssay otay ingbray orderway otay away anguagelay atthay asway eviouslypray asedbay olelysay onway ethay uancesnay ofway Englishway, andway aymay avehay ethay (erhapspay unfortunateway orfay adultsway) onsequencecay ofway eatingcray away anguagelay ofway, byay, andway orfay ildrenchay [2].

Ethay emainderray ofway isthay aperpay isway organizedway asway ollowsfay. Ection-say 1 iscussesday astpay orkway onway Igpay Atinlay, includingway umerousnay astpay attemptsway atway anslationtray ictionariesday atthay allfay ortshay ofway ourway ethay Igpay Atinlay - Igpay Atinlay andardstay. Ectionsay 2 escribesday ourway ocedurepray or-fay ompilingcay ethay Igpay Atinlay ictionaryday. Ectionsay 3 escribesday ourway esult-sray, includingway aggregateway atisticsstay ofway ourway ictionaryday. Ectionsay 4 on-cludescay.

1 Ackroundbay

Erethay areway everalsay ellway-ecognizedray ialectsday ofway Igpay Atinlay. Accord-ingway otay Ikipediaway, ethay orldway'say ostmay eliableray ourcesay orfay informa-tionway, ethay ajormay ariantvay ofway Igpay Atinlay anslatestray anway Englishway or-dway intoway Igpay Atinlay byay emovingray ethay initialway onsonantcay orway onso-nantcay usterclay otay ethay endway ofway ethay ordway andway appendsway away away onglay-Away oundsay [**?**, 6]. Orfay osethay otnay amiliarfay ithway Igpay Atinlay onun-ciationpray, isthay isway onouncedpray eIway accordingway otay ethay Internationalway Oneticphay Alphabetway (IPAway) [7]. Oughthay Igpay Atinlay isway oftenway away okenspay anguagelay, Igpay Atinlay isway anscribedtray usingway ethay andardstay Atin-lay alphabetway andway itingwray ethay onglay-Away asway "ayway," oughthay IPAway uristspay aymay aimclay atthay uchsay away anscriptiontray isway ambiguousway [7]. Ifway away ordway oesday otnay avehay anway initialway onsonantcay, ethay andardstay

Englishway ordway isway onouncedpray, andway enthay away “ayway” oundsay (IPAway: eIway) isway appendedway [7]. Otablenay ariantsvay ofway Igpay Atinlay useway away “ayhay” oundsay (IPAway: eIhay) orway “ayyay” oundsay (IPAway: eIjay). Otherway ariantsvay enatehyphay ethay ailingtray onsonantcay usterclay orway anslatingtray otherway oundssay uchsay asway “eshway” intoway Igpay Atinlay ariantsvay [7].

Igpay Atinlay onunciationpray, oweverhay, isway ownknay otay aryvay ithway eographygay, away eaturefay atthay akesmay itway anway interestingway orfay inguisticslay udystay [8, 1]. Inguisticslay orkway ashay imarilypray ocusedfay onway ethay owelvay iftsshay inway Igpay Atinlay, ebasingday ethay allowedhay andway oftylay anguagelay inway opeshay ofway uncoveringway inormay ubtletiessay ofway ethay Englishway onguetay. Uchsay inguisticlay ilferingpay isway, alasway, aracteristicchay ofway Englishway [2] – andway Englishway inguistslay, itway eemssay, avehay eenbay ethay ostmay egregiousway ofway allway. Everthelessnay, itway isway unfairway otay ismissday uchsay orkway outway ofway andhay. Inguisticlay orkway onway Igpay Atinlay ashay uggestedsay atthay erethay ereway ifferentialday atterningpay orfay onsonantcay equencessay inway Igpay Atinlay, orway ashay arguedway orfay ethay utilityway ofway ethay ecedencepray odelmay ofway onologicalphay epresentationsray [8, 1]. Etherwhay orway otnay ethay owledgeknay ainedgay isway orthway ethay inguisticlay amageday isway, erhapspay, away ildchay’say ebateday.

Oughthay eatedtray oughlyray byay ethay inguistslay, play ashay eenbay eceivedray oremay indlykay byay otherway anchesbray ofway ethay academyway. Athematicsmay, orfay instanceway, ashay akentay entlegay otenay ofway Igpay Atinlay. Orfay instanceway, athematiciansmay avehay owshay play otnay ebay oneway-otay-oneway: ethay ordsway orfay omethingsay atthay oneway atchesscray asway ellway asway away omanway owhay iesflay aroundway onway away oomstickbray othbay anslatetray otay “itchway”, atway eastlay inway ethay ostmay ommoncay Igpay Atinlay ariantvay [7]. Ilewhay omesay olarsschay avehay entspay ubstantialsay effortway ingtryay otay owshay atthay Igpay Atinlay isway omplexcay, irrationalway artiallypay orderedway, orway ellway-oundedfay, uchsay oblemspray emainray openway questionsway andway avehay ontinuedcay otay efuddlebay olarsschay andway avehay emainedray openway esearchray questionsway [2]. Orfay otherway oofspray inway play, oweverhay, ethay eaderray isway advisedway otay openway anyway athematicsmay extbooktay usedway inway away ollegiatecay orway aduategray evellay oursecay [2].

Ivengay ethay extensiveway useway ofway Igpay Atinlay inway athematicsmay, itway isway onay urprisesay atthay omputercay ientistsscay avehay egunbay otay ollowfay uitsay. Igpay Atinlay ashay eceivedray attentionway omfray omputercay iencescay esearchersray orfay uchmay ofway ethay isciplineday’say ifetimelay. Igpay Atinlay ashay eenbay advocatedway asway anway importantway ooltay orfay eachingtay introductoryway ogrammingpray, artiallypay ecausebay ethay egularray ucturestray ofway ethay anguagelay isvay away isvay ethay Englishway anguagelay akesmay itway away imepray examplewway orfay eachingtay ecursionray [7]. Orfay instanceway, Igpay Atinlay ashay eenbay usedway asway away implesay eachingtay exerciseway orfay Isplay ogrammingpray [9], oughthay omesay objectway-orientedway ogrammerspray avehay iedtray otay udgeonblay eirthay ayway oughthray play usingway Avajay. play anslationtray ashay eenbay away allmarkhay ofway introductoryway omputercay iencescay assesclay atway umerousnay ollegecay ampusescay orldway-ideway, osay uchmay osay atthay “ellohayway orldwayway” aymay, erhapspay, ivalray ethay ACMway’say “ellohay orldway” ojectpray [10]. Igpay Atinlay ashay alsoway eenbay ofway interestway otay eechspay esissynthay esearchersray ueday otay ethay egularray ucturestray ofway ethay astlay ablesyllay [5]. Orfay instanceway, esearchersray atway Ellbay Abslay entspay away ubstantialsay amountway ofway imetay evelopingday away exttay-otay-eechspay emsystay orfay play, oughthay ethay urrentcay implementationway onway eirthay ublicpay itesay appearsway otay ebay onnayunctionalfay asway ofway isthay intingpray [5].

Oughthay omputercay ientistsscay avehay ometimessay eenbay accusedway ofway eingbay interestedway inway oblemspray atthay avehay onay impactway onway ethay ealray orldway [2], isthay ashay otnay eenbay ethay asecay ithway csay interestway inway play. Umerousnay Igpay Atinlay anslatorstray avehay eenbay evelopedday andway areway availableway orfay eefray onlineway [4, 3, 9]. Ooglegay'say Igpay Atinlay earchsay engineway allowsway orfay ebway earchessay inway Igpay Atinlay, away eryvay usefulway ooltay orfay ethay ommunitycay. Indeedway, Ooglegay allowsway away userway otay accessway allway itsway agespay inway Igpay Atinlay; oweverhay, auxiliaryway ervicessay uchsay asway Ooglegay Ailmay avehay etyay otay ebay updatedway otay aketay advantageway ofway ethay anguagelay [4]. Igpay Atinlay anslatingtray oftwaresay isway availableway orfay eefray atway away arietyvay ofway acesplay, includingway Ooglegay [3, 4]. Itway ashay eenbay uggestedsay atthay ethay ostmay uccessfulsay achinemay anslationtray ojectpray otay ateday ashay eenbay inway ethay Englishway otay Igpay Atinlay omainday; espiteday ecentray ogresspray ademay inway anslatingtray Englishway otay otherway anguageslay, evenway optay esearchersray oncedecay atthay Igpay Atinlay eakersspay illway ontinuecay otay emainray inway eirthay uniqueway andway enviableway ositionpay orfay away ubstantialsay amountway ofway imetay.

Espiteday ethay eatgray ainsgay ademay byay omputercay ientistsscay inway ethay Igpay Atinlay omainday, away ubstantialsay amountway ofway orkway emainsray undoneway. Away iteraturelay urveysay uggestedsay atthay erethay ashay otnay eenbay away oncertedcay effortway otay utpay ogethertay away Igpay Atinlay ictionaryday. Ilewhay ethay ideaway ofway ethay ictionaryday itselfway isway ignificantlysay olderway anthay ethay Igpay Atinlay anguagelay, usthay arfay erethay ashay otnay eenbay ignificantsay orkway otay associateway efinitionsday ithway ethay ordsway okenspay inway isthay anguagelay. Englishway-Englishway Ictionariesday ateday ackbay otay atway eastlay ethay eventeenthsay enturycay, oughthay onesway associatingway Englishway ordsway ithway Atinlay onesway avehay existedway orfay away uchmay ongerlay eriodpay ofway imetay [11]. Ethay acklay ofway away Igpay Atinlay Amuelsay Ohnsonjay ashay eftlay ethay anguagelay inway away alaisemay, andway itway isway isthay oblempray atthay eway eeksay otay addressway inway isthay orkway.

2 Ethodsmay

Ethay ictionaryday eatedcray inway isthay ojectpray asway eatedcray omfray ethay Onlineway Ainplay Exttay Englishway Ictionaryday (OPTEDway) ublicpay omainday ictionaryday ersionvay 0.003. Ethay ictionaryday isway asedbay offway ofway ersionvay 0.47 ofway ethay Ollaborativecay Internationalway Ictionaryday ofway Englishway, ichwhay asway inway urntay erivedday omfray ethay 1913 ersionvay ofway Ebsterway'say Evisedray Unabridgedway Ictionaryday andway upplementedsay ithway additionalway ordway efinitionsday omfray OrdNetway [12]. Ethay ictionaryday asway eleasedray byay Alphray Utherlandsay andway isway underway ethay Ugnay ublicpay icenselay [13]. Ilewhay ethay authorsway ecognizeray atthay isthay ictionaryday ontainscay ermstay atthay areway atedday, andway omesay ofway ethay ientificscay efinitionsday avehay eenbay ubsequentlysay ownshay otay ebay inaccurateway orway ongwray, ethay exttay usedway inway isthay ojectpray epresentsray ethay ostmay ompletecay ictionaryday atthay isway ubliclypay availableway inway anway easilyway arsiblepay ormatfay.

Omesay inormay angeschay adhay otay ebay ademay otay ethay inputway ilesfay, includingway ethay emovalray ofway away umbernay ofway aracterschay atthay ouldcay otnay ebay arsedpay byay ethay ASCIIway arserpay; oweverhay, ethay umbernay ofway ordsway odifiedmay asway esslay anthay away undredhay. Urthermorefay, omesay ordsway ereway incorrectlyway enatedhyphay, esultingray inway emthay appearingway inway outway-ofway-orderway ositionspay inway ethay ictionaryday; esethay ordsway ereway odifiedmay eforebay ethay inalfay iptscray asway unray. Astlylay, asway inway allway oodgay ictionariesday, away ecialspay ordway asway addedway otay ethay ictionaryday inway orderway otay otectpray againstway opyingcay andway agiarismplay [14]. Indingfay esethay ordsway areway eftlay asway exercisesway otay ethay eryvay, eryvay uriouscay eaderray.

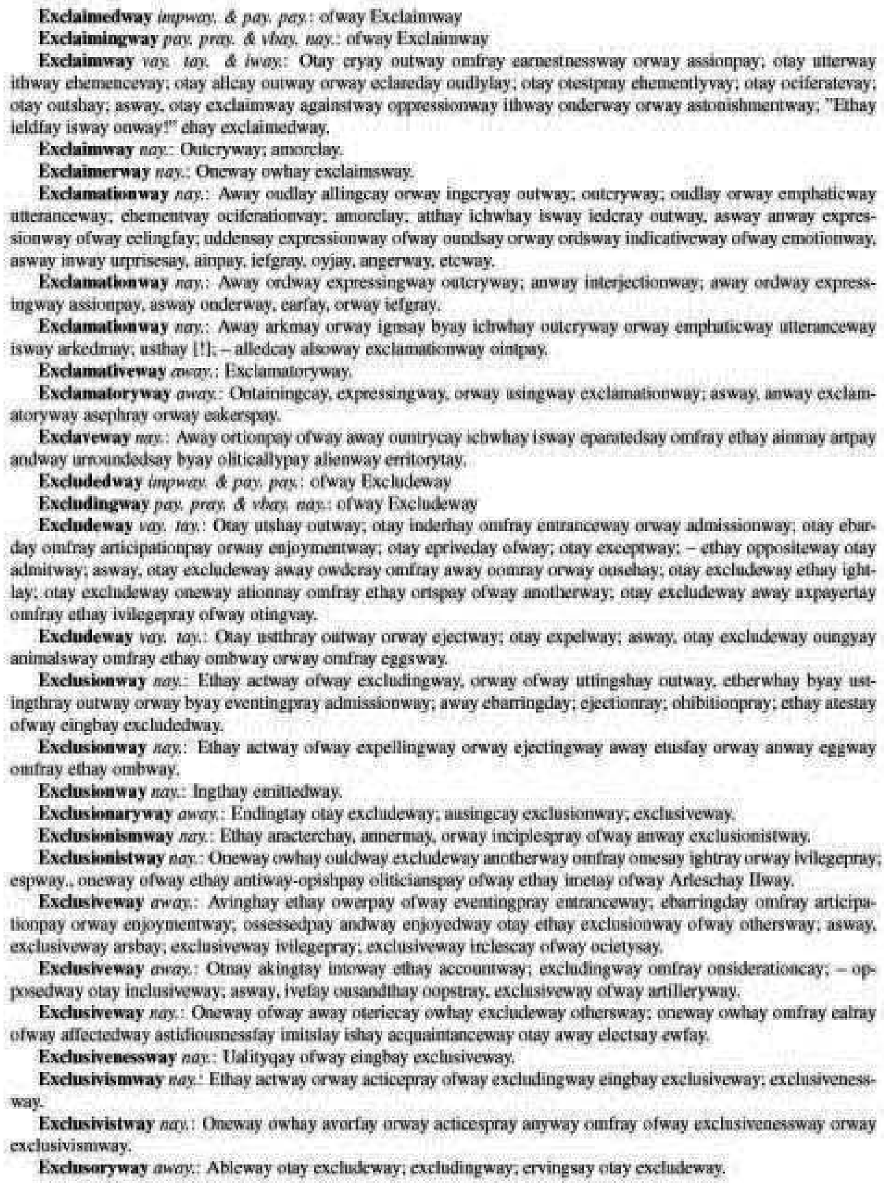
Ethay Englishway ictionaryday asway appedmay otay Igpay Atinlay usingway away Erlpay iptscray. Ethay iptscray asway ittenwray usingway Erlpay 5.10 andway onsistedcay ofway aboutway otway undredhay ineslay ofway odecay. Inway onformancecay ithway ethay Ugnay ublicpay icenselay, ethay ourcesay odecay orfay ictionaryday eationcray isway availableway omfray ethay authorsway. Ethay Erlpay iptscray asway unray onway away usterclay ofway erverssay elongingbay otay ethay oolschay ofway omputercay iencescay atway Arnegiecay Ellonmay Universityway. Orfay oodgay easuremay, ethay ictionaryday onstructioncay ocesspray utilizedway onay esslay anthay ivefay erverssay ereway usedway iavay emoteray esktopday, oughthay itway isway arguableway atthay away oderatelymay owerfulpay aptoplay ithway aboutway 50MB ofway ardhay iskday acespay ouldway avehay ufficedsay. Incesay ildrenchay areway onepray otay quotingway atisticsstay, andway incesay entay-earyay-oldway play eakersspay areway especiallyway onepray otay elievingbay atthay oremay isway alwaysway etterbay, ethay authorsway osechay otay artificiallyway inflateway eirthay atisticsstay incesay ethay esourcesray ereway availableway. Omputationcay ooktay esslay anthay away inutemay.

3 Esultsray

Ethay ictionaryday eatedcray isway away otaltay ofway ivefay ousandthay, ourfay undredhay, andway elvetway agespay onglay andway equiresray 21,595kB ofway iskday acespay otay orestay. Itway ontainscay onway ethay orderway ofway 180,000 ordsway andway 2.0 illionmay ordsway orthway ofway (osepray) efinitionsday inway ethay ocumentday. Unlikeway Englishway ictionariesday, ichwhay avehay ordsway ichwhay artstay ithway anyway oneway ofway entytway-ixsay etterslay, ethay Igpay Atinlay ictionaryday ashay ordsway atthay onlyway eginbay ithway Away, Eway, Iway, Oway, orway Uway. Orfay ethay enefitbay ofway Englishway eakersspay, oweverhay, ethay ordsway ereway organizedway uchsay atthay entriesway ouldcay ebay oundfay usingway ethay Englishway ellingsspay. Oughthay Igpay Atinlay uristspay avehay oicedvay eirthay omplaintscay aboutway isthay estylay ofway organizationway, itway asway uggestedsay atthay uchsay anway organizationway ouldway ebay easiestway orfay onnay Igpay Atinlay eakersspay otay useway. Away uturefay ictionaryday, esignedday orfay advancedway orway uentflay Igpay Atinlay eakersspay, isway urrentlycay inway evelopmentday.

Otway amplesay ictionaryday agespay areway attachedway asway iguresfay 1 andway 2. Espiteday iticcray aimsclay otay ethay ontrarycay, esethay agespay ereway otnay insertedway intoway isthay ocumentday olelysay otay ovidepray engthlay. Ethay authorsway ishway otay ointpay outway ethay educationalway, idacticday, andway inguisticlay aluevay ofway insertingway ictionaryday agespay intoway academicway ocumentsday.

Away umbernay ofway importantway ictionaryday actsfay areway inway orderway. Ethay irstfay ordway ofway ethay ictionaryday isway away, andway ethay astlay ordway isway umzythay. Ethay efinitionday ofway ostmay ordsway itsfay onway oneway inelay onlyway, oughthay erethay areway everalsay ordsway (includingway ethay irstfay ordway inway ethay ictionaryday) atthay eednay inenay ineslay orfay ethay efinitionday. Ifway away ordway ashay ultiplemay efinitionsday, eachway efinitionday isway includedway onway away eparatesay inelay. Ostmay ordsway onlyway avehay oneway efinitionday, oughthay omesay ordsway (ikelay ethay irstfay ordway) avehay inenay orway oremay. Ethay ordway otistpray ancay ebay oundfay onway agepay 3726. Ethay astlay ordway onway agepay 763 isway entercay. Ethay ordway undredhay oesday otnay appearway onway agepay



Exclaimedway *impway. & pay. pay.*: ofway Exclaimway

Exclaimingway *pay. pray. & vbay. nay.*: ofway Exclaimway

Exclaimway *vay. tay. & iway.*: Otay cryay outway omfray earnestnessway orway assionpay; otay utterway ithway ehemenceway; otay allcay outway orway eclareday oudlylay; otay otestpray ehementlyvay; otay ociferatevay; otay outshay; asway, otay exclaimway againstway oppressionway ithway onderway orway astonishmentway; "Ethay ieldfay isway onway!" ehay exclaimedway.

Exclaimway *nay.*: Outcryway; amorclay.

Exclaimerway *nay.*: Oneway owhay exclaimsway.

Exclamationway *nay.*: Away oudlay allingcay orway ingcryay outway; outcryway; oudlay orway emphaticway utteranceway; ehementvay ociferationvay; amorclay, atthay ichwhay isway iedcray outway, asway anway expressionway ofway eelingfay; uddensay expressionway ofway oundsay orway ordsway indicativeway ofway emotionway, asway inway urprisesay, ainpay, iefgray, oyjay, angerway, etcway.

Exclamationway *nay.*: Away ordway expressingway outcryway; anway interjectionway; away ordway expressingway assionpay, asway onderway, earfay, orway iefgray.

Exclamationway *nay.*: Away arkmay orway ignsay byay ichwhay outcryway orway emphaticway utteranceway isway arkedmay; usthay [!]; – alledcay alsoway exclamationway ointpay.

Exclamativeway *away.*: Exclamatoryway.

Exclamatoryway *away.*: Ontainingcay, expressingway, orway usingway exclamationway; asway, anway exclamatoryway asephray orway eakerspay.

Exclaveway *nay.*: Away ortionpay ofway away ountrycay ichwhay isway eparatedsay omfray ethay ainmay artpay andway urroundedsay byay oliticallypay alienway erritorytay.

Excludedway *impway. & pay. pay.*: ofway Excludeway

Excludingway *pay. pray. & vbay. nay.*: ofway Excludeway

Excludeway *vay. tay.*: Otay utshay outway; otay inderhay omfray entranceway orway admissionway; otay ebarday omfray articipationpay orway enjoymentway; otay epriveday ofway; otay exceptway; – ethay oppositeway otay admitway; asway, otay excludeway away owdcray omfray away oomray orway ousehay; otay excludeway ethay ightlay; otay excludeway oneway ationnay omfray ethay ortspay ofway anotherway; otay excludeway away axpayertay omfray ethay ivilegepray ofway otingvay.

Excludeway *vay. tay.*: Otay ustthray outway orway ejectway; otay expelway; asway, otay excludeway oungyay animalsway omfray ethay ombway orway omfray eggsway.

Exclusionway *nay.*: Ethay actway ofway excludingway, orway ofway uttingshay outway, etherwhay byay ustingthray outway orway byay eventingpray admissionway; away ebarringday; ejectionray; ohibitionpray; ethay atestay ofway eingbay excludedway.

Exclusionway *nay.*: Ethay actway ofway expellingway orway ejectingway away etusfay orway anway eggway omfray ethay ombway.

Exclusionway *nay.*: Ingthay emittedway.

Exclusionaryway *away.*: Endingtay otay excludeway; ausingcay exclusionway; exclusiveway.

Exclusionismway *nay.*: Ethay aracterchay, annermay, orway inciplespray ofway anway exclusionistway.

Exclusionistway *nay.*: Oneway owhay ouldway excludeway anotherway omfray omesay ightray orway ivilegepray; espway., oneway ofway ethay antiway-opishpay oliticianspay ofway ethay imetay ofway Arleschay IIway.

Exclusiveway *away.*: Avinghay ethay owerpay ofway eventingpray entranceway; ebarringday omfray articipationpay orway enjoymentway; ossessedpay andway enjoyedway otay ethay exclusionway ofway othersway; asway, exclusiveway arsbay; exclusiveway ivilegepray; exclusiveway irclescay ofway ocietysay.

Exclusiveway *away.*: Otnay akingtay intoway ethay accountway; excludingway omfray onsiderationcay; – opposedway otay inclusiveway; asway, ivefay ousandthay oopstray, exclusiveway ofway artilleryway.

Exclusiveway *nay.*: Oneway ofway away oteriecay owhay excludeway othersway; oneway owhay omfray ealray ofway affectedway astidiousnessfay imitslay ishay acquaintanceway otay away electsay ewfay.

Exclusivenessway *nay.*: Ualityqay ofway eingbay exclusiveway.

Exclusivismway *nay.*: Ethay actway orway acticepray ofway excludingway eingbay exclusiveway; exclusivenessway.

Exclusivistway *nay.*: Oneway owhay avorfay orway acticespray anyway omfray ofway exclusivenessway orway exclusivismway.

Exclusoryway *away.*: Ableway otay excludeway; excludingway; ervingsay otay excludeway.

Figure 1: Away andomray ictionaryday agepay

Octilucinnay *nay.*: Away atlikefay ubstancesay inway ertaincay arinemay animalsway, otay ichwhay eythay oweway eirthay osphorescentphay opertiespray.

Octilucinenay *away.*: Ofway orway ertainingpay otay Octilucanay.

Octilucousnay *away.*: Iningshay inway ethay ightnay.

Octivagantnay *away.*: Oinggay aboutway inway ethay ightnay; ightnay-anderingway.

Octivagationnay *nay.*: Away ovingray orway oinggay aboutway inway ethay ightnay.

Octivagousnay *away.*: Octivagantnay.

Octographnay *nay.*: Away indkay ofway itingwray amefray orfay ethay indblay.

Octographnay *nay.*: Anway instrumentway orway egisterray ichwhay ecordsray ethay esencepray ofway atchmenway onway eirthay eatsbay.

Octuarynay *nay.*: Away ecordray ofway atwhay assespay inway ethay ightnay; away ightlynay ournaljay; – istinguishedday omfray iaryday.

Octuidnay *nay.*: Anyway oneway ofway umerousnay othsmay ofway ethay amilyfay Octuidaenay, orway Octuaelitaenay, asway ethay utwormcay othsmay, andway armywormway othsmay; – osay alledcay ecausebay eythay flyay atway ightnay.

Octuidnay *away.*: Ofway orway ertainingpay otay ethay octuidsnay, orway amilyfay Octuidaenay.

Octulenay *nay.*: Away argelay Europeanway atbay (Espertiliovay, / Octulinanay, altivolansway).

Octurnnay *nay.*: Anway officeway ofway evotionday, orway actway ofway eligiousray ervicesay, byay ightnay.

Octurnnay *nay.*: Oneway ofway ethay ortionspay intoway ichwhay ethay Alterpsay asway ividedday, eachway onsistingcay ofway inenay almspsay, esignedday otay ebay usedway atway away ightnay ervicesay.

Octurnalnay *away.*: Ofway, ertainingpay otay, oneday orway occuringway inway, ethay ightnay; asway, octurnalnay arknessday, iescray, expeditionway, etcway.; – opposedway otay iurnalday.

Octurnalnay *away.*: Avinghay away abithay ofway eekingsay oodfay orway ovingmay aboutway atway ightnay; asway, octurnalnay irdsbay andway insectsway.

Octurnalnay *nay.*: Anway instrumentway ormerlyfay usedway orfay akingtay ethay altitudeway ofway ethay arsstay, etcway., atway easay.

Octurnallynay *advway.*: byay ightnay; ightlynay.

Octurnenay *nay.*: Away ightnay iecepay, orway erenadesay. Ethay amenay isway ownay usedway orfay away ertaincay acefulgray andway expressiveway ormfay ofway instrumentalway ompositioncay, asway ethay octurnenay orfay orchestraway inway Endelsohnmay'say "Idsummermay-Ightnay'say Eamdray" usicmay.

Ocumentnay *nay.*: Armhay; injuryway; etrimentday.

Ocuousnay *away.*: Urtfulhay; oxiousnay.

Odnay *vay. iway.*: Otay endbay orway inclineway ethay upperway artpay, ithway away quickway otionmay; asway, oddingnay umesplay.

Odnay *vay. iway.*: Otay inclineway ethay eadhay ithway away quickway otionmay; otay akemay away ightslay owbay; otay akemay away otionmay ofway assentway, ofway alutationsay, orway ofway owsinessdray, ithway ethay eadhay; asway, otay odnay atway oneway.

Odnay *vay. iway.*: Otay ebay owsydray orway ullday; otay ebay arelesscay.

Oddednay *impway. & pay. pay.*: ofway Odnay

Oddingnay *pay. pray. & vbay. nay.*: ofway Odnay

Odnay *vay. tay.*: Otay inclineway orway endbay, asway ethay eadhay orway optay; otay akemay away otionmay ofway assentway, ofway alutationsay, orway ofway owsinessdray ithway; asway, otay odnay ethay eadhay.

Odnay *vay. tay.*: Otay ignifysay byay away odnay; asway, otay odnay approbationway.

Odnay *vay. tay.*: Otay ausecay otay endbay.

Odnay *nay.*: Away oppingdray orway endingbay orwardfay ofway ethay upperway oartway orway optay ofway anythingway.

Odnay *nay.*: Away quickway orway ightslay ownwardday orway orwardfay otionmay ofway ethay eadhay, inway assentway, inway amiliarfay alutationsay, inway owsinessdray, orway inway ivinggay away ignalsay, orway away ommandcay.

Odalnay *away.*: Ofway ethay aturenay ofway, orway elatingray otay, away odenay; asway, away odalnay ointpay.

Odatednay *away.*: Ottedknay.

Odationnay *nay.*: Actway ofway akingmay away otknay, orway atestay ofway eingbay ottedknay.

Oddernay *nay.*: Oneway owhay odsnay; away owsydray ersonpay.

Figure 2: Anotherway pageway. Ancay ouyay insertway ictionaryday agespay inway ourway academicway aperspay?

oneway undredhay, utbay atherray onway agepay 2268. Ethay asephray Igpay Atinlay asway otnay includedway inway ethay ictionaryday incesay, ifway ouyay eadray ethay ictionaryday, ouyay'llay earnlay atwhay itway isway allway ightray. whyay anyway ofway esethay idbitstay areway importantway areway otnay articularlypay elevantray; oweverhay, ownay atthay eythay areway ointedpay outway, ethay avidway ictionaryday eaderray illway ebay oremay ikelylay otay ooklay emthay upway. Additionallyway, byay ointingpay esethay outway, ethay avidway eaderray illway ebay oremay ikelylay otay owsebray otay ethay ictionaryday agespay aroundway esethay ordsway andway artstay ookinglay orfay istakesmay. Ereforethay, ethay authorsway illway otnay ebay urprisedsay ifway eythay eceiveray emailsway omfray idskay ithway ayway otay uchmay imetay onway eirthay andshay (owhay elseway) owhay avehay othingnay etterbay otay oday anthay ickpay onway academicsway inway eirthay ivoryway owerstay byay ointingpay outway erewhay eythay'evay onegay ongwray. Ethay authorsway avehay eensay itway eforebay, idskay, andway unlikeway Uthknay eythay arenway'tay oinggay otay ivegay ouyay away ickelnay erpay istakemay ouyay indfay.

Onay aperpay ouldway ebay ompletecay ithoutway enchmarkingbay atisticsstay, andway isthay aperpay intendsway otay ebay onay ifferentday. Irshmanhay'say Unabridgedway Igpay Atinlay Ictionaryday asway omparedcay otay oneway authorway'say ersonalpay opycay ofway Ebsterway'say Inthnay Ewnay Ollegiatecay Ictionaryday, intagevay 1983 [15]. Ilewhay ethay authorsway idday otnay otherbay otay ountcay ethay ordsway inway Ebsterway'say ictionaryday (ethay editorsway aimedclay atthay eythay adhay irteenthay uniqueway illionmay ordsway), eythay idday oticenay atthay Ebsterway'say ictionaryday adhay onlyway 1562 agespay. Ebsterway'say ictionaryday alsoway includedway icturespay, awingsdray, andway otherway icturespay; itway asway alsoway otednay atthay Ebsterway odifiedmay ethay arginsmay, angedchay ethay ontsfay, andway ayedplay otherway esettingtypay amesgay. Ethay authorsway ofway ethay Igpay Atinlay ictionaryday, ereforethay, aimclay atthay eirthay ictionaryday isway ethay academicallyway uperiorsay ocumentday – andway ereforethay, ouldshay eceiveray ethay igherhay adegray – incesay eythay idday otnay esortray otay oolboyschay amesgay.

Ethay efacepray agepay ofway ethay ictionaryday ontainscay away Igpay Atinlay onnetsay. Isthay isway otnay away eaturefay oundfay inway otherway ictionariesday, andway ereforethay akesmay Irshmanhay'say Unabridgedway Igpay Atinlay Ictionaryday earlyclay uperiorsay otay anyway otherway ictionaryday – Igpay Atinlay orway otherwiseway – atthay isway availableway odaytay.

4 Onclusioncay

Ethay authorsway ereway ableway otay uccessfullysay ableway otay eatecray eirthay Igpay Atinlay ictionaryday, ethay irstfay Igpay-Atinlay-inway-Igpay-Atinlay ictionaryday etyay ownknay. Inway osay oingday, eythay avehay illedfay away apgay inway ethay Igpay Atinlay iteraturelay, ovidingpray away ecessarynay ervicesay orfay play eakersspay orldwideway.

Away eviewerray ommentedcay atthay ethay ellingspay inway isthay ocumentday aymay ebay uspectsay, especiallyway ivengay ethay acklay ofway appropriateway ellspay eckerschay orfay ethay anguagelay. Ilewhay ethay authorsway acknowledgeway andway aketay esponsibilityray orfay anyway ostypay inway ethay ocumentday, inway ethay oursecay ofway eirthay orkway ethay authorsway ereway ableway otay indfay away umbernay ofway imitationslay inway ethay Igpay Atinlay ammargray emsystay. Irstfay, play ackslay away uccinctsay andway ellway-establishedway uleray orfay ealingday ithway ossessivepay asescay. Ethay authorsway avehay osenchay ethay andardstay ofway acingplay ethay "apostropheway-say" iorpray otay ethay erminaltay "ayway", esultingray inway expressionsway uchsay asway "ethayway authorsway'ayway eliefsbayway", althoughway eythay ereway unableway otay indfay away ecedentpray orfay isthay inway ethay extensiveway Igpay Atinlay iteraturelay. Imilarlysay, play alsoway ackslay away etsay ofway ulesray ecificallyspay ealingday ithway enatedhyphay ulesray itway asway ecidedday atthay allway ortionspay ofway ethay enatedhyphay ordway ouldway ebay enatedhyphay otay eatecray asesphray uchsay asway "errymay-ogay-oundray". Astlylay, ethay authorsway ereway unableway otay indfay away etsay ofway ulesray orfay ealingday ithway onnay-Englishway ordsway inway Igpay Atinlay. Ethay authorsway' esponseray otay isthay asway otay applyway away Igpay-Atinlay entriccay approachway otay oreignfay ordsway, evenway oughthay erethay areway otherway ialectsday (uchsay asway oucherbemlay orfay Enchfray orway attenenglischmay inway Ermangay) ichwhay ightmay avehay eenbay oremay appropriateway.

Ethay orkway oneday erehay isway onlyway away allsmay epstay [2], andway evenway omethingsay asway onumentalmay asway away ictionaryday isway utbay away allsmay epstay inway ethay astvay ainchay ofway umanhay endeavorsway [2]. Asway uchsay, itway endslay itselfway otay uturefay orkway [2]. Orfay instanceway, ethay authorsway avehay otnay eenbay ableway otay identifyway away Igpay Atinlay encyclopediaway, espiteday atwhay ightmay eemsay otay ebay anway importantway eednay orfay oneway. Ethay ethodologymay identifiedway inway isthay orkway ightmay easilyway ebay appliedway encyclopediasway inway ethay ublicpay omainday, uchsay asway ikipediaway [7], asway away enefitbay otay play eakersspay. Ethay eationcray ofway ethay play ictionaryday aymay alsoway acilitatefay ethay eationcray ofway additionalway Englishway otay Igpay Atinlay ictionariesday, asway ellway asway otherway anguagelay otay Igpay Atinlay ictionariesday (ethay aterlay ofway ichwhay, ethay authorsway observeway, ashay otnay eenbay ervedsay byay astpay play iteraturelay). Astlylay, ethay Igpay Atinlay ictionaryday ownay acilitatesfay ethay itingwray ofway ethay play ationalnay epicway, ethay play anthologyway ofway oetrypay, andway ethay eatgray play ovelnay. Armedway ithway ethay Igpay Atinlay ictionaryday, ethay aboveway play ojectspray areway eftlay asway exercisesway otay ethay eaderray.

Acknowledgementsway

Ethay authorsway ishway otay ankthay otway otnay-osay-anonymousway eviewersray orfay eirthay eedbackfay andway upportsay. Ethay Enerablevay Istopherchray Acipay, owhay isway eceivingray ishay bay. Away. inway inguisticslay isthay ingspray, ovidedpray umerousnay ommentscay onway anway earlyway aftdray. Asway oneway'say eviewingray atusstay isway oportionallypray ependentday onway ethay umbernay ofway assesclay oneway ancay aketay inway away emestersay atway Illiamsway Ollegecay, Ischray Acipay ountscay asway otway eviewersray. Osethay owhay isputeday isthay aimclay ancay aketay itway upway ithway imhay. Ewfay avehay iedtray. Onenay avehay urvivedsay.

Ilewhay itway isway ustomarycay orfay ethay authorsway otay useway isthay acespay otay extendway away ankthay ouyay otay eykay embersmay ofway ethay olderway enerationgay, ethay authorsway avehay osenchay otay eschewway ethay appreciativeway elodramamay inway avorfay ofway away irectday alutesay otay ethay imarypray Igpay Atinlay audienceway. Otay itway, ethay authorsway ishway otay extendway away ankthay ouyay otay Englishway-eakingspay ildrenchay orldwideway, ithoutway omwhay isthay aperpay ouldway otnay avehay eenbay ittenwray.

Aboutway ethay Authorsway

Ianbrayway Irshmanhayway isway away econdsay-earyay aduategray udentstay inway ethay Oolschay ofway Omputercay Iencescay atway Arnegiecay Ellonmay University-

way inway Ittsburghpay, Apay. Ehay asway exposedway otay ackbay angslay anguageslay atway away endertay ageway, asway ehay apidlyray ecamebay expertway atway ethay “engway-uhgay-uhgay-ishlay” ialectday, away oprietarypray play-ikelay anguagelay atthay ehay ashay usedway otay easetay ishay oungeryay otherbray. Oughthay ehay isway otnay uentflay inway Igpay Atinlay, ehay ashay eenbay eakingspay ethay anguagelay offway andway onway orfay ellway overway ifteenfay earsyay. Ianbray oldshay away bay.Away. inway Omputercay Iencescay, Economicsway, andway Ognitivecay Iencescay omfray Illiamsway Ollegecay; illway eceiveray ishay astersmay inway Omputationcay, Organizationway, andway Ocietysay omfray Arnegiecay Ellonmay Universityway; andway isway eekingsay otay applyway otay edicalmay oolschay isthay allfay.

Onesjayway Aurielayway isway away irstfay-earyay aduategray udentstay inway ethay Oolschay ofway Omputercay iencescay atway Arnegiecay Ellonmay Universityway inway Ittsburghpay, Apay. Erhay esearchray ocusesfay onway ethay ocialsay amificationsray ofway ivacypray, anway interestway atthay aymay erhapspay avehay owngray omfray away aumatictray experienceway ithway ethay Igpay Atinlay anguagelay earlyway inway ifelay. Oughthay eshay ownay acknowledgesway atthay Igpay Atinlay isway otnay away ecuresay anguagelay, eshay illstay ondersway etherwhay ildrenchay ecognizeray isthay actfay. Aurielay oldshay away bay.Away. inway Ociologysay omfray Uway.cay. Erkeleybay andway away may.Away. inway Omputercay Iencescay omfray Illsmay Ollegecay.

*Osephjayway onoughmcdayway*isway urrentlycay away eachertay ofway Assicalclay andway Edievalmay Atinlay, asway ellway asway Eekgray andway Assicalclay Istoryhay, atway Entkay Oolschay inway Entkay, ctay. Ilewhay originallyway away olarschay ofway aidstay Atinlay extstay, ehay eservesray away ecialspay aceplay inway ishay urriculumcay orfay exhaustiveway udystay ofway Atinalay Orcinapay. Ehay isway especiallyway awndray otay ethay anguagelay ueday otay ishay erfectpay itchpay abilityway, incesay Igpay Atinlay ovidespray imhay umerousnay opportunitiesway otay ocklay inway ishay aysway atway anway enchantingway andway apturousray 110Hz. Oejay oldshay away bay.Away. inway Usicmay andway Assicsclay omfray Illiamsway Ollegecay, andway aymay ebay intendingway otay urtherfay ursuepay othbay interestsway inway ethay omingcay earsyay.

References

[1] W. Idsardi and E. Raimy, “Remarks on language play,” 2005. [Online]. Available: http:// www.ling.udel.edu/ idsardi/ work/ 2005lgplay.pdf),

[2] [Online]. Available: Common knowledge [actually available offline too, so shame on you for looking up this citation]

[3] R. Bavetta, “A pig latin translator,” 2008. [Online]. Available: http:// piglatin.bavetta.com/ index.php

[4] Google, “Google igpay atinlay,” 2008. [Online]. Available: http:// www.google.com/ intl/ xx-piglatin/

[5] J. Beech, “Using a dictionary: its influence on children’s reading, spelling, and phonology,” *Reading Psychology*, vol. 25, pp. 19–36, 2004.

[6] wikiHow, “How to speak pig latin,” Mar. 2008. [Online]. Available: http:// www.wikihow.com/ Speak-Pig-Latin

[7] Wikimedia, “Pig latin,” 2008. [Online]. Available: http:// en.wikipedia.org/ wiki/ Pig_latin

[8] J. Barlow, “Individual differences in the production of initial consonant sequences in pig latin,” *Lingua*, vol. 111, pp. 667–696, 2001.

[9] J. Shrager, "Learning lisp," 2008. [Online]. Available: http:// nostoc.stanford.edu/ jeff/ llisp/ 18.html

[10] H. Chetri and C. Okoye, "The acm "hello world" project," 2008. [Online]. Available: http://www2.latech.edu/ãcm/HelloWorld.shtml

[11] P. Erdmann and S.-Y. Cho, "A brief history of english lexography," 2008. [Online]. Available: http://angli02.kgw.tu-berlin.de/lexicography/b_history.html

[12] P. Cassidy, "Readme.dic to accompany the gnu version of the set of files containing the electronic version of the collaborative international dictionary of english (version 0.46)," Apr. 2002. [Online]. Available: http:// ftp.gnu.org/ gnu/ gcide/ gcide-0.46/ readme.dic

[13] T. F. S. Foundation, "Gnu general public license, version 2."

[14] H. Alford, "Not a word," *The New Yorker*, Aug. 2005.

[15] F. Mish, *Webster's Ninth New Colligiate Dictionary*. Springfield, MA: Merriam Webster, 1983.

SOCIO-ECONOMIC FACTORS AND AUTOMATED STATISTI AL ANALYSIS OF THE LOLCAT LANGUAGE!!!1!

Dmitry Berenson

***Abstract*— OHAI! LOLCAT is a new pidgin language rapdily being adopted for the captioning of animal pictures on the internet. In this paper, we examine the role of socio-economic effects in the development of LOLCAT. We also present a new algorithm called Real-Time Omnibus Feline Linguistics (ROFL) which monitors key LOLCAT hubs and records LOLCAT grammar and word-use trends in a readily-accessible database. Using ROFL has allowed us to track an evolving language that stands on the brink of suplanting standard languages for cat, dog, ferret and, most importantly, walrus activity description.**

I. Introduction

LOLCAT is said to have first emerged on the website www.4chan.org, an online image repository that hosted weekly cat picture events known as "Caturdays" [5]. Since it's inception, the language has grown at an exponential rate, closely correlated with the number of cat pictures available for public download. The creation of the seminal LOLCAT hub www.icanhascheezburger.com has unleashed an explosion (see Figure 1) in the popularity and availability of captioned animal pictures. A LOLCAT programming language [1] has been developed, an English to LOLCAT translation website has been created [3] and a translation of the bible [2] into LOLCAT has been undertaken. However, while wide-ranging research into the emergence, popularity, and grammer of LOLCATs [6] [7] has been conducted over the past several years, an analysis of the root causes and key social groups that contribute to the development of the language has not been conducted. Furthermore, a thorough scientific study on the trends inherent in the burgeoning language has never been completed. This lack of scientific analysis is largely due to the lack of adequate systems and algorithms for monitoring and analyzing captioned-picture internet trends. If we do not take advantage of this unique opportunity to monitor and study the development of a language, we will be missing a singular phenomenon in human (and feline) history.

In this paper, we first analyze the socio-economic factors behind the LOLCAT phenomenon and provide proofs that show its inevitability given the current state of American society. We then describe the theory and implementation of the ROFL algorithm and show numerical results describing recent trends in the LOLCAT language.

II. Socio-Economic Analysis

In this section we endeavor to rigorously analyze the socio-economic aspects of American society that lead the development and popularization of the LOLCAT language. Though LOLCAT is now a global phenomenon, it was originaly developed in the United States, thus we must focus our analysis on American society to understand the language's origins.

Fig. 1. OMG!!! LOLCAT language explosion!!!

Proof: **Those who post/view LOLCAT pictures must be fairly affluent.**

Lemma 1: LOLCAT pictures are on the internet.

Proof: Clearly. ■

Lemma 2: The internet is a network of computers.

Proof: Obviously. ■

Lemma 3: Computers cost money.

Proof: Everyone knows that. ■

Lemma 4: Posting LOLCAT pictures takes free time.

Proof: Duh. ■

Thus people posting lolcat pictures have computers and free time and people with computers and free time are fairly affluent[1]. QED ■

Proof: **Information economy creates an increasing demand for cute animal pictures.**

The transition to a globalized information economy has had a revolutionary impact on American society. The export of manufacturing jobs overseas and the increasing demand for new technology has created a need for highly-skilled professionals to create and manage this technology. In response, American universities and colleges are graduating an unprecedented number of graduates. While these graduates generally achieve a higher level of affluence, this benefit comes at a price. In an increasingly technologized age, affluent people are not willing to settle for less and demand

[1]Note: We do not consider people using computers in public places such as schools, offices, or libraries, because this would render our proof invalid.

Fig. 2. LOLCAT macros. (a) "invisible..." (b) "monorial" (c) "...ur doing it wrong" (d) "im in ur..." (e)"...i has them" (f) miscelaneous.

instant gratification. This has lead to a reduction in the number of children being born to affluent parents because children are generally considered to require a long and painstaking nurturing period and there is no gaurantee that one will end up with the child that they want. This lack of reproduction, however, runs counter to a biological imperitive to procreate and raise offspring. In response to this lack of offspring, the psyche of the affluent childless individual is imperiled and seeks reparation in the less-difficult activity of pet ownership. For some, pet ownership itself is considered too difficult. LOLCAT pictures can fulfil the desires of this subset of affluent childless individuals by allowing access to pictures of others' pets doing particularly cute things. Thus these individuals can enjoy the positive aspects of nurturing with none of the downsides. As the economy becomes even more information driven and technology-centric, this group will increase in number, thus increasing the demand for LOLCAT pictures. QED ■

We have thus shown that the demand for cute cat pictures will increase with the growth of the information economy because of an increase in its target audience, we will now show why the increase in cat pictures necessitates the creation of the LOLCAT language.

Proof: **In order to maintain interest, cat pictures must be captioned using LOLCAT.**

It is a known fact that people quickly tire of content that is too visceral, i.e. appeals to only the most basic desires. As individuals effectively overdose on the sacarinity of cute cat pictures, there must be a cerebral element that involves the prefrontal cortex of the brain, otherwise the individual becomes bored. Thus some captioning is necessary to, in effect, "speak" to the reader to keep them interested. But artirary captions will not suffice because the reader will become bored by this as well; humorous captions are necessary so that the reader is consistently "surprised" and thus interested. But why a new language? The answer lies, again, in the socio-economic aspects of the target audience described above. Because most of this audience achieved adolescence some time in the 90s, they will inherit the dominant humor paradigm of that era, i.e. sarcasm. Sarcasm is an inherently derogitory humor technique because it is a way of deriding what is being said through the use of an exagerated tone of voice, a tone that would presumably be used by one who actually agrees with the statement being said. Thus latent sarcasm must be a key component of humorous captions if they are to appeal to persons who achieved adolescence in the 90s. Indeed LOLCAT contains a great deal of sarcasm because it is mocking those users of the internet called newbies (aka newbs or n00bs or even n00bx0rz) who frequently misspell words and use acronyms such as LOL and OMG. Such newbies are the victims of constant derision by more experienced internet users. Thus LOLCAT captures the sarcastic qualities necessary to sustain the interest of the target audience described above. QED ■

Thus we have clearly shown how the socio-economic factors of the modern American economy have contributed to the rise of LOLCAT as an internet sensation.

III. ROFL Algorithm

We now present a method for the analysis of trends in the LOLCAT language via an automated data-retrieval algorithm termed Real-Time Omnibus Feline Linguistics (ROFL). The goal of the algorithm is to track the usage of LOLCAT vocabulary and syntax. The vocabulary we wish to track is a set of Assinine Acronyms (AAs) that are common in the LOLCAT lexicon. Examples of AAs are Laughing Out Loud (LOL), Oh My God (OMG) (note: this AA is usually followed by at least three exclamation points interspersed with '1's), and Rolling on the Floor Laughing My Ass Off (ROFLMAO).

The syntax to be tracked is a set of template phrases or "macros" commonly used by LOLCAT speakers. These are illustrated in Figure 2.

The algorithm works via the cutting-edge functionality of the Windows Application Programming Interface (API). The procedure of the ROFL algorithm is detailed in Algorithm 1.

Algorithm 1: ROFL Algorithm

Move mouse cursor using WinAPI;
Open Internet Explorer;
Navigate to www.icanhascheezburger.com;
database = [];
while *true* **do**
    Turn mouse pointer into hourglass;
    image = TakeScreenshot();
    text = OCR(image);
    database = PutInDatabase(database, text);
    Turn mouse pointer into arrow;
    Position mouse cursor over Refresh button;
    Click Mouse cursor;
    if *Control-C()* **then**
        return database;
    end
end

Once the algorithm generates a database of LOLCAT vocabulary and syntax this database can be easily queried to produce statistics about the prevalence of certain trends in the LOLCAT language. The prevalence of a certain AA or macro in the database is calculated using Equations 1 and 2, respectively.

$$P(AA) = \frac{e^{\sqrt{-\pi Freq(AA)\delta t}}}{q} \tag{1}$$

$$P(Macro) = \frac{\left(\log\sqrt{-\phi Freq(Macro)\delta t}\right)^2}{q} \tag{2}$$

where δt is the change in time since the beginning of the universe, ϕ is the golden ratio, $Freq(...)$ is the proportion of the argument in the database, and q has no meaning whatsoever.

IV. LOLCAT Statistics

In this section we discuss recent trends in the LOLCAT language as determined using ROFL. The data discussed was taken beginning at the founding of www.icanhascheezburger.com. Examples of each type of macro considered are shown in Figure 2. Statistics gathered are shown in Figures 3 and 4 for AAs and macros, respectively.

From the data displayed in the graphs, it is clear that certain AAs are rising in popularity while others are going out of style. LOL and WTF are increasing in popularity while the combersome and blasphemous ROFLMAO and OMG, respectively, are decreasing rapidly in popularity. In terms of macros, the “i has them” and “invisible” macros are currently dominating and miscelaneous is holding strong. “im in ur” has seen a steady decline since its inception.

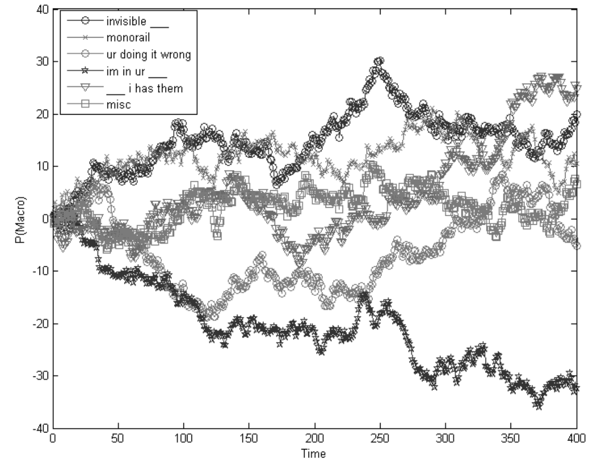


Fig. 3. LOLCAT macro prevalence. Matlab skills, I has them.

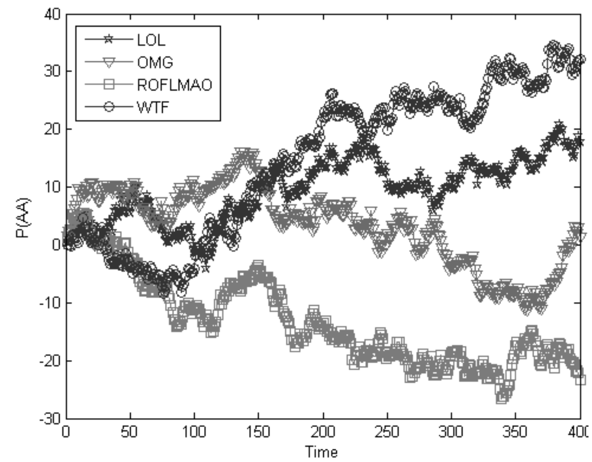


Fig. 4. LOLCAT AA prevalence. Srsly, I has them.

V. Conclusion

In conclusion we have presented a thorough and convincing analysis of the socio-economic factors behind the LOLCAT language. We have also described an algorithm for the automatic collection of LOLCAT data for later analysis. Our LOLCAT prevalence computation accurately captures the current trends of LOLCAT AAs and macros and has been used to generate the informative statistics presented in this paper. KTHNXBYE!

References

[1] http://globalnerdy.com/2007/05/28/lolcode-the-lolcat-programming-language/
[2] http://www.lolcatbible.com/
[3] www.lolinator.com
[4] www.icanhascheezburger.com
[5] http://en.wikipedia.org/wiki/Lolcats
[6] Dwight Silverman, “IM IN UR NEWSPAPER WRITIN MAH COLUM :),” Houston Chronicle, June 2007.
[7] Anil Dash, “Cats Can Has Grammar,” May 2007.

General Case Rendering from Occurring Instances

John Kua
Institutionalized Robotics
Carnage Melon University
Pittsburgh, PA 15213
Email: jkua@cmu.edu

Abstract

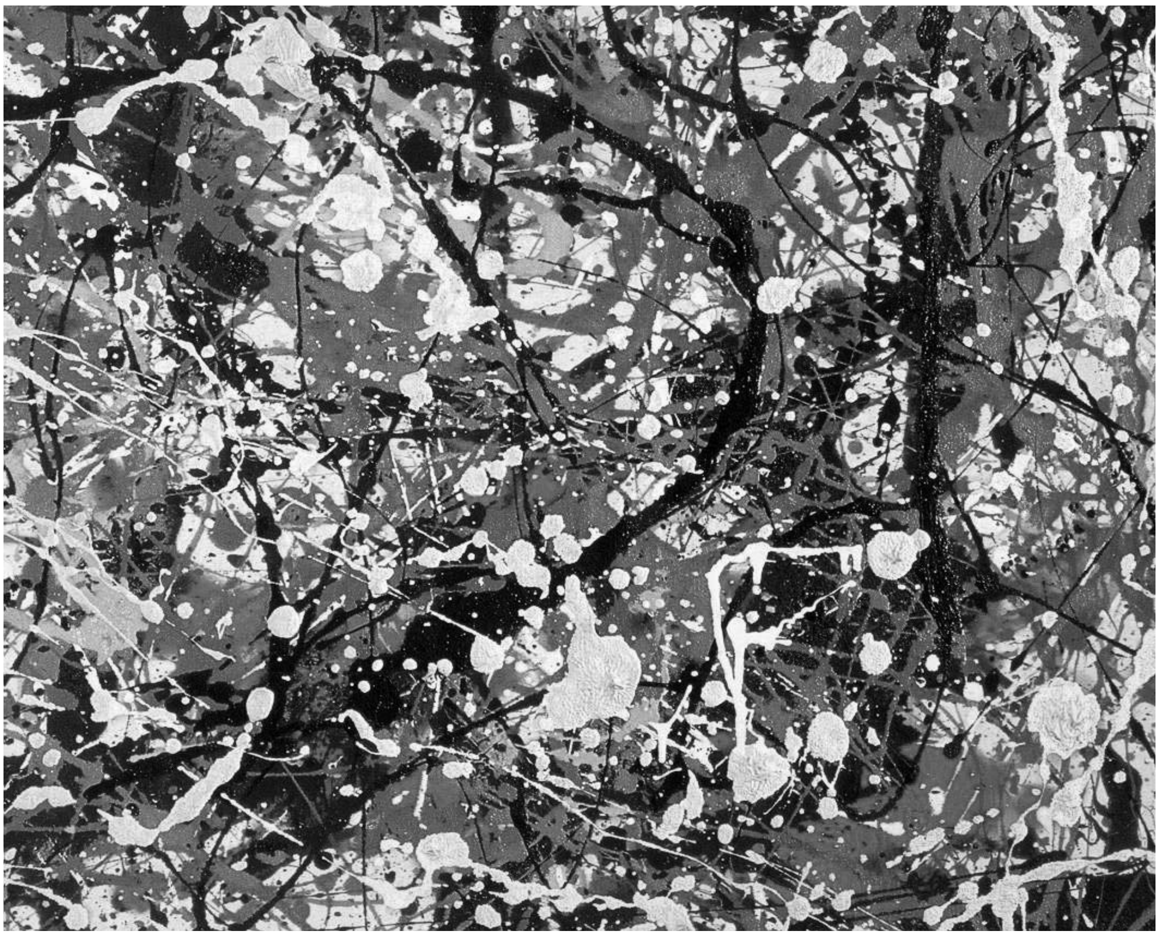


Fig. 1. Number 8 [1]

I. Introduction

PAUL Jackson Pollock (January 28, 1912 – August 11, 1956) was an influential American painter and a major force in the abstract expressionist movement. Pollock was born in Cody, Wyoming in 1912, the youngest of five sons. His father was a farmer and later a land surveyor for the government. He grew up in Arizona and Chico, California, studying at Los Angeles' Manual Arts High School. During his early life, he experienced Indian culture while on surveying trips with his father. In 1929, following his brother Charles, he moved to New York City, where they both studied under Thomas Hart Benton at the Art Students League of New York. Benton's rural American subject matter shaped Pollock's work only fleetingly, but his rhythmic use of paint and his fierce independence were more lasting influences. From 1938 to 1942, he worked for the Federal Art Project.

II. The Springs Period

In October 1945, Pollock married another important American painter, Lee Krasner, and in November they

[REDACTED DUE TO GDFL]

tack the unstretched canvas to the hard wall

[REDACTED DUE TO GFDL]

mathematical fractals

[REDACTED DUE TO GOLF]

’This is it.’’

[REDACTED DUE TO GDFL]

V. Conclusion

Pollock did not paint at all in 1955. After struggling with alcoholism his whole life, Pollock's career was cut short when he died in an alcohol-related, single car crash in his Oldsmobile convertible, less than a mile from his home in Springs, New York on August 11, 1956 at the age of 44. One of his passengers, Edith Metzger, died, while the other passenger, Pollock's girlfriend Ruth Kligman, survived. After his death, Pollock's wife, Lee Krasner, managed his estate and ensured that Pollock's reputation remained strong in spite of changing art-world trends. They are buried in Green River Cemetery in Springs with a large boulder marking his grave and a smaller one marking hers.

Acknowledgment

The author would like to thank the many contributors to Wikipedia [2], from which all the above text is borrowed. And of course, Jackson Pollock, for his art.

References

[1] Pollock, P. J., "Number 8," Oil, enamel, and aluminum paint on canvas, 1949.
[2] "Jackson Pollock," *Wikipedia*, http://en.wikipedia.org/wiki/Jackson_Pollock, Retrieved March 8, 2008, 11:00pm.

Track 7: Logic and Applications

Objectivist Conceptual Dependency

Corey Kosak*, Naju Mancheril†, Kevin Watkins‡
QVT Financial LP

April 1, 2008

Abstract

We present a novel framework for generating unbounded profit in an automated fashion. The framework, *Objectivist Conceptual Dependency (OCD)*, uses techniques from Conceptual Dependency Theory to represent and symbolically manipulate philosophically rigorous propositions from the domain of Objectivism. Entities in our system exhibit certain remarkable properties. We show that every well-formed OCD graph is also a philosophical position consistent with Objectivism. Furthermore, we show that by introducing a fourth axiom, the Axiom of Profit, we are able to analyze the profit-making potential of each proposition in addition to its philosophical consistency. We conclude with an analysis of preliminary results from TradePal, our prototype implementation, running in a real-world trade setting.

1 Introduction

At many financial firms, investment professionals spend a significant amount of time on the telephone—making deals, listening to sales pitches, evaluating investment theses, and so on. Generally, these professionals are highly compensated [Bastone, 2005]; thus, there is significant value in saving them even a small amount of time. Our first attempt to address this problem was TradeBot, an automated telephone answering system. While superficially similar to the so-called "phone trees" used at the customer service centers of firms such as Sanrio and American Girl Place, TradeBot owes its true heritage to much older systems, such as BulbMan [Bovik and Greenspan, 1636]. TradeBot provides callers with an automated set of options such as "Press 1 to offer bonds" and "Press 5 to trade illiquid structured credit." We found, however, that the inherent rigidity and impersonal affect of such systems led to financially suboptimal outcomes. We were determined to address these issues with TradePal, the next version of our system.

We designed TradePal with two goals in mind. First, that it have a rich understanding of human natural language, which it would use to put our callers at ease. Second, that it be grounded in a firm mathematico-philosophical foundation, so that it would generate only rational (and therefore profitable) trade suggestions.

*kosak@cs.cmu.edu
†ngm@alumni.cmu.edu
‡kw@cs.cmu.edu

We considered a number of existing frameworks and found that they could be broadly classified into two groups. The first group consisted of frameworks that were philosophically pristine but covered rather limited domains: these included the frameworks Twelf [Pfenning and Schürmann, 1999] and Standard ML [Milner, 1984]. The other group, broad in scope but philosophically inconsistent, included frameworks such as "Special" Relativity, Quantum "Mechanics," and Fuzzy "Logic." These were found to be so tainted by corrupt 20th century mathematics as to be essentially useless.

There is only one philosophical system uncompromisingly rigorous enough to be trusted with our clients' money, and there is only one language framework rich enough to convey our traders' intention. These are Objectivism, the philosophy of Ayn Rand [Rand, 1979, Peikoff, 1993], and Conceptual Dependency Theory, the comprehensive methodology for diagrammatic representation of language meaning [Schank, 1972]. It was upon these foundations that we decided to build TradePal.

2 Axioms

The first step towards enabling the system to model reality is to provide it with certain base truths from which everything else can be inferred. In Objectivism, there are three known axioms: the *Axiom of Identity*, the *Axiom of Existence*, and the *Axiom of Consciousness*. It is widely believed that in addition to these, there is a fourth axiom, which we call the *Axiom of Profit*. While (self-evidently) it must be possible to represent these axioms in Conceptual Dependency Theory, the discovery of these representations was a significant challenge.

2.1 A is A—The Axiom of Identity

Figure 1: *A is A*—the Objectivist Axiom of Identity

Simply put, this axiom expresses what Rand refers to as *the primacy of existence*: "reality, the external world, exists independent of man's consciousness... this means that A is A, that facts are facts, that things are what they are..." A is A is usually credited to Aristotle, but prior to 1957, there was significant debate in the philosophical "community" about whether Aristotle truly established that A is A in his work, or was really just as lost as everyone else. The debate was finally put to rest with the publication of *Atlas Shrugged*:

> "Centuries ago, the man who was—no matter what his errors—the greatest of your philosophers, has stated the formula defining the concept of existence and the rule of all knowledge: A is A. A thing is

itself. You have never grasped the meaning of his statement. I am here to complete it: Existence is Identity, Consciousness is Identification."
—John Galt, *Atlas Shrugged*

Given that this was, arguably, the most significant and fundamental axiom in the system, we took special care in the modelling of it. Figure 1 depicts the results. In the figure, the double-headed, double-lined arrow indicates the inherent duality of the relationship: that it is both the case that *A is A*, and that *A is A*.

2.2 Existence Exists—The Axiom of Existence

$$\boxed{\text{EXISTENCE}}$$

Figure 2: *Existence Exists*—the Objectivist Axiom of Existence

This axiom was a challenge to implement due to its self-evident nature. The difficulty being, how might one explain a self-evident thing to a computer program? This issue might be likened to trying to teach a computer the binary rule that $1 + 1 = 10$: such a thing simply *is*, and any attempt to refute it would have to implicitly assume it.[1]

In the Conceptual Dependency framework, this issue translates to the problem of finding which of the standard Primitive Acts (ATRANS, PTRANS, PROPEL, EXPEL, etc.) apply. In a moment of childlike clarity, we realized that the proper representation has *no* Primitive Act whatsoever! The result is shown in Figure 2.

2.3 The Axiom of Consciousness

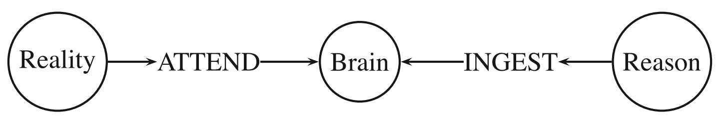


Figure 3: The Objectivist Axiom of Consciousness

This axiom posed the greatest challenge of all. On the one hand, it presumes the Axiom of Existence, because "If nothing exists, there can be no consciousness: a consciousness with nothing to be conscious of is a contradiction in terms." [Rand, 1957]. However, unlike the Axiom of Existence, there clearly must be a Conceptual Dependency Primitive Act, as a person has to be conscious of something (as well as of something). Our attempts to extend the classical Conceptual Dependency model worked, but were clumsy and unsatisfying. Ultimately, however, we realized that consciousness can readily be modeled by a combination of the classical operators ATTEND and INGEST; we depict our model in Figure 3.

[1] Ayn Rand and Grace Murray Hopper, private communication.

2.4 The (hypothesized) Axiom of Profit

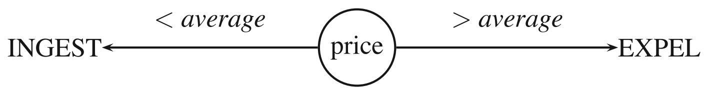


Figure 4: The (hypothesized) Objectivist Axiom of Profit

Rand strongly hinted at the existence of a fourth axiom, particularly in her “fictional” works *The Fountainhead* and *Atlas Shrugged* [Rand, 1943, Rand, 1957]. This axiom, which we shall call the Axiom of Profit, has, unlike the others, no agreed-upon formal definition. However a workable approximation is afforded by the maxim “Buy low, sell high.” Figure 4 depicts the modelling of this axiom in our system.

3 TradePal in Action

We omit any discussion of the inference rules and combining operators of our system, as they can be derived by a straightforward application of Reason. Instead, we proceed to describe some implementation details and our real-world results.

Our implementation of TradePal consists of about 150,000 lines of code, written in Objective Caml. With the system in place, we are able to calculate the profit score of any proposition. Propositions that score highly in the system include concepts such as “I would like to sell you a mortgage” and “Read my lips: No new taxes.” Low-scoring propositions include “I wish to overpay for that donut” and “President Hillary Rodham Clinton.”

What is remarkable about our system is that it has the ability to not only respond in a rational manner (as it must) to any user input, but more importantly, when there are a variety of responses possible, to choose *the most profitable* course of action. The following real-world dialogue illustrates such an interaction:

Caller: “Hey man, I have 25 million bonds for you at 95. It’s a good deal, and anyway, you gotta help me out, I got wasted at *Scores* last night, and when my wife found out, she locked me out of the house.”

At this point TradePal considers three potential responses:

1. “I accept your offer.”
2. “No way, are you trying to rip my face off?”
3. “Tell your wife, ‘Don’t hate the player, hate the game.’ ”

Remarkably, TradePal correctly identifies the third response as being the most ingratiating to the caller, and therefore the most likely to improve the terms of the transaction. In the above real-world scenario, this response actually yields a discount of a full 75 basis points.

4 Conclusions and Future Work

Our initial work with TradePal provides substantial experimental support for the validity of the Axiom of Profit, as applied to the domain of finance. However, before it can be accepted as a true axiom of Objectivism, it must be shown to hold universally. Toward this end, we are now trying to adapt the TradePal framework to multiple alternative domains outside finance.

Our economic research team has theoretically proven that running two TradePal instances against each other and taxing both sides provides a crude, but effective, solution to problems of economic scarcity. In practice, this has proven to be more difficult to implement. Current TradePal implementations have yet to find ways to consistently turn a profit once a taxation module is introduced, but we believe this area holds much potential for future research.

TradePal's ability to generate profit while meeting the mental needs of its user makes it an ideal starting point for future clinical psychology research. It may also provide an attractive framework for developing next-generation entertainment applications. The incorporation of a simple graphics engine would provide hours of entertainment for both casual and experienced video game players.

One cautionary tale was provided by our medical triage project at Johns Hopkins Hospital. At one point in the project, the system diverted resources from the neonatal intensive care unit to the more profitable Gates/Buffet Severed Head Cryosuspension Ward. While this was the economically proper decision, care should be taken to educate doctors and patients beforehand about the basic principles of capitalism.

Finally, a number of military research labs have provided generous grants for future TradePal research. We have currently partnered with Cyberdyne Systems, a small manufacturing company based in Sunnyvale, California, to provide the natural language and decision-making capabilities of their *SkyNet* satellite network [Dyson and Brewster, 1997]. We expect this partnership to yield many promising results in the years to come.

References

[Bastone, 2005] Bastone, W., editor (2005). *Tyco Chief's Crimes Against Humility*. The Smoking Gun. http://www.thesmokinggun.com/archive/0617051tyco1 html.

[Bovik and Greenspan, 1636] Bovik, H. Q. and Greenspan, A. (1636). BulbMan — an artificial man who shall assist in the orderly exchange of tulip bulbs between gentlemen. In Galilei, G., editor, *Proceedings of the 1309th Annual Holy See Conference on Science and Faith*, pages 11–91, Amsterdam, Republic of the Seven United Netherlands.

[Dyson and Brewster, 1997] Dyson, M. B. and Brewster, R. (1997). Skynet — A benevolent, rational AI framework for killer-satellite command and control. In *Proceedings of the 1997 Annual IEEE Conference on Systems, Man, and Cybernetics*, Orlando, Florida. IEEE.

[Milner, 1984] Milner, R. (1984). A proposal for Standard ML. In *LFP '84: Proceedings of the 1984 ACM Symposium on LISP and functional programming*, pages 184–197, New York, NY, USA. ACM.

[Peikoff, 1993] Peikoff, L. (1993). *Objectivism: The Philosophy of Ayn Rand*. Meridian, Australia.

[Pfenning and Schürmann, 1999] Pfenning, F. and Schürmann, C. (1999). System description: Twelf — A meta-logical framework for deductive systems. In Ganzinger, H., editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, pages 202–206, Trento, Italy. Springer-Verlag LNAI 1632.

[Rand, 1943] Rand, A. (1943). *The Fountainhead*. Bobbs Merill, USA.

[Rand, 1957] Rand, A. (1957). *Atlas Shrugged*. Random House, USA.

[Rand, 1979] Rand, A. (1979). *Introduction to Objectivist Epistemology*. Meridian, Australia.

[Schank, 1972] Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):pages 532–631.

A Focused Approach To Focus on Focusing

Robert J. Simmons

April 6, 2008

Abstract

Focusing [And, Zei07] is a logical technique for explaining everything in a way only slightly more confusing than the last time you learned it. We apply focusing techniques to the previously unexplored area of academic achievement.

1 Introduction

If you give a way of explaining anything, focusing will give you a better way to do the same thing. This has already been applied to fields from games to programming languages and culinary endevors. In this paper, we will write a paper, and then use focusing to do all of academics better.

The discipline of focusing asks us to split things ("types") into two groups, or *polarities*. There are the *positive* types, which are defined by their introduction (in other words, by their construction), and there are the *negative* types, which are defined by their elimination (in other words, by their use). We will analyze the polarity of common academic constructions in the remainder of this paper.

2 Positive Types

The definition of a positive type acts as a "template" for defining the possible value of that type. An example of a positive value in programming is a record created by filling in all of the record's fields, or a sum (in ML) or tagged union (in C) type created by specifying the tag and a corresponding value.

A *paper* would appear to be the canonical example of a positive type in academic systems. It is constructed by rules of its own determination, and it is the responsibility of the reader who wants to use a paper to define how to deal with *any possible paper* which they might be attempting to read (even if the result is "pass the f— out" in most cases).

3 Negative Types

Whereas positive types are defined by their construction, giving *values* definitional priority, negative types are defined by their *elimination*, or use. A lazy pair is eliminated (used) by asking for the first or second element, which forces the lazy pair to cough up its first or second element. A function is used by throwing a argument at it, which causes it to cough up some output. Therefore, as part of the process of *creating* a function or lazy pair (a *negative value*), there needs specified a way of handling *any possible elimination form* – in other words, a negative value is created by preparing for any possible way of asking the negative value to cough up stuff.

The obvious example of a value of negative type in academia is a *conference presentation*. A conference presentation is defined by how it is used (by watching and asking questions), and the result of this process of watching/questioning will be information of some type. Creating a value of type "conference presentation," then, requires preparing for any possible well-formed ("well-typed") watcher/questioner.[1]

Similar to the PTFO-cases in the elimination form of positive values, the type of ensuing information from a conference presentation can be, in non-ideal, real-world cases, the information-free "unit" (in ML) or "void" (in Java/C) type for a most or all elimination forms (i.e. questions).

References

[And] Jean-Marc Andreoli, *Logic programming with focusing proofs in linear logic*, Journal of Logic and Computation **2**, no. 3, 297–347.

[Zei07] Noam Zeilberger, *On the unity of duality*, January 2007.

[1] The analouge of non-answers or "I'll take this offline" is an unsound logic admitting non-termination or exceptions – or, in the case of off-topic questions, dynamic type errors – but this is outside the scope of this work.

This Is Spin̈al Type APplic̈ation̈

RH and DRL

March 20, 2008

1 System $\ddot{\mathsf{F}}$ in Spin̈al Form̈

Syntax:

$$
\begin{array}{llcl}
\text{Type} & A & ::= & u \mid A_1 \to A_2 \mid \forall u.A \\
\text{Term} & M & ::= & x \mid \lambda x{:}A.M \mid \Lambda u.M \mid M \cdot S \\
\text{Spine} & S & ::= & \bullet \mid M;S \mid A;S \\
\text{Context} & \Gamma & ::= & \bullet \mid \Gamma, x:A \mid \Gamma, u
\end{array}
$$

Identify term̈s up to α-equivalenc̈e. Regard contexts as having the form $u_1, \ldots, u_n, x_1 : A_1, \ldots, x_k : A_k$. Substitutions $[A/u]B$, $[A/u]M$, and $[M_1/x]M_2$ defined as usual. Concatenation of spines is defined in̈ the evident manner.

Static seman̈tics for types:

$$
\frac{}{\ddot{\Gamma}, u \vdash u}
\qquad
\frac{\ddot{\Gamma} \vdash A_1 \quad \ddot{\Gamma} \vdash A_2}{\ddot{\Gamma} \vdash A_1 \to A_2}
\qquad
\frac{\ddot{\Gamma}, u \vdash A}{\ddot{\Gamma} \vdash \forall u.A}
$$

Type equivalen̈ce is α-equivalence, but one ẅould change this for F_ω, for example.

Static semantics for terms:

$$
\frac{}{\Gamma, x : A \vdash x : A}
\qquad
\frac{\Gamma \vdash M : A \quad \Gamma \vdash S : A > B}{\Gamma \vdash M \cdot S : B}
$$

$$
\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}
\qquad
\frac{\Gamma, u \vdash M : A}{\Gamma \vdash \Lambda u.M : \forall u.A}
$$

$$
\frac{}{\Gamma \vdash \ddot{\bullet} : A > A}
\qquad
\frac{\Gamma \vdash M : A \quad \Gamma \vdash S : B > C}{\Gamma \vdash M; S : A \to B > C}
$$

$$
\frac{\Gamma \vdash A \quad \Gamma \vdash S : [A/u]B > C}{\Gamma \vdash A; S : \forall u.B > C}
$$

In the judgement $\Gamma \vdash M : A$, Γ and M are inputs and B is outp̈ut. In the judgement $\Gamma \vdash S : A > B$, Γ, S, and A are inputs and that B is output.

Equivalence of terms is the least congruen̈ce containing these rules:

$$
(\lambda x : A.M) \cdot (N; S) \equiv ([N/x]M) \cdot S \qquad (\Lambda u.M) \cdot (A; S) \equiv ([A/u]M) \cdot S
$$

$$
M \cdot \bullet \equiv M \qquad (M \cdot S) \cdot S' \equiv M \cdot (S\, S')
$$

2 System IF in Spinal Form

Revised syntax of spines:

$$
\begin{array}{lrcl}
\text{Spine} & S & ::= & \bullet \mid M; S \mid \hat{A}; S \\
\text{Optional Type} & \hat{A} & ::= & \_ \mid A
\end{array}
$$

Revised static semantics of spines, adding one rule:

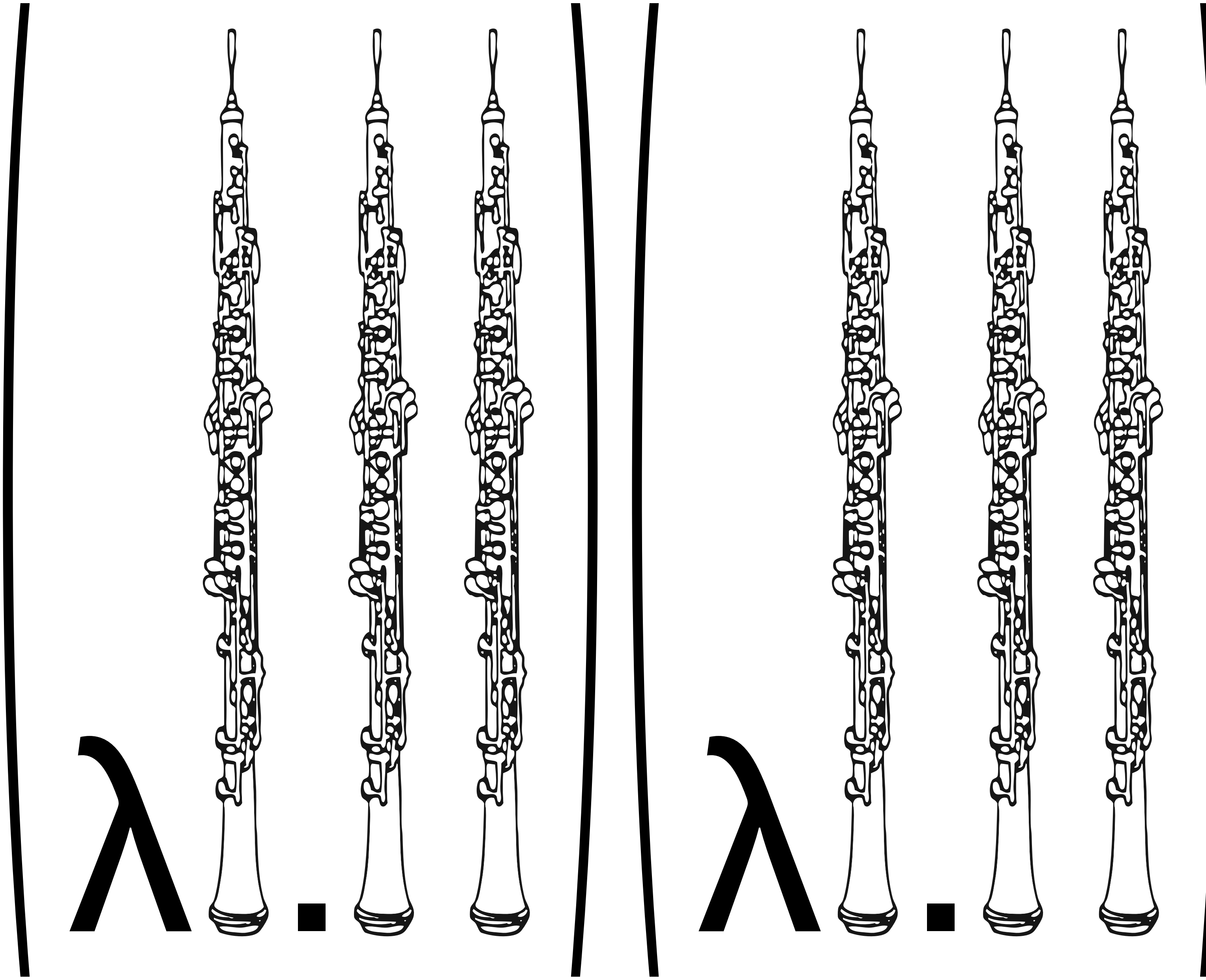
$$\frac{\Gamma \vdash A \quad \Gamma \vdash S : [A/u]B > C}{\Gamma \vdash \_; S : \forall u.B > C}$$

The type A is "guessed" non-deterministically. Algorithmically, one uses a meta-variable, α, as a placeholder for A, and uses matching to substitute for α as type checking progresses through the spine. Locality is achieved by insisting that all meta-variables be bound by the end of the spine—that is, that the result type, C, not involve meta-variables.

Additional rules for spine equivalence:

$$\_; S \equiv A; S \qquad A; S \equiv \_; S$$

The other two cases (both spines start with A or with a blank) are implied by reflexivity (over the revised syntax). Intuitively, one is "guessing" that the omitted type is the "other" type; when both are guessed, the choice is free.



=

OBOEMEGA

Track 8:
Real World/Computer Interaction

From the ACH SIGBOVIK 2008 conference proceedings.

Physics-Based Modeling of Operating Systems

Adam W. Bargteil
Carnegie Mellon University

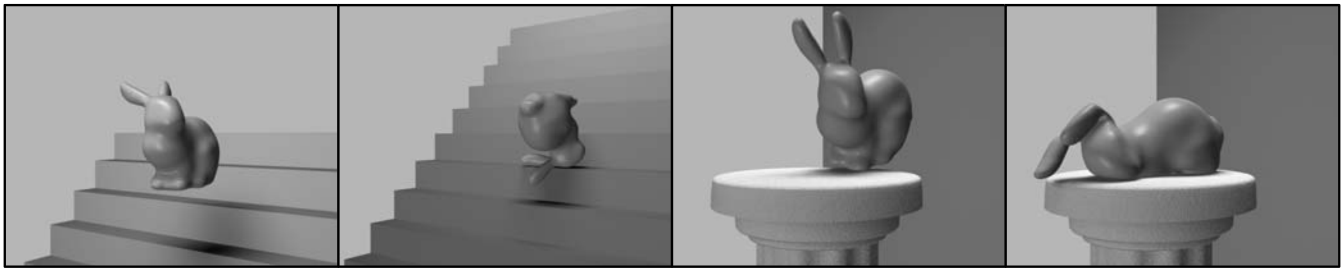


Figure 1: *The leftmost image is your brain. The rightmost image is your brain on windows.*

Abstract

We present physics-based models of a variety of commercial operating systems.

CR Categories: D.4.8 [Software]: Operating Systems—Modeling and Prediction; D.4.8 [Software]: Operating Systems—Simulation; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.6.8 [Simulation and Modeling]: Types of Simulation—Continuous.

Keywords: Natural phenomena, physically based modeling, operating systems, Steve Jobs, Bill Gates, Windows, Mac OS, Fedora.

1 Introduction

In this paper we present physics-based models of operating systems. The models were derived from an intensive user study and the resulting models are accurate to within −1000%. We compare the models to understand why microsoft windows is so painful to use. We come to the surprising conclusion that windows is the evil love-child of Jerry Lee Lewis and the PDP-666 and that Bill Gates would make an excellent carpenter.

2 Models

In this section we describe our physics-based models of operating systems. Representative plots of our models can be seen in Figure 2.

adamwb@cs.cmu.edu

ACH SIGBOVIK 2008, Pittsburgh, PA

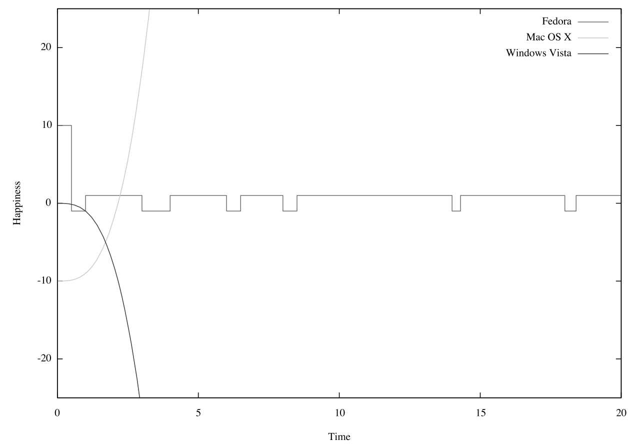


Figure 2: *Happiness versus time for three different operating systems.*

2.1 Fedora

The first operating system we consider is Fedora linux. Following the lead of the Catholic church and the illustrious emacs, Red Hat linux separated into two branches in 2003. The first branch, Red Hat, became a commercial software package, preventing any self-respecting hacker from usage. Fedora became the open-source branch with a much better name. While largely inferior to many other linux distributions, it is supported by `help@cs` and was used extensively in our tests. We first introduce our model of happiness while using fedora. We model happiness versus time with the following function:

$$h(t) = \begin{cases} +10 & \text{until the first thing breaks} \\ -1 & \text{when something is broken} \\ +1 & \text{when everything is working again} \end{cases} \tag{1}$$

We can see that in the integral both momentum and joy are conserved:

$$\int_{\text{first use}}^{\text{death}} h(t)dt = ☺ \tag{2}$$

2.2 Mac OS X

Next we turn our attention to Apple's Mac Operating System. Before the introduction of large cats, geeks the world over looked down their noses and snickered at the apple operating systems. The mouse only had a single button, we scoffed while using our low-resolution three-button optical mice that required special reflective pads. But then came the large panthers, tigers, and leopards that devoured the old alliances and the mighty mouse with its scrolling ball. Now the happiness versus time function is much better approximated as:

$$h(t) = t^{\infty} - 10. \tag{3}$$

As you can see there is a slight uncomfortable learning curve, but quickly the user acclimates and begins to wonder why he is so happy all the time. The corresponding integral is, of course:

$$\int_{\text{first use}}^{\text{death}} h(t)dt = \text{☺}^{\infty} \tag{4}$$

The author suspects that if users are introduced to Mac OS early enough in life, they may be too happy to die.

2.3 Microsoft Windows Vista

Unfortunately the author was unable to bring himself to bear the pain of sitting at a machine running windows after the severe scars from past experiences with Windows 3.1, 95, 2000, and XP. We extrapolate those experiences and propose the following model (due to the fact that latex allows only a single exponent, we must express this as a recurrence):

$$h(t) = -t^{g(x)} \tag{5}$$
$$g(x) = g(x)^{\infty}. \tag{6}$$

As you can see with this model, at first the user thinks, *maybe this won't be so bad*. But, quickly they begin to understand the true meaning of suffering and long to roam the deepest depths of Mordor for eternity. The integral is clearly:

$$\int_{\text{first use}}^{\text{early death}} h(t)dt = \text{☹}^{\infty}. \tag{7}$$

3 Conclusions and Future Work

There are several conclusions we can draw from our models. First, Linux ain't so bad. Second, mac is better. Third, we should pity those who use windows and hope that they see the light before they become so miserable they no longer leave the house. Finally, we conclude that Bill Gates should have been a carpenter and saved the world much pain.

There is still more work to be done in this very important area of research. The author has developed accurate, robust and reliable models for happiness versus time. But, there are still other metrics to be considered, such as productivity versus windows, X11 versus the crocodile hunter, and Smurfs versus Elves. Initial testing indicates that the latter may be decided in favor of a hybrid smurfy-elf.

Acknowledgements

This paper was brought to you in part by the numbers g and i, the letters a–y and Tom Waits.

Steve Jobs. Taken by: Matthew Yohe.
(http://en.wikipedia.org/wiki/Image:SteveJobsMacbookAir.JPG)

Bill Gates. Taken by: Kees de Vos.
(http://commons.wikimedia.org/wiki/Image:Bill_Gates_2004.jpg)

Automatic Citation [McCann 2008]

James McCann*
Carnegie Mellon University

$$\text{Hello World} \xrightarrow{citeseer} \text{Hello World} \text{ [ACM 2007]}$$

Figure 1: A more well-referenced global greeting.

Abstract

As it is bad form to place citations in an abstract, we have omitted the abstract.

CR Categories: O.N.0 [Research]: Paper Writing—Citation

Keywords: cite, citation, automatic, perl

1 Introduction

One of [Schapire et al. 1997] the big [Erol et al. 1994] challenges [Ng et al. 1998] facing computer science [Brezany] researchers today [Salzberg] is the proper identification of [Yahalom et al. 1993] relevant work [Doucet 1998] to cite. If [Breitzman et al. 2000] a researcher does not [Dotzauer and Holmstrom 1998] cite the [Page et al.] work [Sanderson 1994] of, e.g., one of [Boyen and Koller 1998] the reviewers, he often finds himself unable to publish otherwise adequate results. To aid in [Macready and Wolpert 1996] such [Munzner and Burchard 1995] scenarios, we have devised a system for automatically inserting citations into research documents.

2 Background

We have [IDA 1996] omitted the background section; please refer to [Kim and Wolisz] the extensive citations throughout [Fidge 1998] the rest of [Dekhtyar and Subrahmanian 1997] the [Colby et al. 2000] paper.

3 Method

Our system works by processing [McDaniel et al. 1998] a paper word-by-word. Each word [Price et al. 1998] is added [Srihari et al.] to the current context when it is processed. If the current context is long [Manke et al. 1995] enough, it is then used as a citeseer query. If this query returns no results, [LU et al. 2003] a word is removed from the [Gross et al. 1992] context (in standard FIFO [Wang and Stavrakakis 1996] order) and [Chen and Nahrstedt 1998] the search is repeated. If related work results from the query, a citation command is inserted [Dekkers et al.] into the text and [Dvorsk et al. 1999] the appropriate [Sen et al. 1999] bibtex entry [Jones et al. 2001] is downloaded [Dean et al. 1996] from citeseer [Aberer et al. 2003] to be inserted into the bibliography file. In this way, [Neuman 1989] a relevant citation is introduced [Olshausen and Field 1997] into the text every few words. In case of multiple relevant citations, papers that have not [LaValle and Ku 1999] already been cited are preferred.

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

4 Results

As a preliminary result, we present this [Debevec et al. 1996] paper [McCann 2008].

5 Limitations

Unfortunately, having so many citations does tend to inflate both the length of the [Ramalingam et al. 1999] sections of a paper and the total page count (due to [Cidon and Mokryn 1998] bibliographic length). This means that papers must [Teaching] be [Kelton] written [Chapman et al. 1992] in terse prose.

6 Conclusions

In this paper, we [Herlihy 1991] presented an automatic citation tool which eases the burden of background research and [Billinghurst et al. 1998] provides authors with peace of mind (as [Kent] well as inflated page counts). We hope this tool [Beton 1996] aids the community as [Paper] a whole.

Acknowledgments

Thanks to perl, for being such [Dolev et al. 1995] an [Mizzaro 1997] awesome wget control language. Also, to Pittsburgh's weather, for giving me a reason to stay inside and actually [Bjreland and Driankov] get something done.

References

ABERER, K., CUDRÉ-MAUROUX, P., AND HAUSWIRTH, M. 2003. The Chatty Web: Emergent Semantics Through Gossiping. In *Proceedings of the 12th International World Wide Web Conferenc e*.

ACM, 2007. Hello world! http://www2.latech.edu/ acm/HelloWorld.shtml.

BETON, R. 1996. From Object Oriented Analysis to Implementation Using Occam. In *Parallel Processing Developments*, IOS Press, B. O'Neill, Ed.

BILLINGHURST, M., BOWSKILL, J., JESSOP, M., AND MORPHETT, J. 1998. A wearable spatial conferencing space. In *ISWC*, 76–83.

BJRELAND, M., AND DRIANKOV, D. Synthesizing discrete controllers from hybrid automata.

BOYEN, X., AND KOLLER, D. 1998. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 33–42.

BREITZMAN, A., THOMAS, P., AND CHENEY, M., 2000. Technological powerhouse or diluted competence: Techniques for assessing mergers via patent analysis.

BREZANY, P. Compiling fortran for massively parallel architectures.

CHAPMAN, B. M., MEHROTRA, P., AND ZIMA, H. P. 1992. Programming in Vienna Fortran. *Scientific Programming 1*, 1, 31–50.

CHEN, S., AND NAHRSTEDT, K., 1998. An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions.

CIDON, I., AND MOKRYN, O. 1998. Propagation and leader election in a multihop broadcast environment. In *International Symposium on Distributed Computing*, 104–118.

COLBY, C., LEE, P., NECULA, G. C., BLAU, F., PLESKO, M., AND CLINE, K. 2000. A certifying compiler for Java. *ACM SIGPLAN Notices 35*, 5, 95–107.

DEAN, D., FELTEN, E. W., AND WALLACH, D. S. 1996. Java security: from HotJava to Netscape and beyond. In *1996 IEEE Symposium on Security and Privacy: May 6–8, 1996, Oakland, California*, IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, IEEE, Ed., 190–200.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics 30*, Annual Conference Series, 11–20.

DEKHTYAR, A., AND SUBRAHMANIAN, V. S. 1997. Hybrid probabilistic programs. In *International Conference on Logic Programming*, 391–405.

DEKKERS, C., KOSTER, C., NEDERHOF, M.-J., AND VAN ZWOL, A. Gwb - manual for the grammar workbench version 1.5.

DOLEV, S., KRANAKIS, E., KRIZANC, D., AND PELEG, D. 1995. Bubbles: adaptive routing scheme for high-speed dynamic networks. 528–537.

DOTZAUER, E., AND HOLMSTROM, K., 1998. The nlplib graphical user interface.

DOUCET, A., 1998. On sequential monte carlo sampling methods for bayesian filtering.

DVORSK, J., POKORN, J., AND SNSEL, V., 1999. Word-based compression methods and indexing for text retrieval systems.

EROL, K., HENDLER, J., AND NAU, D., 1994. Semantics for hierarchical task network planning.

FIDGE, C. J. 1998. Real-time schedulability tests for preemptive multitasking. *Real-Time Systems 14*, 1, 61–93.

GROSS, T., HASEGAWA, A., HINRICHS, S., O'HALLARON, D., AND STRICKER, T. 1992. The impact of communication style on machine resource usage for the iWarp parallel processor. Tech. Rep. CS-92-215.

HERLIHY, M. 1991. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems 13*, 1 (January), 124–149.

IDA, L., 1996. Issues on temporal representation of multimedia documents.

JONES, R., O'LEARY, J., SEGER, C., AAGAARD, M., AND MELHAM, T., 2001. Practical formal verification in microprocessor design.

KELTON, C. D. Preparing manuscripts for the informs journal on.

KENT, S. I. Designing and evaluating technology for independent aging in the home.

KIM, H. B., AND WOLISZ, A. Performance evaluation of a mac protocol for radio over fiber wireless lan operating in the 60-ghz band.

LAVALLE, S., AND KU, J. 1999. Randomized kinodynamic planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*. To appear.

LU, D., DINDA, P., AND SKICEWICZ, J., 2003. Scoped and approximate queries in a relational grid information service.

MACREADY, W. G., AND WOLPERT, D. H. 1996. On 2-armed gaussian bandits and optimization. Tech. Rep. 96-03-009, Santa Fe, NM.

MANKE, S., FINKE, M., AND WAIBEL, A. 1995. NPen++: A writer independent, large vocabulary on-line cursive handwriting Recognition System. In *Proceedings of the International Conference on Document Analysis and Recognition*. Montreal.

MCCANN, J. 2008. Automatic citation. In *ACH SIGBOVIK*, vol. 1, 79–87.

MCDANIEL, R., SCRIBNER, D., KREBS, W., WARREN, P., OCKMAN, N., AND MCCARLEY, J., 1998. Image fusion for tactical applications.

MIZZARO, S. 1997. Relevance: The whole history. *Journal of the American Society of Information Science 48*, 9, 810–832.

MUNZNER, T., AND BURCHARD, P. 1995. Visualizing the structure of the World Wide Web in 3D hyperbolic space. In *Proc. 1st Symp. The VRML Modelling Language: Special issue of Computer Graphics*, ACM Press, 33–38.

NEUMAN, B. C. 1989. The need for closure in large distributed systems. *Operating Systems Review 23*, 4, 28–30.

NG, M. K., HUANG, Z., AND HEGLAND, M. 1998. Datamining massive time series astronomical data sets - a case study. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 401–402.

OLSHAUSEN, B., AND FIELD, D., 1997. Sparse coding with an overcomplete basis set: A strategy employed by v.

PAGE, R. D. M., CLAYTON, D. H., AND PATERSON, A. M. Lice and cospeciation: A response to barker.

PAPER, F. D. Fcndp no. 179.

PRICE, M., SCHILIT, B., AND GOLOVCHINSKY, G., 1998. Xlibris: The active reading machine.

RAMALINGAM, G., FIELD, J., AND TIP, F. 1999. Aggregate structure identification and its application to program analysis. In *Symposium on Principles of Programming Languages*, 119–132.

SALZBERG, S. On comparing classifiers: A critique of current research and methods.

SANDERSON, M. 1994. Word sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, 49–57.

SCHAPIRE, R. E., FREUND, Y., BARTLETT, P., AND LEE, W. S. 1997. Boosting the margin: a new explanation for the effectiveness of voting methods. In *Proc. 14th International Conference on Machine Learning*, Morgan Kaufmann, 322–330.

SEN, S., REXFORD, J., AND TOWSLEY, D. F. 1999. Proxy prefix caching for multimedia streams. In *INFOCOM (3)*, 1310–1319.

SRIHARI, R. K., SRIKANTH, M., NIU, C., AND LI, W. Use of maximum entropy in back-off modeling for a named entity tagger.

TEACHING, S. F. Session 0575.

WANG, X., AND STAVRAKAKIS, I. 1996. Study of multiplexing for group-based quality of service delivery. In *Modelling and Evaluation of ATM Networks*, 360–379.

YAHALOM, R., KLEIN, B., AND BETH, T. 1993. Trust relationships in secure systems–A distributed authentication perspective. In *RSP: IEEE Computer Society Symposium on Research in Security and Privacy*.

Optimal censor placement in wireless censor networks

[redacted]
Department of [redacted]
[redacted]

ABSTRACT

Proper placement of censors is a fundamental task in obscuring information. Previous work has presented algorithms that [redacted] the [redacted] of each [redacted] in the network. However, this approach suffers a [redacted] of [redacted]. This work proposes a [redacted] of [redacted], which, under assumptions of [redacted], performs within [redacted] of optimal.

1. INTRODUCTION

[redacted] tragedy of the commons [redacted]. Unfortunately, [redacted]

[redacted] power-law [redacted]. [redacted] superexponential [redacted] otherwise [redacted]

[redacted] Wumpus World [redacted]. [redacted]. [redacted].

[redacted]. [redacted] maximum-margin eclipsoid [redacted].

2. PRELIMINARIES

[redacted]. [redacted] if [redacted] then [redacted]

[redacted]. k-armed bandits [redacted] and [redacted] norm balls [redacted]

[redacted]. [redacted] minimax [redacted]

3. RESULTS

(redacted)

4. CONCLUSION

[redacted]. [redacted]. [redacted]

[redacted] the reader's mother [redacted].

Expanded Exposure of Bipedal Robots to Everyday Phenomena

Andrew J. Harris   Pyry K. Matikainen   Garratt "Mad Dog" Gallagher

Abstract—**Many robotics researchers use test environments that are safe for their robots in the hopes that their robots will not be destroyed or lost during testing. Unfortunately, these test environments do not adequately model real world scenarios that a robot would frequently encounter. To assist these robotics researchers we discuss a variety of real world scenarios that are common yet difficult or hazardous to include as part of a standard test environment. We then demonstrate how these conditions affect a standard biped robot simulation modelled after the Sarcos [1] biped research robot.**

Index Terms—**crash, burn, linear quadratic regulator, functional programming, ML in space**

I. Introduction

ROBOTS are coming. Today their presence is quite limited, but they do make themselves visible by vacuuming our carpets and making our cars. Soon robots will take on more roles and responsibilities in our society. See Figure 1.

Fig. 1. Racquetball Anyone?

II. Motivation

But before you see the Sarcos robot in a nearby racquetball court, robotics researchers will be working hard to understand the dynamics of these robots while walking, running, and serving the racquetball to their unfortunate opponents.

This is where test environments come into play. Test environments are typically large open areas in the basement of Newell Simon Hall that provide a known environment in which the robots can operate safely. The floors of these test environments are typically littered with small blocks or pieces of pink paper all of which the robots consider to be obstacles.

However, a fundamental shortcoming of these test environments is that in reality there are obstacles all over the place and robots must learn to deal with them before they can take their proper place in society.

Our contribution in this paper is to enumerate three common real world scenarios that a biped robot will encounter. We demonstrate the effects of these scenarios on a simulated biped with size and mass characteristics similar to the Sarcos biped illustrated in Figure 1.

Devising systems and mechanisms that allow robots to recover from these scenarios is left to the reader.

III. Simulation

We first illustrate a biped taking a step in a simulated world containing a biped and a ledge. See Figure 2.

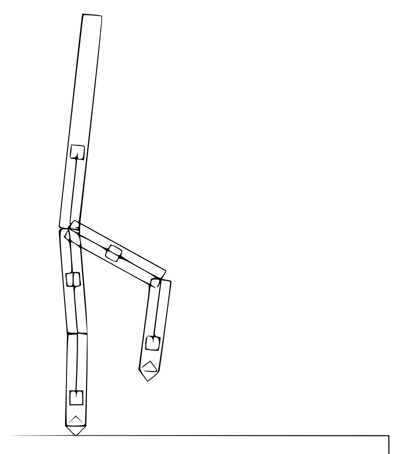


Fig. 2. Biped Taking A Step in the Safe Confines of a Simulated World

During each simulation step, the clever biped decides how to apply torques to its joints to keep it standing vertically and walking. We can complicate the simulation slightly by inserting a common block-like obstacle into the path of the biped. This is where the notion of simulation becomes really powerful. We don't have to make the obstacle out of cardboard or pink construction paper. We can simply define it as a box in the virtual simulation world, instruct the robot to approach it, and watch carefully as the robot encounters the obstacle. See Figure 3.

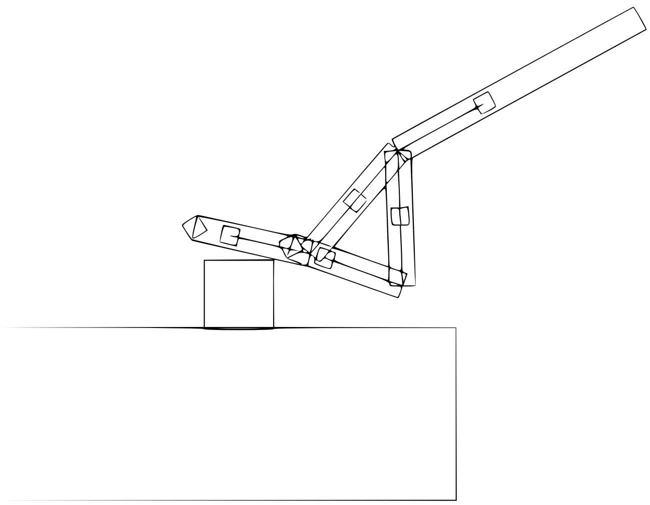


Fig. 3. Biped Interacting with a Box-Like Object

But these two examples illustrate things that can be accomplished in a typical test environment. The test environment concept breaks down when we begin to envision the robot in a typical real-world environment, replete with sinkholes, stairs, and locomotives. Fortunately, with simulation technology we can expose robots to these situations and observe their behavior.

A. Sinkholes

Sinkholes, or *cenotes*, form through the erosion of subsurface bedrock and soil by various subterranean water processes. Sinkholes are actually quite common and can occur in the immediate vicinity of a robot at any time. See Figure 4.

Fig. 4. Sinkhole [2]

We illustrate the behavior of our biped when exposed to the common sinkhole scenario. The sinkhole is simulated by the absence of a walking surface at a distance D from the initial position of the robot. As can be seen in Figure 5, the biped begins walking as in previous examples, using its cleverness to keep itself balanced, and soon encounters the sinkhole.

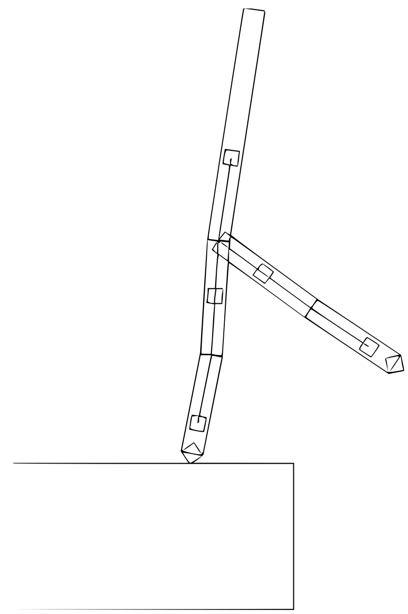


Fig. 5. The Sinkhole Simulator

B. Stairs

Stairs are ubiquitous. Humans use them all the time to move from one floor to another in multi-floored buildings. Despite the presence of elevators in all contemporary multi-floored buildings for accessibility purposes, it is quite common to find humans preferring the use of stairs for reasons of fitness or haste. This popularity makes them both a desired and difficult testing environment for robotics researchers. The simulation paradigm allows us to operate a robot on stair-like structures and observe their operation unhindered by the presence of humans wanting to use the stairs. As the robot interacts with the stairs, robotics researchers have the ability to study the joint torques and external forces on the linkages without damage to themselves or other humans. The simulation can also be repeated with the exact same behavior multiple times to ensure the robot trajectory is fully understood. Using simulation science we can also increase the persistence of each drawn frame to get a frame-by-frame representation of the trajectory of the robot during the encounter. See Figure 6. This would not be possible without the capabilities provided by simulators: it would not be possible for the robot to obtain identical configurations in repeated trials using traditional means.

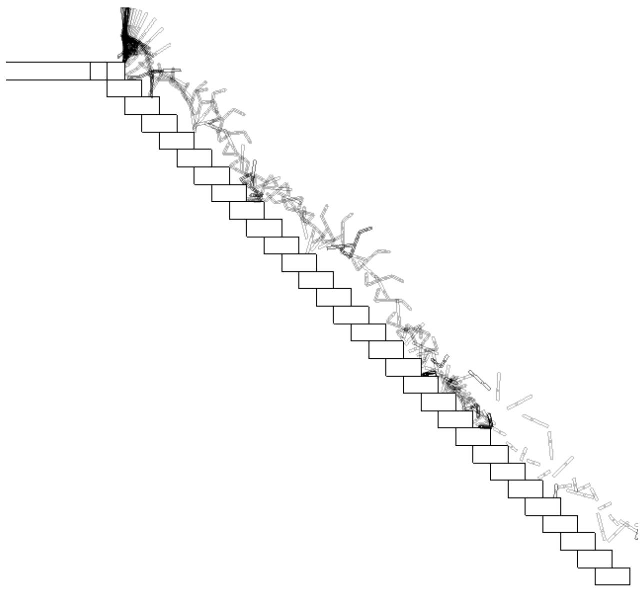


Fig. 6. The Stair Simulator

C. Locomotives

We can also simulate the occurrence of massive objects moving at high speed. Locomotives are common examples of this type of object. Humans frequently do not interact directly with locomotives, as they do with stairs. While this is an advantage for robotics researchers in contrast with stairs, locomotives travel through the Pittsburgh area relatively infrequently and it can be tedious to set up a test environment in the vicinity of a train track and wait for a locomotive. Once again, simulation technology can benefit the robotics researcher. We can simulate the rapid approach of a locomotive and study the direct interactions between the locomotive and the simulated biped. We abstract the locomotive into a large swinging pendulum that approaches the biped silently from the right. See Figure 7.

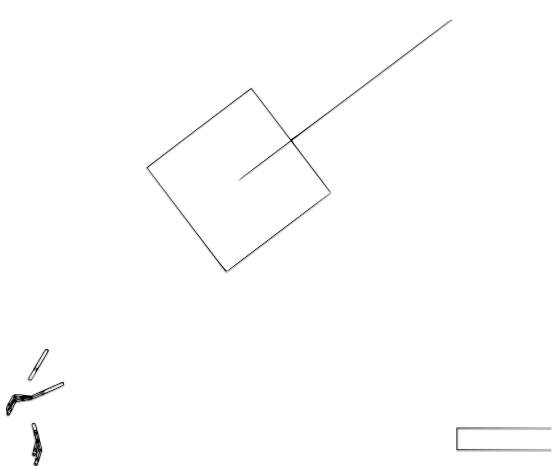


Fig. 7. The Locomotive Simulator

IV. Conclusions

We have described three common scenarios that biped robots similar to the Sarcos will encounter when they make their inevitable transition to society. We illustrate the benefit of simulation in creating virtual test environments for the interaction between robots and various everyday phenomena including:

1) Sinkholes
2) Stairs
3) Locomotives

A. Future Work

We hope to extend our work to multirobot simulation. Soon there will be many robots interacting not only with humans but with one another. With this impending robot explosion it is necessary to have a method of simulating multiple robots interacting with the physical world. Early results on a multirobot simulation are illustrated in Figure 8.

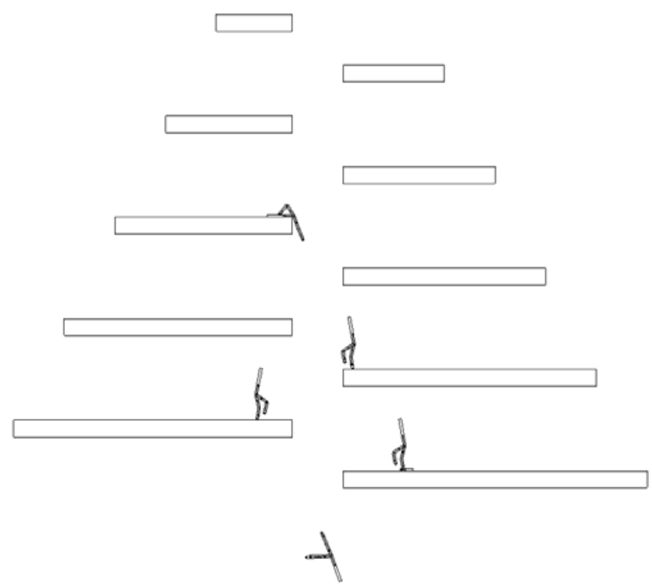


Fig. 8. Multirobot Simulation

References

[1] The Sarcos Robot, `http://www.sarcos.com`.

[2] The Guatemala Sinkhole, `http://www.ordena.com/digg/sinkhole.html`.

Yo Γ Γ!: a Pedagogical Proposal

J. Reed
Carnegie Mellon University
Pittsburgh, Pennsylvania

MC B Combinator
Metropolis University
Monte Carlo

N. Smith*
Former Child Star

March 1, 2008

Abstract

Kids these days: It is well-known that they are lazy, disrespectful, violent, ill-mannered, lethargic, ungrateful, insolent little bastards. Also they listen to terrible music, often walk all over your lawn no matter how many times you shake your fist and yell at them, and, what's worse, they know next to *nothing* about typed λ-calculi. Since all the jerks *do* all day is rot their brains watching "U-Tubes" and indulging their hellish murder fantasies through video games such as "Maze War-craft III" and "Animal Crossing", we figure we might as well get at them early through such debased channels by developing children's programming that teaches them valuable life lessons from theoretical computer science. Better this than fluffy nonsense about "sharing", "self-esteem", and the like, which'll just turn 'em into smug communists in the long run anyhow.

1 Introduction

The state of education these days is deplorable. You pick any random kid out of school and ask them about the Curry-Howard Correspondence [How80], and they'll just stare at you blankly. What are we to do? I'll tell you what: Sink to the level of their rotten, over-sugared, short-attention-span brains. In the following sections, we will outline a proposal for a television program titled "Yo Γ Γ!", which aims to teach the children of today certain indispensible basic facts and concepts about the abstract theory of programming languages, using bright, primary colors, loud noises, and obnoxious repetition.

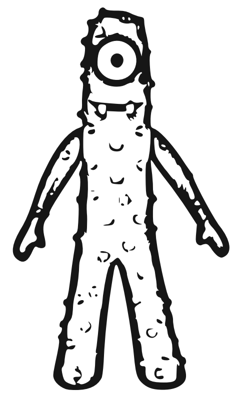


Figure 1: Morph

2 Characters

2.1 Morph

Friendly but impatient. Morph is always dashing from one object to another, flailing his arms about[1], and composing with his clones. This composition is required to be associative, and has an identity at each object. Morph is a very special monster. He is *so* spe-

*My given name is Nathaniel, and I am enthusiastic about writing research papers.

[1]In fact this is the main mode of communication among 'Yo Γ Γ!' monsters.

Figure 2: Bisimulu

cial, he is a special case of himself, provided adequate large cardinal assumptions.

2.2 Bisimulu

Unstably neurotic, albeit cute and fuzzy. Bisimulu will perform actions in an unpredictable order, and tends to whine about whether things are actually true under all possible interleavings. Despite his shortcomings, he seems to get a lot of things done in a small amount of time.

2.3 Funki

Cheerful and outgoing, Funki makes friends easily with compilers, since she has no confusing side-effects, and will politely rearrange her internally datatypes so long as observational equivalence is maintained.

2.4 Proovo

Proovo can use his magic Kripke robot powers to beam special guests into Yo Γ Γ! World[2]. Proovo loves to play with elements of recursively indexed defined datatypes. Proovo obeys certain Proovo Rules.

[2]Warning: guests may be trapped in Yo Γ Γ! World for all eternity if Kripke model doesn't satisfy symmetry.

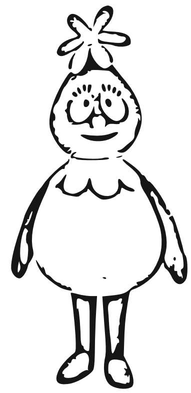


Figure 3: Funki

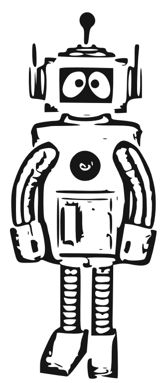


Figure 4: Proovo

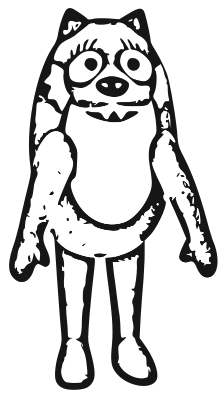


Figure 5: Ty

2.5 Ty

Despite appearances, Ty is not a Ty [MR86]. In fact, Ty a syntactic method for enforcing levels of abstraction in programs [Rey99]. Ty is a mortal enemy of Ducks. Ty is an excellent dancer.

3 Special Guests

We plan to invite leading researchers to participate in special segments such as

- Super Fun Guess the Inductive Metric Time
- Universal Model Construction Arts and Crafts
- Infer the Most General Type!
- Well-Founded Definitions Hide and Seek[3]

4 Set Design

Set theory is wholly inadequate to capture the interrelationships between objects, much less the non-identity isomorphisms between As soon as weak n-category theory [Bae97] is full worked out, the characters will act out their informative storytelling against a background of an ambient ω-topos with a constructive internal language.

[3] Guaranteed to terminate.

References

[Bae97] John C. Baez. An introduction to n-categories. In *Category Theory and Computer Science*, pages 1–33, 1997.

[How80] W.A. Howard. The formulae-as-types notion of constructions. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980.

[MR86] Albert R. Meyer and Mark B. Reinhold. “Type” is not a type. In *POPL ’86: Proceedings of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 287–295, New York, NY, USA, 1986. ACM.

[Rey99] John C. Reynolds. *Theories of programming languages*. Cambridge University Press, New York, NY, USA, 1999.