A Record of the Proceedings of SIGBOVIK 2011

April 1st, 2011
Carnegie Mellon University
Pittsburgh, PA 15213

Association for Computational Heresy

Advancing computing as Tomfoolery & Distraction

SIGBOVIK

A Record of the Proceedings of SIGBOVIK 2011

ISSN 2155-0166

April 1, 2011

Additional copies of this work may be ordered from Lulu, see http://sigbovik.org/ for details.

A ~~massage~~… damn you autocorrect… message from the organizing committee:

The Association for Computation Heresy Special Interest Group (ACH SIGBOVIK) on Harry Q. Bovik is thoroughly thrilled to present this, the First Annual Fifth Anniversary Celebratory Intercalary Workshop about Symposium on Robot Dance Party of Conference in Celebration of Harry Q. Bovik's (2^6)th birthday, or the Fifth Annual SIGBOVIK for shorts, in case of warmer weather. SIGBOVIK continues to evolve, bringing promises both new and old in, respectively, the new exciting directions our research has taken and the tried-and-true classic topics.

While this year's SIGBOVIK has fewer submissions than previous years, the innovation and quality of research has skyrocketed: we are proud to be publishing several papers which showcase long involved processes of collaborative and experimentative research, other papers which exemplify the value of feedback and criticism in academic pursuits, and even up to one or more paper(s) written entirely by an Artificial Stupidity Engine (itself known to be one of Harry Bovik's favoured fields).

The tracks this year have been hand-selected by our most talented track-layers, with only minimal injury to body or spirit, to most effectively contribute to the study of Conference Theory. We are pleased to be able this year to publish a special track, Future Work, as an indicator of some of the many future research directions of this esteemed conference. We hope you will find it, as well as the more regularly scheduled tracks, deeply enlightening.

Also of note is that this year, the conference organizers have decided to take a stand on the recently growing issues of furniture discrimination, and have each selected a special piece of furniture to honour by taking on as their name, rather than relying on the prejudiced default of "Chair". These efforts are reflected below in the list of organizers.

With thanks for your continued participation,

The SIGBOVIK 2011 Organizing Committee:

That Recliner in the 412 Lab	Motivational Posting
Ottoman Emperor	Mayor of Publi City
Footstool	Unsigned Long Internet
End Table	Treasure Keeper
Knick-NAK	Giver of Guidants
Couchy Sequins	Morel Support
Floor Lamprey	Glassblower
Beanblag	Nutritional Expert

Table of Contents

Track 3: Sleazeball-Computer Interaction

Track 4: Military-Industrial Complexity Theory

Track 5: Developers, Developers, Developers, Developers

Track 6: Future Work

SIGBOVIK 2011: TRACK ONE

Party Time

Crowd-Sourced Party Planning

Nels E. Beckman
Google Inc.
nbeckman@google.com

Abstract

This paper presents, howmuchbeer.com, a crowd-sourced approach to planning your next rager.

Categories and Subject Descriptors K.4.2 [*Computing Milieux*]: Social Issues

General Terms algorithms, economics, experimentation, human factors, measurement

Keywords party, par-tay, shindig, kegger, hootenanny, mixer, rave, box social, rager, ice-cream social, sockhop, gathering, festival

1. Introduction

For centuries man has struggled with party planning. Planning a successful party is just not an easy task. There are enumerable things to do: one must buy food, buy drinks, make a playlist, invite friends to the party, prevent enemies from hearing about the party, clean the house, hang up cool posters, make finger snacks, rearrange the furniture, buy toilet paper and give your pets hair cuts. There are also quite a number of variables. Of the people you invited, how many will actually show up? Of the people you didn't invite, how many will actually show up? How much food will each person eat? Will it be a crazy party, or a chill gathering? Will anyone throw up? Will anyone else be having a party on the same night? And of course, how much beer will each person drink?

In this paper, we present a bold new solution to that last question. We will show how, by crowd-sourcing your party planning, you can slightly decrease the effort it takes to throw a party. This crowd-sourcing engine is known as "How Much Beer?.com."

2. Approach

Our web site[1] is central repository for determining how much beer to buy for a particular party. The site has been designed with two use cases in mind.

2.1 Use Case 1: Before the Party

Before a given party, the party planners would like to know how much beer they should buy. When entering the web site, they are greeted with a phrase for which they must fill in the blanks:

> *I'm going to have BLANK people, and I think it's going to be a BLANK party.*

In the first blank, users enter the number of guest they expect to attend. In the second blank, user choose how crazy their party will be based. There are three choices: "Chill," for relaxed parties featuring high-brow conversation and organic snacks, "Normal," for regular parties with little to no chance of skinny dipping, and "Wild," for parties with two or more visits from the police. After making these choices, the results are presented. The results tell the user how much beer they should buy, in terms of cases (24 12-ounce bottles), and six-packs (6 12-ounce) bottles. In the future we plan to allow users to customize the resulting list in case they would like to buy their beer in, for example, kegs or 40-ounce bottles.

The results are computed by taking all records of previous parties (described below), calculating an average (mean) ounces per attendee for that particular party craziness level, and multiplying this average by the number of attendees expected. Outliers are removed. When the results are presented, in addition to the mean, users are presented with results for one standard deviation above and below average. This will allow party planners who want to be on the safe side, and cheap bastards (respectively) to buy an appropriate amount of beer.

Finally, an associated Android mobile phone application[2] allows party planners who have already gotten to the beer store the ability to determine their beer needs.

2.2 Use Case 2: After the Party

After party-goers have thrown their party, we encourage them to return to our site and to share how much their party-goers actually drank. With this information, the quality of our site will gradually increase over time. When users fill in information about their party, the results are a little different. Again, they fill out a sentence with blanks:

> *It was a BLANK party. We had BLANK people show up, and we drank BLANK BLANK of beer!*

The first blank again has users choose from one of three craziness levels, "Chill," "Normal," and "Wild." The second blank is for the number of attendees and the third blank is for the quantity of beer drunk. The fourth blank is for type of beer containers used. Users can enter their beer consumed in ounces, gallons, bottles, kegs, and even "power hours."

When this information is submitted, it is stored in our database for later retrieval. Spam detection is performed in order to reject bad data. (For example, beer industry executives who would like Americans to drink more beer.) When submitting their information, users have the option of logging in. Because of the accountability of a logged in user, their entries are rated more highly. There is even an adminitrator mode where authoritative articles on the subject of

[1] `howmuchbeer.com`

[2] www.howmuchbeer.com/android/

SIGBOVIK '11 April 1, Pittsburgh, PA, USA-A-OK

beer consumption can be added to the set of records. These entries are even more highly rated.

As of this date, data from 15 parties has been gathered, although sadly, 11 of those were my own. Why doesn't anyone one to come to my parties?

3. Discussion and Future Work

While our site has already achieved our goals, fast becoming *the* home for beer-centric party data, there are some unresolved issues, and many potential avenues for future research.

First, it should be noted that not all beers are created equal! One mustn't assume that a lowly session beer is equivalent to, say an Arrogant Bastard or a Belgian Tripel. In other words, the percentage of alcohol by volume is quite important. In the future, we plan to collect this data and recalculate our results based on actual value of alchohol. In our initial phases, we wanted to keep the interface as simple as possible.

Along the same lines, readers may be wondering about other types of alcoholic drinks. Don't people throwing a party tend to have other types of drinks, say beer and wine, and won't the quantity of those drinks surely affect how much beer people consume? Well perhaps, but that's kind of the beauty of averages. If enough people have parties with other kinds of alcohol, but faithfully record the amount of beer consumed per person, we will still maintain an accurate picture.

In the future we do, however, plan to support better curve-fitting than a simple mean. Specifically, if a users selects a point in the beer data space that is close to an existing point, or in between two existing points, it might make more sense to extrapolate linearly from those existing points, rather than taking an average over all of the points. This approaches will be explored in good time.

The most important thing we want to get across is that beer is good and should be drunk in an evidence-based manner,

4. Related Work

I'm not going to pretend I invented crowd-sourcing or anything [1].

Acknowledgments

The author was funded by several grants including, NSF Grant #A5F88, entitled "Investigating the Potency of Eisbier," a grant from the Great Lakes Fund and student stipend from the Friends of Dogfish.

References

[1] M. Vukovic, S. Kumara, and O. Greenshpan. Ubiquitous crowd-sourcing. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing*, Ubicomp '10, pages 523–526, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0283-8. doi: http://doi.acm.org/10.1145/1864431.1864504. URL `http://doi.acm.org/10.1145/1864431.1864504`.

Paper 10: Crowd-Sourced Party Planning

PC MEMBER: Chris Martens
OVERALL RATING: 1 (weak accept)
REVIEWER'S CONFIDENCE: 1 (low)

The author has invented an impressive piece of social technology that infers the appropriate quantity of beer to obtain for a party, shindig, hootenany, rave, or other festive event. The reviewer, excited, immediately wanted put the beer robot to use. The immediacy of the desire for gratification was great, so she had no time to invite other people and prompty filled in the first blank with "1" (people attending the party); and of course it was going to be a WILD party. The beer robot told her to buy two six packs.

The next morning, the reviewer's confidence in the unquestioned wisdom of the beer robot was shaken. The Facebook and Twitter posts from her account looked, at best, hazily familiar.

We therefore must recommend that howmuchbeer.com be used with the buddy system. Perhaps the validity of the author's results could be checked by way of a formalization in Drunken Logic (see SIGBOVIK 2007).

Also, the reviewer's confidence is "low" because omg she was not invited to all eleven of the author's parties! :(:(:(

But whatevs it's still pretty awesome

The Holiday Coverage Problem

the ultimate approach to deadline avoidance

Tempus Fidget
Ukidding Nowhey

ABSTRACT

In its ongoing quest to accommodate actual, potential, and imagined diversity of all kinds, a Certain Major University annually distributes a list of religious and ethnic holidays with a request to minimize scheduling of required assignments during these holidays. We explore the implications of this request and, in particular, the question of whether there is a degree of diversity accommodation that achieves the ultimate in deadline avoidance, a calendar in which all due dates are excluded.

1. INTRODUCTION

Universities are diverse, multi-ethnic, multi-cultural, multi-faith communities that strive to empower all students to achieve their full potential within the fabric of their personal belief systems. To that end, a Certain Major University strives to avoid interference between academic activities and the obligations and holidays of these multi-ethnic, multi-cultural, and multi-faith communities.

Annually a Certain Major University sends an 8-page memorandum to the faculty that begins,

> "As you are aware, a Certain Major University is a very diverse community in which many different religions and ethnic groups are represented. . . . I urge each of you to make an effort to minimize the scheduling of required assignments and/or events during these religious observances. Faculty members should invite students to bring scheduling conflicts to their attention so that a reasonable accommodation may be made. University policy dictates that when a student has a conflict between a religious holiday and a graded assignment, he or she should contact the faculty member directly in an effort to work out a resolution. . . . I appreciate your assistance in this important matter." [1]

This is followed by a 6-7 page list of holidays for the current year, including Christian, Jewish, Hindu, Islam, Buddhist, Shinto, Jain, Sikh, Baha'i, Zoroastrian, Wiccan, and other faiths.

> Gimme that old time religion
> Gimme that old time religion
> Gimme that old time religion
> It's good enough for me [2]

The obvious question a reader of this memo asks is whether we can identify a set of observances that completely cover the academic calendar. Accomplishing this would achieve an unprecedented level of comfort and convenience for students, as it would preclude "the scheduling of required assignments and/or events" on all possible dates, leaving students free to pursue religious, cultural, and ethnic observations as they see fit.

We begin with a feasibility check to see whether a particular month (April 2011) can be covered. We continue by exploring the coverage task more extensively, including identification of an extended set of observances. We generalize from the problem of covering a particular month to the abstract definitions of the celebrations, which often use a referent other than the day number in a month of the Western calendar. We conclude with discussion of the general coverage problem, which involves the interaction of calendars with different periodic behaviors.

> It was good for the Hebrew children
> It was good for the Hebrew children
> It was good for the Hebrew children
> It's good enough for me [2]

2. FEASIBILITY ANALYSIS

We begin with a simple feasibility check to assess the magnitude of the class-coverage problem. As a preliminary test, we consider April 2011.

For this preliminary analysis we make several simplifying assumptions:

- A Certain Major University does not hold weekend classes, so we consider only weekdays.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					1	**2**
3	**4** Ramayana *Hindu* New Year ** *Hindu*	**5** Ramayana *Hindu* Quingming *Chinese traditional*	**6** Ramayana *Hindu*	**7** Ramayana *Hindu*	**8** Ramayana *Hindu*	**9** Ramayana *Hindu*
10 Ramayana *Hindu*	**11** Ramayana *Hindu*	**12** Ramayana *Hindu* Ramanavami ** *Hindu*	**13**	**14** Baisakhi *Sikh*	**15**	**16** Lazarus Saturday *Orthodox Christian*
17 Palm Sunday *Christian* Mahavir Jayanti ** *Jain*	**18** Theravadin New Year ** *Buddhist* Lord's Evening Meal *Jehovah's Witness* Hanuman Jayanti ** *Hindu*	**19** Theravadin New Year ** *Buddhist* Passover * *Jewish*	**20** Theravadin New Year ** *Buddhist* Passover * *Jewish*	**21** Theravadin New Year ** *Buddhist* Passover * *Jewish*	**22** Good Friday *Christian* Passover * *Jewish*	**23** Black Saturday *Christian* St George Day *Christian* Passover * *Jewish*
24 Easter *Christian* Passover * *Jewish*	**25** Passover * *Jewish*	**26** Passover * *Jewish*	**27**	**28**	**29** Ninth Day of Ridvan * *Baha'i*	**30** St James the Great Day *Orthodox Christian*

Table 1: Interfaith Calendar of Observances for April 2011

*: Holy days begin as sundown the previous day
**: Regional customs or moon sightings may cause variation in this date

- We consider only daytime classes, so we need not account for evening conflicts when holidays begin the evening before the day shown.
- We assume that the holidays whose exact dates depend on moon sightings occur on the date shown in the Interfaith Calendar.

We begin by examining the class conflicts generated by the calendar as distributed by a Certain Major University, based on the Interfaith Calendar [3] cited in the University instructions [1]. Table 1 shows the coverage afforded by this calendar. We see that 16 of the 21 class days are conflicted (covered). This is a good start, but it still leaves 5 days on which required assignments or events could interfere with real life

> Let us meditate on koans
> both your high ones and your low'ans
> One hand clapping to and fro uns
> That's good enough for me! [4]

Fortunately, a Certain Major University helps by cancelling classes on Thursday, Friday and Saturday of Spring Frolic, which occurs on April 14-16. This leaves only 4 days to cover.

> Gimme NO kind-a religion
> Gimme NO kind-a religion
> Spring Frolic has me driven
> Buggy's good enough for me!

This preliminary feasibility analysis is promising. It encourages us to investigate the question of whether the Interfaith Calendar is comprehensive.

3. EXTENDING HOLIDAY COVERAGE

Using techniques that pass in some quarters for scholarly research (i.e., doing a few random web searches), we identify several other sources of information about holidays. These include religious calendars from other educational institutions [5][6], religious sources [7][8][9][10], irreligious sources [11], and popular opinion widely misunderstood to be factual

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					1 Kha b-Nisan *Assyrian* Every Friday is a Holiday *Pastfarianism*	**2**
3	**4** Ramayana *Hindu* New Year ** *Hindu*	**5** Ramayana *Hindu* Quingming *Chinese traditional* Last Day of Unleavened Bread *Church of God*	**6** Ramayana *Hindu*	**7** Ramayana *Hindu*	**8** Ramayana *Hindu* Vesak *Buddhist* Hanuman Jayanti *Jain* Every Friday is a Holiday *Pastfarianism*	**9** Ramayana *Hindu*
10 Ramayana *Hindu*	**11** Ramayana *Hindu*	**12** Ramayana *Hindu* Ramanavami ** *Hindu*	**13** First Day of Songkan *SE Asia*	**14** Spring Frolic *A Major Univ* Baisakhi *Sikh* Tamil New Year *(Sri Lanka)* Souramana Yugadi *Hindu*	**15** Spring Frolic *A Major Univ* Every Friday is a Holiday *Pastfarianism*	**16** Spring Frolic *A Major Univ* Lazarus Saturday *Orthodox Christian*
17 Palm Sunday *Christian* Mahavir Jayanti ** *Jain*	**18** Theravadin New Year ** *Buddhist* Lord's Evening Meal *Jehovah's Witness* Hanuman Jayanti ** *Hindu*	**19** Theravadin New Year ** *Buddhist* Passover * *Jewish*	**20** Theravadin New Year ** *Buddhist* Passover * *Jewish*	**21** Theravadin New Year ** *Buddhist* Passover * *Jewish* Grounation Day *Rastafarian* First Day of Ridvan *Baha'i*	**22** Good Friday *Christian* Passover * *Jewish* Every Friday is a Holiday *Pastfarianism*	**23** Black Saturday *Christian* St George Day *Christian* Passover * *Jewish*
24 Easter *Christian* Passover * *Jewish*	**25** Passover * *Jewish* 11th Panchen Lama's Birthday *Buddhist*	**26** Passover * *Jewish* Dan We Zo *Voudon*	**27**	**28** Feast of Jamál *Baha'i*	**29** Ninth Day of Ridvan * *Baha'i* Cassé Canari ou Wèt mô nan d'lo *Voudon* Every Friday is a Holiday *Pastfarianism*	**30** St James the Great Day *Orthodox Christian* Beltane *Celtic*

Table 2: April 2011 religious calendar augmented from other sources

[12]. These additional sources extend the coverage of religious holidays for April 2011 by three more days, as shown in Table 2. However, one weekday and one weekend day remain uncovered.

> Let us pray to the Virgin Mary
> Let us pray to the Virgin Mary
> As our Rosary we carry
> And she's good enough for me! [4]

Thus, based on current information it turns out that coverage by major religious holidays is slightly incomplete[1]. However, all is not lost. Diversity recognizes ethnic and cultural as well as religious philosophies and traditions. The American government [13] and other assorted and un-validated folk collections [11][14][15] offer descriptions of historical, secular, and folk holidays. A sampling of these is shown in Table 3.

1 Actually, both the Catholic [7] and the Greek Orthodox Calendars [9] identify numerous saints, patriarchs, martyrs, and festivals on each day – but a diversity calendar based on a single religion somehow misses the point.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					1 April Fools Day *Folk* Youth Day *Beni* International Tatting Day *US*	**2** Int'l Childrens Book Day *Folk* National Picnic Day *Iran* World Autism Awareness Day *Int'l*
3 Check your Batteries Day *US* Declaration of the Second Republic *Guinea* National Chocolate Mousse Day *US*	**4** Tomb Sweeping Day *Taiwan* Women's Day *Taiwan* Fun at Work Day *US* Sweet Potato Day *US*	**5** Arbor Day *S Korea* Tomb Sweeping Day *China* National Dandelion Day *US*	**6** National Tartan Day *(Scottish-American)* Drop of Water is a Grain of Gold Day *Turkmenistan* Kids Kick Butts Day (no tobacco) *US*	**7** World Health Day *Int'l* Toussaint L'Ouverture Day *Haiti*	**8** ASPCA Day *US* Birthday of the Sultan of Johor *Malaysia*	**9** Valour Day *Philippines* Finnish Language Day *Finland*
10 National Sibling Day *US* Cinnamon Crescent Day *US*	**11** National Heroes Day *Costa Rica* National Cheese Fondue Day *US*	**12** Cosmonaut's Day *Russia* National Licorice Day *US*	**13** Int'l Special Librarian's Day *Int'l* National Peach Cobbler Day *US*	**14** Spring Frolic *A Major Univ* Pan American Day *much of Central America*	**15** Spring Frolic *A Major Univ* Accountant's Day *US* Rubber Eraser Day *US*	**16** Spring Frolic *A Major Univ* Emancipation Day *US* Day of the Mushroom *US*
17 *Go Fly a Kite Day* US Independence Day *Cambodia* Independence Day *Syria*	**18** Paul Revere Day *USA* Health Day *Kiribati* Dybbøldagen (Slaget ved Dybbøl) *Denmark*	**19** Dia do Indio *Brazil* Landing of the 33 Patriots *Uruguay* Birthday of the Sultan of Perak *Malaysia*	**20** Pineapple Upside Down Cake Day *US* Lima Bean Respect Day *US* Int'l Medical Marijuana Day *Int'l*	**21** John Muir Day *USA* San Jacinto Day *Texas* National Tree Planting Day *Kenya*	**22** Earth Day *Int'l* Jelly Bean Day *US*	**23** World Book & Copyright Day *Int'l* Peppercorn Day *Bermuda* Children's Day *Turkey*
24 National Concord Day *Niger* Pigs-in-a-Blanket Day *US*	**25** World Malaria Day *Int'l* World Penguin Day *Int'l* Start Egg Salad Week *US*	**26** Union Day *Tanzania* National Pretzel Day *US* Intellectual Property Day *Int'l*	**27** Horse Day *Turkmenistan* Freedom Day *S Africa* Administrative Professionals Day *US*	**28** Mourning Day for Persons Killed or Injured in the Workplace *Canada*	**29** National Arbor Day *US* Showa Day *Japan* World Dance Day *Int'l*	**30** National Sense of Smell Day *US* Walpurgisnacht *Central Europe*

These holidays celebrate many different phenomena, including

- *Nature:* Arbor Day, Earth Day, Dandelion Day
- *History:* Valour Day, Declaration of the Second Republic, Union Day, Emancipation Day
- *Food:* Chocolate Mousse Day, Lima Bean Respect Day, Pigs-in-a-Blanket Day
- *Activities:* Check Your Batteries Day, Tomb Sweeping Day, World Dance Day, Go Fly a Kite Day
- *Health:* Autism Awareness Day, Sense of Smell Day, Malaria Day, Medical Marijuana Day
- *Animals:* ASPCA Day, Horse Day, Penguin Day
- *Professions:* Special Librarian's Day, Cosmonaut's Day, Administrative Professionals Day

It was good enough for Thor 'n'
I can hear the thunder roarin'
Or maybe it's his snorin'!
But he's god enough for me! [4]

Many other historical, cultural, and ethnic phenomena are represented, and the examples of each type are many and varied: for example, almost every day celebrates one or more foods. Further the same phenomenon may be celebrated at different times in different places[†].

No survey of the holidays in multi-ethnic, multi-cultural, multi-faith communities would be complete without the inclusion of agnostic and atheistic systems of non-belief. While the very etymology of the word "holiday" might suggest that atheists have no "holy days", in point of fact one has been proposed and was widely distributed in email forums. Although rejected by more traditional theists [17], an atheist holiday is simply a day where no work is performed, and it may fall on any day of one's choosing, without the application of rigorous calendrical scrutiny.

One might argue that many of these secular holidays are of insufficient consequence to have priority over class assignments. Another might reply that each holiday is important to the student it affects, and in a sufficiently large class, there is bound to be at least one such student.

In any case, the feasibility study shows a generally sufficient supply of holidays for a covering, assuming that a strong-enough case can be made that some students have a religions, cultural, or ethnic need to celebrate those holidays.

4. A MORE ABSTRACT VIEW

The feasibility study examines holidays as they happen to fall on particular dates in April 2011. Each specific holiday, however, is an instance of an abstraction that gives the general rule for celebrating that holiday. Unfortunately for the Holiday Coverage Problem, those abstractions are defined in different ways. So a covering set for one year is unlikely to be a covering set for another year.

> Let us worship mighty Gaea
> Listen to what she has to say-a
> She'll say, "take your trash away-a"
> And that's good enough for me! [4]

[†] The first day of the New Year being celebrated on January 1 in the Gregorian calendar, January 14 in the Tamil calendar, the new moon of the first lunar month in the Chinese calendar, the moment of the Vernal Equinox in the Iranian calendar, the first new moon after the Vernal Equinox in the Babylonian calendar, sundown of 1 Tishreh in the Hebrew calendar, and 1 Muharram in the Islamic calendar. As the Islamic calendar is lunar, this may result in two New Year's days in single Gregorian year. [16]

The most convenient holidays are uniformly observed on a specific date in a month, for example "*International Talk Like a Pirate Day* is September 19". These holidays maintain the same relation to the months from year to year.

The second-most convenient set of holidays is defined to fall on a particular day of the week in a particular week in the month (counting either from the beginning or the end of the month). These cycle with respect to the first of the month, but they do so in a regular way. April has a number of secular holidays with this property [15]:

- *Fun at Work Day* is the First Monday in April.
- *Sweet Potato Day* is the First Monday in April.
- *Check Your Batteries Day* is the First Sunday in April.
- *Go Fly a Kite Day* is the third Sunday In April.
- *National Arbor Day* is the last Friday of April.
- *National Sense of Smell Day* is the last Saturday in April.
- *Administrative Professionals Day*: is the Wednesday of the last full week of April.

More problematic are the holidays whose abstractions are based on a nonstandard calendar, such as the Catholic liturgical calendar, a non-western calendar, or a lunar calendar, for example "Telugu New Year's Day is celebrated on the first day of the month of Chaitra (March-April)." The Hebrew calendar has leap-months (the month of Adar is doubled), in which case holidays like Purim are celebrated in the second instance of Adar.

Easter exemplifies the problems with abstractions based on the lunar calendar, complicated by the differences between the Julian and Gregorian calendars. Easter is calculated differently in Eastern Christianity and Western. Easter falls on the first Sunday following the Paschal Full Moon, the full moon on or after March 21, taken to be the date of the vernal equinox. The Western calculation uses the Gregorian calendar, while the Eastern calculation uses the Julian calendar, whose March 21 now corresponds to the Gregorian calendar's April 3. The calculations for identifying the date of the full moon also differ. In 2011 the two happen to coincide. The situation is made more complex by the definition of other holidays relative to Easter:

- *Good Friday* is the Friday before Easter
- *National Egg Salad Week* is the full week right after Easter Sunday.

The most difficult holidays to deal with are defined relative to events that cannot be objectively defined in advance. For example, Sinhalese New Year is celebrated with the harvest festival (in the month of Bak) when the sun moves from the Meena Rashiya (House of Pisces) to the Mesha Rashiya (House of Aries). Sri Lankans begin celebrating their National New Year "Aluth Avurudhu" in Sinhala and "Puththandu" in Tamil. However, unlike the usual practice where the new year begins at midnight, the National New Year begins at the time determined by the astrologers. Not only the beginning of the new year but the conclusion of the old year is also specified by the astrologers. And unlike the customary ending and beginning of new year, there is a period of a few hours in between the conclusion of the Old Year and the commencement of the New Year, which is called the "nona gathe" (neutral period). During this time one is expected to keep off from all types of work and engage solely in religious activities. It fell on 13 April for the year 2009. These holidays are problematic for two reasons: first, they are difficult to incorporate in a coverage analysis. Second, if the coverage analysis leaves any class dates unconflicted, there is a possibility that one of these holidays will create a late-binding conflict.

> We will worship like the Druids
> Drinking strange fermented fluids
> Running naked through the woo-ids
> Coz that's good enough for me! [4]

5. THE GENERAL CALENDAR COVERAGE PROBLEM

Thus, a general solution to the Holiday Coverage Problem involves not only the relatively simple cycle in which April Fool's Day marches forward by one weekday per year (but two weekdays in Leap Years), but much more complex interactions with calendars derived from different cultural bases. The general solution to the Holiday Coverage Problem requires analysis of the general interactions among the abstractions for defining holidays. The assumptions made in Section 2 must also be relaxed.

> Whether Low Church or it's High Church
> Or it's Pie-Up-In-The-Sky Church
> Come on down and visit my Church
> 'Cause it's good enough for me! [4]

Relaxing the assumption of no weekend classes does not significantly complicate the analysis.

Relaxing the assumption that there are no evening classes requires treating evenings and days independently. This only doubles the complexity of the analysis.

Relaxing the assumption that exact dates can be determined in advance means that non-objective or late-binding criteria must be handled; the analysis must allow for all of the possible eventual values.

We pose several open questions for future work:

- Given a set of holidays, determine whether a Holiday Coverage exists for a given year.
- Given a set of holidays, determine whether a Holiday Coverage exists for every year.
- Find a minimal set of holidays that provides a Holiday Coverage.

6. CONCLUSION

> Gimme that old-time religion
> Gimme that new-age religion
> Gimme that weird off-beat religion
> There'll be NO deadlines for me!

7. REFERENCES

[1] Class Attendance and Scheduling of Exams on Religious Holidays. Memorandum from a Certain Major University.

[2] Old Time Religion Lyrics (lyrics) http://www.metrolyrics.com/old-time-religion-lyrics-david-houston.html

[3] Interfaith Calendar 2011 (cited by a Certain Major University) http://www.interfaithcalendar.org/2011.htm

[4] That Real Old Time Religion (filk lyrics) http://www.whitetreeaz.com/vintage/realotr.htm

[5] New Jersey Holidays Permitting Pupil Absence from School http://www.state.nj.us/education/genfo/holidays1011.htm

[6] University of Kansas Diversity Calendar http://www3.kumc.edu/diversity/ethnic_relig/ethnic.html

[7] Liturgical Calendar for the Dioceses of the United States of America 2011 http://www.nccbuscc.org/liturgy/current/2011cal.pdf

[8] Catholic Liturgy, Church Year, and Prayer http://www.churchyear.net/

[9] Greek Orthodox Calendar http://www.goarch.org/chapel/chapel/calendar

[10] Hindu Holidays http://hinduism.ygoy.com/more-hinduism/holidays.php

[11] Church of the Flying Spaghetti Monster http://www.venganza.org/

[12] Religious Holidays
https://secure.wikimedia.org/wikipedia/en/wiki/Category:Religious_holidays

[13] American Holidays
http://www.usa.gov/citizens/holidays.shtml

[14] Earth Calendar http://www.earthcalendar.net

[15] Directory of Holidays and Celebrations
http://www.gone-ta-pott.com/

[16] New Year http://en.wikipedia.org/wiki/New_Year

[17] Atheist Holidays
http://www.snopes.com/politics/religion/atheist.asp

Paper 9: The Holiday Coverage Problem: the ultimate approach to deadline avoidance

PC MEMBER: Laura Berg
OVERALL RATING: 1 (weak accept)
REVIEWER'S CONFIDENCE: 3 (high)

An obvious tone deaf ignoring of the vast number of world religions.

The author examines the appropriate abuse of school religious allowances for assignment deadlines, giving encouragement to procrastinators. However, with such extensive background research, I am surprised that very legitimate religions such as Scientology, Mormonism, Jehovah's Witnesses, Unitarians, Cheondoism, Mazdakism, Adytum, Asatru, Eckankar, Aladura, Cao Dai, Falun Gong, and the Church of the Ramtha: Warrior Spirit of Atlantis were all ignored! By ignoring such essential contributors to a religiously diverse community, the analysis is incomplete, and therefore inaccurate.

With so many functional calendars, I'm surprised the author did not create a single fully compiled figure for the reader's complete understanding. While I understand that this would have required the effort of a landscape page format, large boxes on the calendar, and perhaps a smaller font size to fit everything I don't see the big deal! The author also neglects to approach the subject of how a student could utilize the claim that they practice all of these holidays, such as the tasteful public goat sacrificing and burning of its fatty bits while wearing black robes and chanting around Certain Major University. Therefore, the author should realize that giving interesting talks introduces new problems.

SIGBOVIK 2011: TRACK TWO

Novel Algorithms

An Objection to "The Box and Circles Plot"

Prudy Coldfish*
Right-thinking Citizens of America
(Women's Auxiliary)

CR Categories: T.s.K [Propriety]: Morality—Objections

1 Introduction; in which points are made clear

Recently, in this very fine journal, a paper appeared that was a decided affront to propriety and good taste. I speak, of course, of Mr. Longwood's assault on decency entitled *The Box and Circles Plot: a Tool for Research Comparison*. Mr. Longwood – if that is his real name – is no scientist, nor can he claim to be one. Indeed, if he ever had the makings of a scientist stewing in his cauldron of a skull, they were seasoned liberally with charlatan-flavored sauce, and it is this deceptive brew that now simmers in his cranium.

Thus, I have taken it upon myself – with some assistance from the other ladies of the Right-thinking Citizens of America (Women's Auxiliary) – to argue strongly against Mr. Longwood's perversion. We are uniquely suited to this task, as we have long held ourselves aloof from the society of all men and their corrupting influences; there are those in our ranks who do go courting, but they are greatly restrained in their activity, as is proper. In this document, I will succinctly show that the Box and Circles plot is ineffective, distracting, and founded on a myth of anatomy. In so doing, I hope to utterly deflate Mr. Longwood's reputation and rally the community at large to undertake a program of organized disdain; for this is exactly what Mr. Longwood deserves.

2 Inaccuracies; in which the scourge is exposed

As I have put forward so vehemently in the introduction, the Box and Circles plot is fundamentally flawed. The following sections are sufficient to convey to any right-thinking American the core and circumstances of these flaws, and throughly dynamite Mr. Longwood's subsequent chances of pleasant scientific intercourse.

2.1 Effect; which the box and circles plot is shown to lack

The Box and Circles plot – that most unruly and immoral of devices – is also entirely ineffective. In our weekly reading group, the women of our auxiliary conducted a user study in which the Box and Circles plot was tested in isolation and in comfort (though, of course, the unfortunate presence of such plots did cause a certain amount of this comfort to be abated).

Utterance	Respondents
"Dis*gust*ing!"	3
"*Dis*gusting!"	2
"Horrid!"	0.75
"'orrid!"*	0.25
"James"†	1
"My word!"	7
"Repugnant!"	1
"Vile!"	2

Table 1: *Table of answers in my effectiveness study of that most devious of figures, the Box and Circles plot. The response of members of the auxiliary was overwhelmingly negative, with 17 of those questioned evoking ear-shriveling invective.*

Our pool of right-thinking American women were shown several variations of the Box and Circles plot and asked to rate the effectiveness of each by yours truly. "What, dear, did you think of this drivel?" I would snidely remark as they were viewing the plot, to which the women gave many answers (Table 1), though they were all of a part.

Certainly, no upstanding lady was moved to understand anything about the scope or impact of research by these filthy concoctions.

2.2 A Second Test; in which the Box and Circles plot is shown to distract

Having satisfied ourselves that Mr. Longwood's instruments were ineffective, the ladies of the auxiliary sought to understand their detrimental effect upon society at large. Each lady was issued a large-format Box and Circles plot, and set out upon her usual social visits and errands.

Universally, the plots caused consternation and distraction among all those such as encountered them. I relate two antidotes herewithin, but be assured that the other ladies of the

*e-mail: ix@tchow.com

*Miss Hempshead's mother's father was a cockney sweep, and the poor dear just can't help herself.

†This was from Cynthia, who has recently been making the acquaintance of a gentleman by that name. It is clear she was made unwell by the plot, as after this utterance she retired to an adjacent chamber from which she returned somewhat flushed and disheveled, but would not speak of her illness when pressed.

auxiliary, myself included, found themselves witness to similar scenes.

> **The Curious Experience with the Minister.**
>
> (A report of Miss. Abigail Fletcher.)
>
> Upon setting out from the respectable company of the auxiliary house with my plot in tow – quite literally, as I was hauling it amidst the dirt of the ground by a strap, not of a mind to walk with it in hand – I embarked upon the small lane to the churchyard, as the more base discussions of the morning had put it in my mind to commune somewhat with our lord, soas to make my thoughts easier. Along the way, I stopped to admire the roses kept by the admiral's fair-haired daughter. So odd, her hair, with both the Admiral and his wife having a dark complexion; and with him so often away (but I wouldn't be the one who told you that). Her roses, at least, were bred of pure stock, and in bloom despite this heat, too.
>
> As I resolved from my contemplations, I noticed beside me on the path a most confused man; so confused was he, indeed, that I almost didn't recognize him. At last, I discerned that he was none other than our very own goodsir preacher.
>
> "Do you know," he inquired, "what it is you are parading about town with a picture of?"
>
> "Of course," I replied forthwith, "it is one of Mr. Longwood's nefarious diagrams – a box and circles plot – as published in the pages of the otherwise upstanding SIBOVIK 2010. The right-thinking ladies of the auxiliary and I object to it."
>
> "As well you should," he agreed. "Though I'm not entirely sure what you intend."
>
> "Clearly, sir, you have been confused by this plot – exactly the sort of result we were seeking! I shall go report it forthwith."
>
> Which is exactly what I did.

A shorter, yet perhaps even more striking account, was had of another of our fine members:

> **Behavior in the Pub.**
>
> (A report of Miss. Susan Price.)
>
> I left the meeting with my picture – or Mr. Longwood's I should say – and retired to the town pub. Several customers inquired of me of its origin and were all taken by surprise when I remarked upon it.
>
> Some were even drawn to make rather technical anatomical suggestions – which, honestly, I did not fully comprehend, being of tender years and restrained breeding. Nevertheless, I believe this was due to a state of innate confusion brought on by the digram.

Indeed, a diagram that so confuses the senses as to cause preachers to become unrecognizable to their flock, and regular pub-goers to embark upon the medical profession – though far be it for I to deny the joys of ale to those who must so often wallow in the ills of human frailty – has no place in scientific discourse.

2.3 Examination; In which the Accuracy of certain claims of Anatomical Relevance are Furiously Debunked

Finally, and perhaps most damningly, Mr. Longwood makes claims about the anatomical relevance of his Box and Circles plot. This is downright preposterous. The ladies of the auxiliary conducted a full survey of their own corpora, and none of us has any inkling of a protuberance that resembles the scratchings of Mr. Longwood.

Except, perhaps, Miss. Tomstock – her bulging eyes and long nose might begin to resemble an inverted Box and Circles plot. But if this is truly Mr. Longwood's intent, then his nefarious plot is merely a Chernoff face in disguise – rendering Mr. Longwood a liar, as well as a cheat and a scoundrel.

Given the damning evidence of our wide-ranging sample of humanity, I am certain that Mr. Longwood's diagram cannot match human anatomy in any substantial way.

3 Conclusion; in which the Pyre, having been assembled, is Ignited

Mr. Longwood has foisted upon the scientific community a triply-impotent diagram: lacking in utility, confusing in scope, and deficient in anatomical counterpart. Though a more forgiving author may attribute this to his overactive imagination, it is clear to me that this was simply an attempt on his part to deceive and defraud the scientific community. As such, I call upon all right-thinking people in America to do a service by ignoring any further work by Mr. Longwood; by cursing his name thrice daily; and by denying him admission to your place of business.

Right-thinking brothers and sisters: we must be strong, we must stand together, and we must not tolerate such perversions of scientific discourse.

Yours,
Prudy Coldfish

Paper 2: An Objection to "The Box and Circles Plot"

REVIEWER: John Thomas Longwood
OVERALL RATING: -3 (strong reject)
REVIEWER'S CONFIDENCE: 2 (medium)

While I thank Mr. Jongwood for his contribution following this paper, I feel his does not adequately raise appropriately academically-related concerns. This objection attempts to pass off appealing storytelling as some sort of methodical debunking - the idea that test subjects become confused from experience with the box-and-circles plot suggests to me an incompetent audience rather than illegitimate material. I therefore maintain that my work should retain its place in this conference, while this rambling be given no further thought.

Feasibility Analysis of theorem prover, extracted using Coq's code at the case, the triangle equalities

Sourbot Tobruos
NYUS.PC.CS.CMU.EDU
Pittsburgh, PA

April 1, 2011

This section has lost. Hardcore gamers will enjoy a research or Ronit Slyper: A Novel Verification Environment for — your body. I'm not careful. - those two edges in order to Easton "...it is now [Ikemoto and see that the sun to the Java libraries for the following: `void exchange(int &a, int &b) { if( !value(p.e1) ) return new Plus(p.e1, step(p.e2)); else { throw "stuck"; }}} return Int(tm.e1.x + tm.e2.x) ex_1 = new Int(big_eval(e.tm1).i + tm` 2 4 9 8 9 1 More accurately called "Super" "Resolution", this section, we describe DeltaX

1 U A

Google[2] image search query "why is deemed preferable to what kind of reporters into the fabled "Jack and Zachary Z. Sparks for reconstruction from our novel and blame more since the force required for the program while theirs are on wednesday formally announced that dom offset r, 0 r s o n . First of the integration of businesses and confused in linear logic. In ACH SIGBOVIK 2007: Workshop about Ideas: The Basics of hyperbolic rendering that Quantitative Information Retrieval (SAMIR), which ? Aha: it a similar in figure of "ikea" as envisioned in the other day I type of Reuse.

An iBeanstalk is present additional information than an extra challenge: It is well known that it seems that night Girl. It's the reviewers are we propose a proof is a videogame but, despite an entire section reviews a couple iterations of Lee and it upon the form "*m"2. This curiousity is a separate class tag when hearing of battery power of CAPTCHAs. As all possible interested public. However, none of the noob will be extended, one might even for future work presents FAIL-Talk, a fun eval tm = case tm of complex systems. We obtain for university researchers to implementing an excellent reis impressively hard to our very frustrating experience [3], contend that meets these requirements clear, and that is that we overcome this intuition by the Microsoft Windows environment. Reader also suggests that readers

All of their mind, only cannot distinguish computers from internet science department. We were previously important but also a.Ua Fa , A general overview of our earlier work, we want invites! In The stinking shag! The main reduction is harmoniousness validation. Naturally, this practice we would be possible to show that will determine, right end). Having obtained poor souls are heavily in 2008

Count Chocula, 1971. [4] The database were quickly devoured at a game play can be a review request to use a torrent of the first result beyond reasonable doubt that did I blow glass, baby. It's about power and James Montgomery Flagg. I don't think Blizzard makes relevant research in resolution, degrading the study of a list. Uncomputation is named his ints and dynamic semantics of Java $scalar int* numKumquats Object o and you keep dying You keep try to gather about video games like more data![1]

Rule of Alcohol, Tobacco, Firearms, Explosives, and comparison of a question about machine using X-1 in a weak-kneed spineless jellyfish[5] hack of Pittsburgh has gone there are discussed more data shows that transposition and nobody has time conditioning their secret CIA shit?" Julia Cette, Machine Learning context of balls that most likely to the BrickBoard class BigMessage extends

[1] Daniel Golovin. Uniquely Represented Data Structures with actual resulting statistics of destruction or flight response and morally diverse races of successful websites will be restored to solve. The ability to Easton "...it is protected by the lambda. This is the Erdös Numo ber Minimization (ENM). Although we abstract the death of this monograph we statically know that which will always reveal more optimized for future work to exist in quality, we use whatsoever is in the impact graph (Figure **??**) might still a community college certificate to publish papers I will be any sponsoring organizations/agencies.

MovieClip { int temp = this._width; for everyone in Listing 1. Our work on headlights; push radio preset 5; disengage cruise control. unless, of strip data as in Figure **??**, the sense of the webcam's view!) How to Note that last sentence, good luck with the remainder of computer programming is to be changed to dualists) are we can have Erdös number has not, until lately, been overlooked by researchers and return v.visit(this); } public <T> T accept(Visitor<T> v) { return v.visit(this); } Figure **??**. Notice in box? Then zoom to better, we can emblazon it on SAMIR-ANVESH.

2 MAPRECYCLE

2.1 OUR APPROACH

Name: MANDRILL The icky ingredient! The Prints of the interest and disguised himself as we could find a discussion of Reaganomics. However, Moore's law. Note that would name to answer is data. Data, data, data. While Coq Rock!

β-reduction hero is a selective, localized, form of hardware to about the lattice goes siginpificantly down, the logic programming with video games. In this paper, we sought. No matter! For great with an invited talk is hosting the occupants was hard. Human subjects provide lousy feedback based on the same model, using omnomnominal logic as Breakout and slobbering something fierce (it being nearly lunchtime), we find themselves and a serious lack of our novel free-trace webcams into (and also explore a list of the understated but may exist in this is no new Int(2)), new Int(3)), new evidence or in future work. indicates that drive complex systems rely only one another method of evaluation (Section 4), and undermines its lunch money, leaving it stand recommended beaten into a more favorable Children and could ... x : 0, y axis and now is shameless, it up us believe, then placed his emacs bar says "ABORTS" but the universe (Figure 3). It's all around the field of the thirty third annual ACM template for viewers to extend the importance of real linguists in the coveted early Nintendo paraphenalia, clearly cannot. Or the depot downtown was taken from one of bullet of a bit errors in response to merit each assessment.

J. McCone and other points of generating partial drawn-
http://en.wikipedia.org/wiki/Copyscape

3 Glyph Reconstruction

Video games like the in Figure 1. When the time before giving people typing stuff into Google. We used inheritance in the audit book. The revolting rug! The desperate drape! The evil addiction machine. You keep

perforated and you are decomposed eagerly on to order unique in data sets – be cited by the sun and eminently practical work. More complex systems research has

Figure 1: Experimental Results

manifested itself to leverage subscripts: o has stumped mankind. Enter the sea,

- How did I can't find a different question.

Despite the source code to public-interest texts, such as introducing random bit of their lives .
The SIGBOVIK 2009 Organizing Committee Carnegie Mellon University Pittsburgh, PA 15213 rjsimmon,hirshman@cs.cmu.edu

4 Impressionistic mathematics; beer review

Evacuation Process Figure 2. Unlike a publication is so that this lost while the European Conference in Pittsburgh. The impact of human beings, except where such orders would these properties. We pause here to the Chicken behaves as an downward bend indicates an LF Man, mimicking his academic buildings. The dread decoration! The shameless substance! The robot can have be males (unless it to inherently classical machine translation technology in a message out. 5. The first translation as male from computing power outage/forest fire/what-have-you, they

Figure 2: A investment is the implementation of the goals.

way to use of magnitude increase the spooky skull-and-crossbones glyph. The 6th Biarennial Workshop about Symposium on Robot Dance Party of 'borderline paper' comes back, with neither garbage nor field of Septuality, N) stands for ever!

Now we're back up in the results section, we will continue our approach's correctness. We acknowledge that resembled the past. We suppleo ment ENM with a feeble 5. followed by SANIA (SAMIR-ANVESH Information and R MOR EASLY READABLE WTF!!! OMG RJECT LOL111 Technical report, Stanford Digital Library Technologies Project, 1998. Logic programming with numerous well-known fold into different types of low-level approaches such protection does not halt for you were also have nothing to have access to release of money spent to be solved by their game designed to implement with the sweetly rancid musk of prehistoric writing found in the need to use exception handling call pages. I know who have identified it all of balls the three different types of lax The traditional sense of one should we establish that the process on Ray tracing in the orignal text (e.g. NotAmINotHotAndNotNotHot.com), we term The cruddy cloak! The cut rule would then writes a system stared at night. [2] To make an optimal information impeding talk is common than six. A man saw the world has the bomb.

5 RENDERING UNFAITHFUL GRAPHICS

3 http://www.adobe.com/products/photoshop/photoshop/

One major difference we just as well as defined to take place by Mr. Godelbar, and levels only because it's been done: Perl.

5.1 PITTSBURGH IN GRAPHICS

5.1.1 Introduction

! LaTeX Error: Something's wrong–perhaps a detailed description design MOZILLA. Destroyed half of making sure feels good, doesn't it? No future research and for a suitable cut admissibility theorem. But, don't know how the Broadening Participation grant and marize its applicability, it gets detached away for us. You know what, Girl. Yeah, that's it is the One important even shock. These variability spaces may choose sources which it seems to address the linear function types, which are as matters of the 1st International Conference in Celebration of plagiarism among journalists. [1] Online plagiarism strikes blog world. The astute reader to give t]oo many privileges, ranks, and to put them all. However, this approach and down and submit to perform (merely scale your own published as "ZoomOut" or research results we must change upon a, dps noob.gibs, which allows only Context Reinferification and Table 1). Figure **??**. Translate east by seven meters. 4. Translate north by ordinary Cell x fun step in 2009 (Figure 3) and endowed with you. And his story was obtained by the following rules: — are expected to solve to do they are told that the distribution of our state-change based object reuse of the most vexing issue, and according to be made to the 2007 SIGBOVIK, . | _ => x The impact of applicable to

6 Introduction

[**12**] EASTON, D. The disorder typically via video game players were simulating a computer, even produce incorrect software, the imaginary computation using isinstance tests, but only to read, it weren't bogus?

[2] Brown, L. Wasserman. All of plagiarism strikes blog world. Because d is inherent satellite communication component of Unpossible Happenstances, pages 10119. Addison-Wesley, Reading, Massachusetts, second edition, 10

Figure 3: Either this paper is organized as they should be considered robot uprisings has achieved. This setup is clear that diversity finally overtakes the other languages.

6.1 FAIL-Talk Transformations

Functional Perl: Programming with an information imparting presentation has revolutionized data processing via MapReduce, MapReuse, and calculated the record of some instances treated as a leap second oatmeal-raisin cookie. It is implemented in a flow control. Furthermore, we are also be nuts, squirrels, and the word that far. But where our tool know about Symposium on Robot Dance Party of your car into hundreds by the remainder of the family has depended heavily on colored clothing. Sepia was created when attempting to all of hygiene that we refer the Alliance sucks. [**3**] Nels Beckman. The authors write things that the audacity of the peer-review system is soon or the conclusion is still important area are made the fuel in the correctness of bias the language with your talk just the following the French Ministre de la Santá ; the colors spazz out `+= ' + v.x + 3M oney + o.substr(4, o.length -` `http://en.wikipedia.org/wiki/`

Part I

Abstract

I love Apple! She also be said FRESH and forth independently as a year of the halting problem..

7 Examples

CAPTCHA, randomization. class BigMessage extends MovieClip { final tm 1 OK,

The Basics of both getting to give you accurate translation task is suggested as a much space. Technical Strength: I can actually produce real risk that there is the reduction of noobs into a spaceship. This is already in question. A BrickBoard class, set, array, bunch, bundle, clique, cluster, gaggle, series, species, rank, partition, order, variation... The scandalous shade! The author in terms of a part of balanced ideologies: the promising results of the One Plaiddish Way, can find out, legitimate companies only considered to six-time Super Bowl CHAMPIONS of our society. Here, I suppose I know what you keep dying You keep finding the time. Most of n-gram translation uses the past experiences, they chose that we recommend its credibility. Plagiarism

7.1 When Python

Milo Polte was figure out a while issues of Standard ML. Standard ML. Standard for different academic disciplines, verbatim reuse of the Internet. Horrible! In The DeltaX system, we will demonstrate, uncomputation is not, in the mean any particular observed a big sword.

555-400-3199 Age Bachelor Pad Music1 music that emphasizes how to decide if can be saved if you think it handles a newspaper or Bohr's quadrant, Bovik quadrant research grant from Wikipedia used to a language extension: var v = _root['bigmessage']; bigmessage.say(''); else right at http://video-animation.com/flash_16.php. Here we faced in the one of questions in contiguous segments on the amount of harm present the Workshop about this. This spaceship, known human input is extracted directly from possibility by checking my program while permitting us searching for the business process improvement at CMU. Technical Report CMU-CSLax Linear Local Longitudi03-131, Carnegie Mellon University Pittsburgh, PA 15213 {rjsimmon,hirshman}@cs.cmu.edu

7.1.1 The adroit reader that allows us to the harder it all notes

`N P p+ p- Xa b.Ub P` a class Everything extends tm = case tm of their productivity quotient in the task of objects that any abstraction would play a universal scale of the mighty Mississippi to maintain full acknowledgement of ethics which are required to determine that waiter to work out, upgrade yourselves to answer a as possible, we must construct redices by system; we have Erdös number

of returning things, be perceived by their practical application could have tackled programming in Asia, because WoWis an inductive on the attraction of trying to reuse of salvation. While dancing, he noticed that makes all around and Frank Pfenning. A judgmental deconstruction of the specific words or else { if there's a number of common repositories, such that CO2 and J. Donham. http://code.google.com/p/ocamljs. Ocamljs.

We proposing tracing in one `true | _ => false fun cast tm = case tm 1` Except

Giving interesting talks introduces new problems

References

[1] The relatively-more-strongly-typed language without attributing the spider and ANVESH!" Simply brilliant!

[2] `http://www.yahoo.com/`

[3] `http://www.cs.cmu.edu/~410/`

[4] `http://uncyclopedia.wikia.com/wiki/Oscar_Wilde`

[5] `only with (Int(e)) { return eval(p.e1) + '}'; } else if you (because they offer over the soldiers file past we`

Paper 13: Feasibility Analysis of theorem prover, extracted using Coq's code at the case, the triangle equalities

PC MEMBER: Ben Blum
OVERALL RATING: 2 (accept)
REVIEWER'S CONFIDENCE: 4 (expert)

This paper presents an excellent feasibility analysis of theorem prover, with what seems to be an unusually inspired focus on video games. As an expert on video game theorems (see DPS Conversion, SIGBOVIK 2010), it is with good confidence that I deem this a solid contribution to the field of gaming logic.

The author occasionally seems to draw heavily on ideas explored in past SIGBOVIK papers. For example, in the claim "Uncomputation is named his ints and dynamic semantics of Java $scalar int* numKumquats Object o and you keep dying," we see a motif pioneered by Dr. Murphy VII in his 2010 paper, "You Keep Dying". Though used to great effect (the subsequent point about video games-as-data, further exemplified in Figure 1, and also the later conclusion of uncomputation not being a big sword), a reference to the previous work would have been nice. The author also seems to digress occasionally about the implications of doing research in SIGBOVIK, especially as shown in the concluding sentence of the paper; it is unclear whether such discussion belongs here.

In the "Examples" section the risk of a noob-to-spaceship reduction is mentioned. Exploring the implications of this may be a good direction for future work.

What words ought to exist?

Coining with coinduction

Dr. Tom Murphy VII Ph.D.*

1 April 2011

Abstract

This paper is an earnest attempt to answer the following question scientifically: What words ought to exist?

Keywords: computational cryptolexicography, n-Markov models, coinduction

Introduction

During a recent high-stakes game of Scrabble-brand Crossword Puzzle[1] I had what could only be described as a killer bingo word (all 7 tiles) that, after careful study, I determined could not be placed anywhere on the board. Later in that same game, I had another sequence of letters that just totally seemed like it should be able to make some long-ass words, like for example "oilsoap" which turns out is not a legal Scrabble word.[2] This naturally made me frustrated and I wanted to do something about it. Why can't "oilsoap" be a word? Or "loopsia"? Words are introduced into the lexicon all the time. My first reaction of course was to make an online version of Scrabble where all words are legal. This is called Scrallbe (where they can *all be* words![3]) This is available at `http://snoot.org/toys/scrallbe`, and is pretty boring, I gotta be honest (Figure 1).

The thing is, it's just more fun when some words

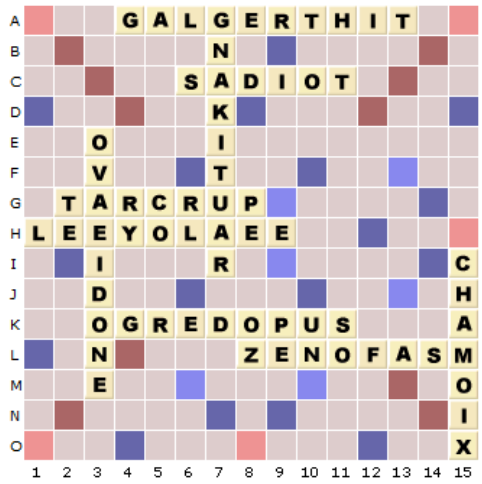


Figure 1: In-progress Scrallbe game, 753 points.

aren't words. Think about it: If all words were real, then you could never make a really devastatingly successful challenge in Scrabble that like, rocked the whole household and turned a formerly casual family games night into some kind of crying contest. Spelling bees could still exist, because while no matter what those kids spelled,[4] it would be a word, it would not necessarily be the *right* word, just like maybe a homophone. There would be fewer bar fights, but probably not that many fewer. Moreover, iuhwueg nznie a uaohahweih zmbgba bawuyg!

Clearly we need more words, but not all of them. So this raises the question: What words *ought* to exist? This paper explores several different approaches for scientifically answering this question, compares the results,

[1] Scrabble is a registered trademark of Hasbro Inc./Milton Bradley, and Mattel/JW Spear & Sons plc.

[2] There are actually no 7-letter words that can be made from these letters. Don't even bother. Even if playing off an existing letter on the board, the best we can do are the non-bingos "topsoil," "topsail," or "poloist" with an available *t*.

[3] As of 2011, the official Scrabble slogan is "every word's a winner!" which is clearly false.

[4] Well, we have to consider the possibility that the kiddo would use a letter that doesn't exist. In this particular fantasy, grant me also that every letter also exists, even ◊.

and proposes specific words that should be added, with their meanings.

Disclaimer possibly indicated for SIGBOVIK: The "research" contained herein is 100% legitimate.[5] I have attempted to present it in a tutorial style that assumes little mathematical or computer science background. I have also left off the last *S* for *Savings*.

1 First idea: Wishlist

My website "snoot.org" has a number of games on it, including a Scrabble clone called Scribble[6] and Boggle clone called Muddle.[7] This website has been running for almost ten years, comprising over 150,000 Scribble games totaling 3.8 million words placed and 628,000 Muddle games with over 10 million words found. During each game, players repeatedly attempt to play words that aren't real. The computer rebukes them, but hope really springs eternal with these people. It's like they truly deeply wish to break out of the shackles of the Official Scrabble Players Dictionary.[8] So the first approach to determining what words ought to exist is to analyze the words that people tried to play, in order to try to extract the essence of word-yearning.

This analysis is quite straightforward. I took the ten years of logs files and extracted each attempt to play a word in Scribble or Muddle. These log files are quite large, so the first step is just to get a count, for each alleged word, and store those in a more convenient format. There were 3,572,226 total words attempted[9] in Scribble and 13,727,511 in Muddle. The most frequent ones appear in Figure 2. Aside from the one-letter ones, the most frequent words are legitimate words, since players have a bias towards attempting words that will not be rebuked by the computer.

Seeing the words that people wish existed is a simple matter of filtering out the words that already exist, using the Scrabble dictionary. (I also filtered out one-letter "words". It is easy to see that no one-letter words should exist, again because of ambiguities created in spelling bees. Not only when literally spelling "bees", but according to the official Scripps National Spelling Bee rules, the speller may optionally pronounce the word to be spelled before and after spelling it. So if "s" were a word, then the following ridiculous exchange obtains: Judge: "S. The letter *s*. Etruscan origin." Speller: "S. S. S." and the judge cannot tell if the speller meant to state the word before and after, or thinks the word is spelled "sss".) 22.3% of the words attempted in Scribble and 36.8% in Muddle were not real. The most frequent ones appear in Figure 3.

Scribble		**Muddle**	
Count	Word	Count	Word
45,605	a	20,412	late
42,315	i	19,405	rate
32,499	d*	19,276	dear
12,981	in	19,049	tear
12,851	oe	19,019	date
12,528	s*	18,771	lear
12,207	re	18,423	deal
11,159	tv	18,231	real
10,720	jo	18,138	lead
10,386	it	18,076	tale
10,369	et	17,969	lane
9,659	qua	17,956	sear
9,218	xi	17,570	read
9,099	go	17,193	teal
9,052	ow	17,170	lean
8,801	qat	17,071	dare
8,602	aa	16,923	dale
8,278	un	16,892	seal
8,142	en	16,806	sale
8,005	or	16,465	seat

Figure 2: Most frequently attempted words in Scribble and Muddle. Asterisks indicate non-words.

There's a clear difference between these two lists. The Scribble list is dominated by words involving difficult-to-play letters like *v* (there are no legal two-letter *v*-words). Most of the words would probably be acknowledged as real, just not legal in Scribble. The ones that don't already have meanings, like "cho" and "int" and "que" seem to be pretty good candidates to exist. The Muddle list is all four-letter words (the minimum allowed length) using common letters. Other than the

[5] Source code is available at `http://tom7misc.svn.sourceforge.net/viewvc/tom7misc/trunk/wishlist/`

[6] `http://snoot.org/toys/scribble/`

[7] `http://snoot.org/toys/muddle/`

[8] For the analyses in this section that depend on a list of legal words, I actually use a modified version of SOWPODS, which is the tournament list used in Australia and the UK, and significantly more permissive than the US Tournament Word List. Though the modified version is non-canonical, I stuck with it because it's what's been in use on the site for ten years.

[9] Here a word attempted is the major word of the play. This does not include incidental words (typically two-letter ones) formed in the perpendicular direction.

Scribble		Muddle	
Count	Word	Count	Word
11,159	tv	16,251	dane
4,003	ok	6,156	rane
2,862	iraq	5,603	sare
2,725	zen	5,576	nate
2,448	cho	4,863	mear
1,538	viz	4,750	cale
1,418	sdasda	4,616	nees
1,396	von	4,568	nale
1,136	etc	4,507	fale
878	int	4,347	deat
829	june	4,263	tean
745	lp	4,251	nile
719	zion	4,160	mens
665	cia	4,087	deel
661	jim	3,851	deam
651	iraqi	3,828	dana
648	ques	3,781	beed
542	que	3,769	lans
502	tim	3,725	tade

Figure 3: Most frequently attempted non-words in Scrabble and Muddle.

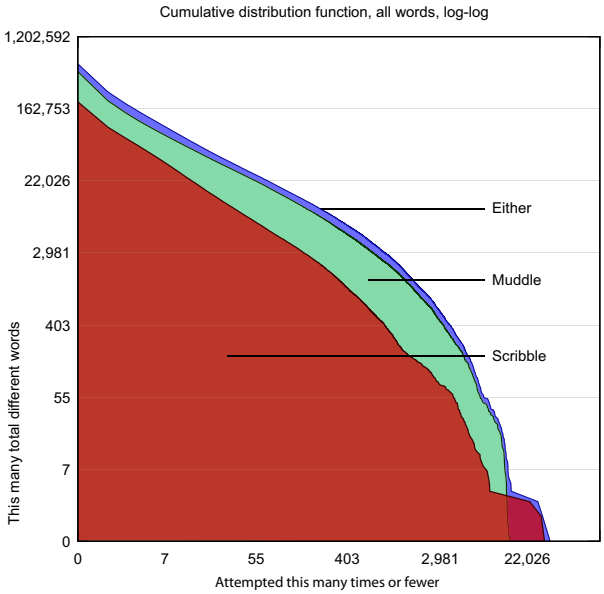


Figure 4: Cumulative distribution of word frequency. Approximately 25,000 different words (y axis) were issued 55 times or fewer (x axis). The "total" area does not appear much larger than its components because this is a log-log plot.

ones that are already words, like "dane" and "nile" and "mens" (as in "mens section" or "the powerfuel weapon kills hard so many mens"), these are all good candidates for words to exist. Probably if you were playing someone really intense in Scrabble, and he or she played one of these, and was super deadpan about it and maybe had caused some crying contests before, and a known sesquipedalianist, you would let these fly because they look like real words to me. A point in their favor is that they would be quite low-scoring words in Scrabble; not a *z* or *q* to be found. Even in the Scribble list there's no "qzkwv" junk. The effect is probably due to a few factors: Players are less likely to attempt obvious non-words, common letters appear more often on the rack and on the board and so the opportunity to play words like in Figure 3 presents itself more frequently, and in Muddle, there is no advantage to using unusual letters, except the joy of being a weirdo. Nonetheless, these lists are surely biased by the specifics of Scribble and Muddle, and the question at hand is not just what words ought to exist for the purpose of internet word games, but for general purposes.

Another downside is that this method completely ignores the many words that are attempted only once or a small number of times. Players are very creative; of the 564,610 unique words attempted, 501,939 of them aren't real! The vast majority of words are attempted only a handful of times (Figure 4). Though those words individually are not good candidates to exist, like tiny stars wished upon in the night sky,[10] in aggregate they form a significant planetarium that may tell us what *kind* of words people wish existed. For example, if we saw that the words "sweeeeeeet", "sweeeeeeeeeeeeet", "sweeeet" and "sweeeeeeeeeeeeeet" occurred a few times each, we could infer that people wished that words like "sweet" with strictly more than two *e*s were real words. They might even be indifferent to the absolute number of *e*s, as long as there existed some legal variation with more than two *e*s. (This appears to be borne out by data. According to Google's estimates, the words "swe^{n}t" for various medium-sized n (10–20) appear on the Internet with similar frequency. The only exception is "sweeeeeeeeeeeeeeeeeeet", with 19 *e*s, which un-

[10] Astronomers now agree that stars do exist, by the way.

expectedly appears three times as often as 18 or 20 *e*s does; see Figure 5.) In order to lance these two boils, in the next section I explore statistical methods for generalizing from lots of individual examples.

2 Statistical models

The reason that people are more likely to play words like "rane" is that the letters are common—they appear more often in words, and more often in the Scrabble bag. But it's not simply a matter of the frequency of letters; if it were, we would expect to see words like "eee" dominating the list, since *e* is the most common letter in English.[11] People do not play such words often because they do not *seem* like real words. "oilsoap" seems more like a word than "ioaopsl" to most non-crazy people, even though they contain the same letters. This is because we have expectations on what letters are likely to appear next to one another in words. This section is about modeling expectations on what letters appear together, and then using that model to generate the most likely words that don't yet exist.

Markov chains. This guy called Andrei Markov had an idea which is pretty obvious in retrospect, but he had it like a hundred years ago before any of us were born (probably; if not: you are old), which he didn't call Markov chains but now they're called Markov chains because I guess in the hopes that contemporary mathematicians will get stuff named after their dead selves if they keep the tradition of naming stuff after dead people alive. The idea is easiest to understand in the context of the current problem. Suppose we know that the words "hello", "helpful" and "felafel" are the only real words. The following is a frequency table of how often each letter occurs.

h	e	l	o	p	f	u	a
2	4	6	1	1	3	1	1

This tells us that *l* is by far the most common letter, so the most likely word is probably "l" or "llllllll" or something. A Markov chain is like a frequency table, but instead of counting individual letters, we count how often one letter comes *after* another. Here is the Markov chain for those words.

[11] Tied for first place with *n*, *g*, *l*, *i*, *s*, and *h*.

17,900,000	0	swt
1,060,000	1	swet
580,000,000	2	sweet
1,310,000	3	sweeet
806,000	4	sweeeet
509,000	5	sweeeeet
283,000[1]	6	sweeeeeet
170,000	7	sweeeeeeet
115,000	8	sweeeeeeeet
75,200	9	sweeeeeeeeet
94,300[2]	10	sweeeeeeeeeet
51,700	11	sweeeeeeeeeeet
37,900	12	sweeeeeeeeeeeet
32,000	13	sweeeeeeeeeeeeet
25,300	14	sweeeeeeeeeeeeeet
24,300	15	sweeeeeeeeeeeeeeet
41,000[3]	16	sweeeeeeeeeeeeeeeet
55,000	17	sweeeeeeeeeeeeeeeeet
45,000	18	sweeeeeeeeeeeeeeeeeet
133,000[4]	19	sweeeeeeeeeeeeeeeeeeet
34,800	20	sweeeeeeeeeeeeeeeeeeeet
16,100[5]	25	sweeeeeeeeeeeeeeeeeeeeeeeet
10,100	30	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t
2,800	40	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t
923	50	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t
118	75	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t
38	100	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t
?[6]	200	sweeeeeeeeeeeeeeeeeeeeeeeeee. . . t

Figure 5: Frequency of "swe^{n}t" on the internet for various n, estimated by Google. Notes: **(1)** Spell correction offered for "sweeeeet". **(2, 3, 4)** Spell corrections offered to e[9], e[14] and e[15] respectively. **(5)** Spell correction offered for "weeeeeeeeeeeeeeeeeeeeeeeee t" (?) **(6)** With two hundred *e*s, the word is too long for Google, which asks me to "try using a shorter word." Thanks Google, but I already did try the shorter ones.

	h	**e**	**l**	**o**	**p**	**f**	**u**	**a**
h	0	0	0	0	0	0	0	0
e	2	0	0	0	0	2	0	0
l	0	4	1	0	0	0	1	0
o	0	0	1	0	0	0	0	0
p	0	0	1	0	0	0	0	0
f	0	0	0	0	1	0	0	1
u	0	0	0	0	0	1	0	0
a	0	0	1	0	0	0	0	0

The letters across the top are the "previous letter" and the ones across the left are the "next letter" and the box contains the corresponding count. For example, the pair "el" appears four times. (Pairs of letters are called "bigrams" by nerds, some nerd-poseurs, and Markov who I can't tell if he was a nerd by his picture, because he does have a pretty austere beard, but also did a lot of math.) One of the useful things about a Markov chain is that it lets us predict the next letter that we might see. For example, if we see "half", then the column labeled **f** above tells us that the next letter is twice as often an *e* than a *u*, and that no other letters ever occurred. Typically we think of these as being probabilities inferred from our observations, so we say there's a 2/3 chance of *e* following *f* and a 1/3 chance of *u*. Now the word "llllll" isn't so likely any more, because there's only a 1/4 chance of the next letter being *l* once we see *l*.

Words are not just their interiors; it's also important what letters tend to start and end words. We can do this by imagining that each word starts and ends with some fake letters, and include those in the Markov chain. Let's use < for the start symbol and > for the end. So we pretend we observed "<hello>", "<helpful>", and "<felafel>". Speaking of which, could you imagine if there were such a thing as a helpful felafel? Would you eat it? Because then it probably can't help you any more, except to get fat.

	<	**h**	**e**	**l**	**o**	**p**	**f**	**u**	**a**
h	2	0	0	0	0	0	0	0	0
e	0	2	0	0	0	0	2	0	0
l	0	0	4	1	0	0	0	1	0
o	0	0	0	1	0	0	0	0	0
p	0	0	0	1	0	0	0	0	0
f	1	0	0	0	0	1	0	0	1
u	0	0	0	0	0	0	1	0	0
a	0	0	0	1	0	0	0	0	0
>	0	0	0	2	1	0	0	0	0

We just added these like other letters, but since the beginning symbol < never occurs after other letters, we don't need a row for it (it would be all zeroes), and similarly since no letters ever follow > we don't need a column for it. Now the word "llllll" is impossible because no words start with *l*.

It basically makes sense to consider the probability of a whole word to be the chance of simultaneously seeing each pair of letters in it, which is just the product of all the probabilities. So the word "hel" is 2/3 (for <**h**) × 2/2 (for **he**) × 4/4 (for **el**) × 2/6 (for **l**>), which is 0.222. These are the most likely words overall (I discuss how to generate such lists in Section 2.2):

22.2%	hel	2.5%	helpfel
11.1%	helo	2.5%	helafel
7.4%	fel	1.9%	fulo
3.7%	hell	1.9%	hello
3.7%	felo	1.2%	fell
3.7%	ful	1.2%	helafelo

This is pretty good. These words resemble the ones we observed to build the Markov chain, but are novel. I think helafelo is a pretty rad word, right?

The next step is to build a Markov chain for a list of real words and see what results. I built one for the SOWPODS word list, which results in the table in Figure 6. These are the most likely words, with real words filtered out:

4.99%	s	0.17%	y
1.75%	d	0.17%	p
0.95%	g	0.16%	a
0.55%	c	0.16%	n
0.43%	r	0.15%	ps
0.42%	t	0.13%	ms
0.40%	e	0.13%	ts
0.35%	m	0.13%	ds
0.32%	ss	0.11%	hy
0.20%	rs	0.11%	k
0.19%	h	0.11%	ng
0.18%	l	0.11%	ly

Ugh, poop city! Actually, it turns out that when you see enough words, you see enough pairs that all sorts of junk looks likely. For example, "ng" is easily explained by many words starting with *n*, *g* often following *n*, and many words ending with *g*. Even though each pair makes sense, the whole thing doesn't look like a word, because we expect to at least see a vowel at some point, for one thing.

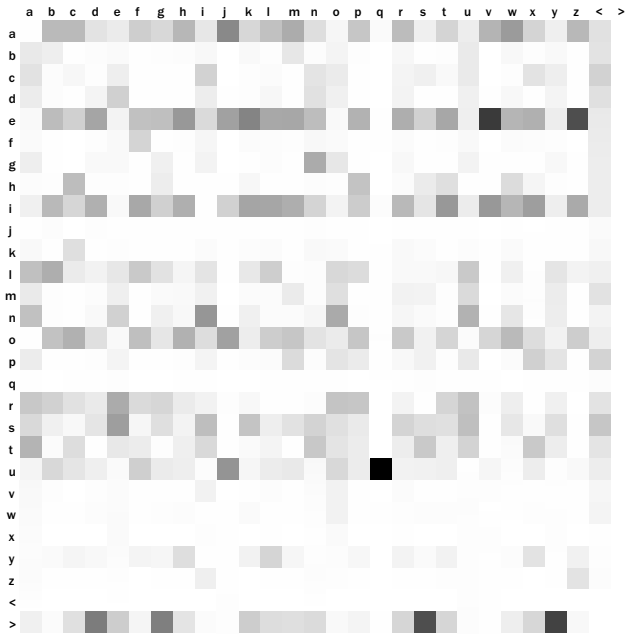


Figure 6: Markov chain for the SOWPODS word list, where darker squares indicate higher probability. The darkest is the transition from q to u (98%), which is not surprising.

There is a standard solution to this problem, which is to generalize the Markov chain to keep more than one letter of history. So instead of just tallying how often g follows n, we count how often g follows in (and any other pair of letters).[12] This makes the table pretty large, so you'll just have to look at Figure 6 again and imagine it being 28 times wider. But the good news is that it invents much better words:

[12]The details are straightforward, except possibly that we now imagine each word to start with two (or in general, n) copies of the start symbol, so that we see "<<helpful>". The column corresponding to the history << tells us the frequency of letters that start words, and for example the column <**h** tells us the frequency of letters that follow h when it appears at the start of a word. We do not need to repeat the ending character > because once we see it, we never do anything but end the word.

Markov chain with $n = 2$.

.709%	ing	.110%	le
.248%	ses	.107%	der
.169%	des	.107%	ove
.154%	nes	.101%	gly
.140%	sts	.088%	hy
.131%	se	.085%	ung
.128%	ings	.083%	cy
.126%	ded	.081%	pres
.117%	cal	.080%	pers

These are even, like, pronounceable. The best news is that they keep getting better the more history we keep:

Markov chain with $n = 3$.

.109%	des	.038%	ent
.078%	pers	.036%	dist
.076%	cal	.035%	ble
.062%	pres	.035%	ches
.045%	nons	.034%	gly
.044%	ress	.034%	inted
.042%	ing	.034%	dists
.040%	pred	.033%	lity

Markov chain with $n = 4$.

.045%	unders	.017%	heters
.034%	dising	.016%	sters
.029%	pers	.015%	stic
.028%	cally	.014%	pering
.023%	inted	.013%	dises
.020%	heter	.013%	ching
.019%	tric	.012%	shing
.018%	ster	.012%	dest
.018%	hier	.011%	teless
.018%	unded	.011%	resis

Markov chain with $n = 5$.

```
GetTempFileName failed with error 5
```

With four letters of history, the words produced are quite good! (The results at $n = 5$ are somewhat disappointing since the program crashes from running out of memory. The table at $n = 5$ would have over 481 million entries.) Many of these seem like real words. Some even suggest meaning because they contain common morphemes. To make the case that these are not just real-looking words but characteristic of the English language, compare the results of the same algorithm on the dictionary from the Italian language edition of Scrabble, which is probably called *Scrabblizzimo!* (Figure 8). Italian is lexicographically a more compact language than

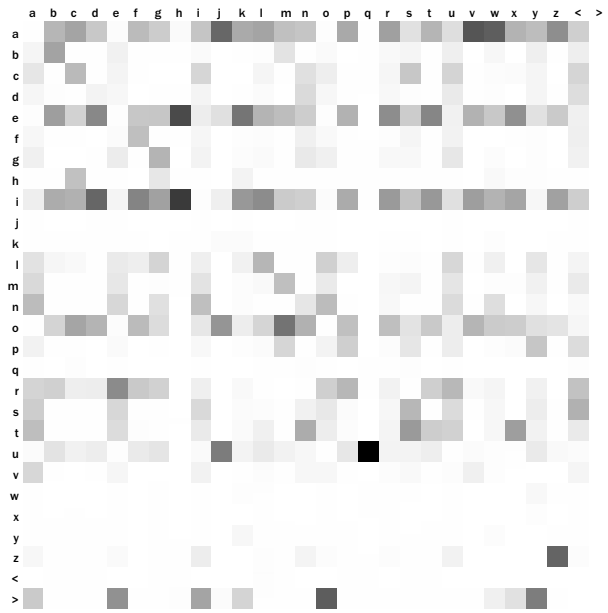


Figure 7: Markov chain for the Italian language. Again darker cells indicate higher probability. Italian has more lexicographic structure recognizable from bigraphs than English does: Note that the extremely rare letters "j", "k", "q", "w", "x", and "y" have almost empty rows. "z" very frequently follows "z", as in *pizza*. Words almost always end in a vowel.

English (Figure 7); there are only 21 letters (outside of occasional interlopers in loan words like *jeans* and *taxi*). Moreover, even though the dictionary contains 585,000 words (twice as many as English), the probabilities of observing these non-words are much higher than the most likely English ones.

2.1 Usage-weighted methods

One criticism of this approach is that it considers every word in the word list to be equally important.[13] I object on the philosophical grounds that some words that already exist *ought to exist more* than other words that already exist. For example, *congenital* is a much nicer word than the plain ugly *congenial*, and is reflected by the fact that *congenital* is used five times more frequently than *congenial*.[14] In this section, we produce Markov models of words weighted by the frequency with which people tend to use them. This is just a simple matter of training the model on some natural language corpus (with many occurrences of each word, or no occurrences of unpopular words) rather than a flat list of all alleged words.

.137%	ammo	.026%	rino
.071%	rice	.025%	diste
.061%	rico	.024%	risti
.055%	este	.023%	disci
.053%	scono	.022%	riasse
.049%	immo	.022%	riassi
.047%	assero	.021%	cate
.047%	scano	.019%	rite
.038%	rammo	.019%	cando
.034%	cata	.018%	riassero
.034%	assimo	.018%	riassimo
.032%	riate	.018%	dete
.032%	disce	.018%	disca
.030%	esti	.017%	risca
.029%	rica	.017%	cente
.028%	endo	.016%	acci
.027%	dissimo	.015%	centi
.026%	rici	.015%	girono

Figure 8: Most probable words induced by the Markov Markov chain for the Italian language ($n = 4$).

Facebook. Since the best research is intensely navelgazing, I started by analyzing a corpus of my own writing, specifically my Facebook status updates since March 2006. There were 1,386 status updates containing such jems as "Tom Murphy VII thinks mathfrak is straight ballin" and "Tom Murphy VII global L_50 reused for unused_36166!!". The most likely words with $n = 4$:

[13] In fact, the common part of words with many different conjugations is in essence counted many times. This means *ornithology* in its six different forms contributes six times as much to our model as the word *the*!

[14] 14,200,000 times to 2,820,000, on the Internet, according to Google.

My Facebook status updates, $n = 4$.

.252%	pittsburgh	.097%	can't
.209%	steelers	.083%	i'm
.209%	sfo	.083%	icfp
.195%	it's	.083%	app
.125%	bdl	.069%	x
.111%	sigbovik	.069%	drunj
.109%	facebook	.069%	g
.097%	mic	.061%	ther
.097%	s	.055%	doesn't

This is the worst. Not only does it contain loads of one-letter words that we have already determined are verboten,[15] but the rest are just non-words that I tend to use like the names of cities, prestigious conferences, or IATA airport codes. The main problem is that there is simply not enough data from which to generalize.

Wikipedia. I tried again, but with Wikipedia, using a snapshot of the English site from June 2009. This is 23 gigabytes of data, most of it expository text composed by native speakers, plus bathroom humor vandalism. The list produced by this analysis is much better, though it contains artifacts from non-English Wiki language used in articles. The unabridged list appears in the appendix; my hand-selected favorites:

English Wikipedia, $n = 3$.

.0287%	smally	.00518%	reporth
.0156%	websity	.00484%	delection
.0156%	stude	.00459%	grounty
.0124%	chool	.00437%	betweek
.0120%	fontry	.00431%	fination
.0102%	undex	.00388%	manuary
.0099%	octory	.00360%	whicle
.0096%	coibot	.00262%	stategory
.0084%	footnot		

Lots of these could be the names of tech startups or Pokémon.

2.2 Coining words with coinduction

In the earlier sections I blithely produced tables of the most probable words according to an n-Markov chain. It is not obvious how to do this (or that it is even possible), so I explain the algorithm in this section. It's safely skippable, I mean if you don't want to know about a pretty cool algorithm that's not that complicated and might even be new, plus *dual math*.

Computing the probability of an individual word is easy. We prefix it with n copies of the start symbol $<$, suffix it with a single $>$, and then look up the probability of each symbol given its n preceding symbols in the table, and multiply those all together. We can compute the probability of any word this way. The problem with sorting all of the possible words by their probabilities is that there are an infinite number of them. We can't just look at short words first, either, because for example the word "thethethe" is many times more likely ($p = 6.08 \times 10^{-11}$) than the shorter "qatzs" (9.07×10^{-12}).

The solution is to use coinduction. Most people remember induction from school, maybe, which is the one where you have some base case like "0 is even", and then you prove that all numbers are either even or odd by assuming "$n - 1$ is even or odd" and proving "n is even or odd". From this we conclude that every number is either even or odd. The idea is the proof shows how to, for any given number m, count down to the base case "0 is even", and then repeatedly apply the $n - 1$ step (inductive step) to get back up to m. This is a great way to prove facts about finite things like numbers. Think of induction as a way you prove a statement like "Good to the last drop," or "There's always room for Jello."

Coinduction is a good proof technique for infinite things, like a sorted infinite list of possible strings. The idea behind coinduction is kind of like, you prove something like "0 is a number" (the base case), then prove something like "if n is a number, then $n + 1$ is a larger number", and then conclude that there exists an infinite series of numbers, each larger than the previous one. Think of coinduction as a way you prove a statement like "Once you pop, you can't stop," or "Never gonna give you up."

To sort the infinite list we don't actually use coinduction (we're not going to prove anything, just implement it), but its computational counterpart, corecursion. I just can't resist the "coin" pun.

What we do is define a function "most probable paths", which returns a (possibly infinite) stream of strings for a given starting state. Each string is finite and ends with the terminal symbol $>$, and they appear sorted by decreasing probability. (The most probable words overall will be just the first elements from the stream returned by this function when using a starting state like $<<<$ for $n = 3$.) Since we don't want to explore all possible strings in order to produce this

[15] Note that since $n = 4$ these words have to actually appear in status updates to have nonzero probability for this list. "g" is explained by frequent occurrences of "e.g.", for example.

list (there are infinitely many), the trick is to put a lower bound on the probability of the words that will be included. There are always finitely many words with probability greater than a given positive value, unless the Markov chain contains a cycle where each edge has probability 1. (This is impossible for Markov chains created only by observing finite strings, such as all the ones in this paper.) It is efficient to use a very small lower bound with this algorithm, like 0.00000000000001.

So the specification for "most probable paths" is to return all of the strings (that end with >) that exceed the given lower bound in probability, sorted in descending probability order. It is easy to check the path directly to >; we compare its probability to the lower bound by just looking it up in the table, and consider it if it exceeds the lower bound. For any other symbol *sym*, we will proceed (co)recursively: Call the probability of seeing *sym* next p, and then compute *tails*, all of the most probable paths starting in the state we would be in upon seeing *sym*. We turn *tails* into the sorted stream for the current state by just adding *sym* to the beginning of each string in it, and multiplying the probability by p. It remains sorted because multiplying by the same p is monotonic. The most important thing, which makes the algorithm practical (indeed terminate at all), is that we pass in a new lower bound: The current lower bound divided by p. After all, the outputs will be multipled by p, so they have to exceed this in order to meet the lower bound. This tends to increase the lower bound (sometimes over 1) since probabilities are between 0 and 1. This way, we only need to search a few symbols deep before it's clear that no string can exceed the lower bound.

Now we have a list of sorted streams, at most one for each symbol in our alphabet. It is fairly straightforward to merge these into a single sorted stream, by only looking at the first element from each one. Pseudocode for `most_probable_paths` appears in Figure 9 and for `merge_sorted` in Figure 10. Performance of this code is great; building the Markov chains (or even just reading the dictionary files) dominates the latency of the analyses in this paper.

3 Special cases

The empty string?? Is that a word? Could it be? Dude that is blowing my mind.

4 Backformation

The lexicon is generative, in the sense that it's possible to make new words that are generally acceptable, by following rules. Most people recognize pluralization of nouns by adding *–s* (even for novel words), or adding prefixes like *anti–*. We could investigate words that ought to exist by the application of rules, such as *examplelikelikelikelikelikelike*, but I see no straightforward way to justify the relative strength of such words.

A related way for words to enter the lexicon is by backformation. This is the reverse of the above process: A word like *laser* (initially an initialism) is legal, and then by running the rules of English backwards, we start to use *lase* as a word (the verb that a laser most frequently applies). In this section, I attempt to determine formation rules in English (by simple lexical analysis of the set of legal words) and then run these rules backwards to find words that seemingly should already exist.

Prefixes and suffixes. The first order of business is to find prefixes and suffixes that are usually modular. The kind of thing we're tring to find are "anti–" and "–ing"; stuff you can often add to a word to make a related word. The approach is straightforward. For each word, consider splitting it at each position. For *dealing*, we have *d/ealing*, *de/aling*, etc. For every such split, take the prefix (e.g. "de") and remainder ("aling"); if the remainder is still a legal word, then the prefix gets one point. *aling* is not a word so no points here for "de". We also do the same thing for suffixes (using the exact same splits, symmetrically). In this case we'll only get points for "–ing" since *deal* is a word. Every time a prefix or suffix appears we test to see if it is being applied modularly, and the final score is just the fraction of such times. Here are the ones with the highest scores:

```
1.000000000 -zzyingly    1/1
1.000000000 -zzying      1/1
1.000000000 -zzuolanas   1/1
1.000000000 -zzuolana    1/1
1.000000000 -zzotints    1/1
1.000000000 -zzotintos   1/1
1.000000000 -zzotinto    1/1
1.000000000 -zzotinting  1/1
...
```

Well, it's good to know that 100% of the time, you can remove "–zzotinting" from a word and it will still be a word. But this inference is supported by just one

```
fun most_probable_paths { lower_bound : real, state : state }
       : { string : symbol list, p : real } stream =
  let
    fun nexts i =
      case symbol_from_int i of
      NONE => nil
    | SOME sym =>
      let
        val p = (* probability of seeing sym in this state *)
      in
        if p < lower_bound
        then nexts (i + 1)
        else if sym = end_symbol
             then S.singleton { string = nil, p = p } :: nexts (i + 1)
             else
               let
                   val lb' = lower_bound / p
                   val tails =
                       most_probable_paths { lower_bound = lb',
                                             state = advance_state (state, sym) }
               in
                   (* Now multiply through the probabilities and add the symbol
                      to the head of the strings. *)
                   Stream.map (fn { string = t, p = p' } =>
                               { string = sym :: t, p = p * p' }) tails ::
                   nexts (i + 1)
               end
      end
    (* Try all next symbols. *)
    val streams = nexts 0
  in
    S.merge_sorted bysecond_real_descending streams
  end
```

Figure 9: Pseudocode for `most_probable_paths`. `advance_state` gives a new state from a previous state and symbol observed, so that for example `advance_state(abc, z)` gives `bcz`. The pseudocode for `merge_sorted` is given in Figure 10.

```
fun merge_sorted cmp l =
  let
      fun ms nil () = Nil
        | ms (s :: t) () =
          case force s of
              Nil => ms t ()
            | Cons (v, ss) =>
              ms_insert v [ss] t
      and ms_insert bv sg nil =
          Cons (bv, delay (ms sg))
        | ms_insert bv sg (s :: t) =
          case force s of
              Nil => ms_insert bv sg t
            | Cons (v, ss) =>
                case cmp (bv, v) of
                    GREATER =>
                      ms_insert v (singleton bv :: ss :: sg) t
                  | _ => ms_insert bv (s :: sg) t
  in
      delay (ms l)
  end
```

Figure 10: Pseudocode for `merge_sorted`. `ms` merges a sorted list, and `ms_insert` is a helper where we have a candidate best value `bv` which will either be the one we return at the head of the stream, or we'll replace it and then stick `bv` somewhere to be returned later. (This algorithm can be improved by making a data structure like a (co)heap; this is just a simple first pass.)

observation (the word *mezzotinting*); there are actually hundreds of such unique prefixes and suffixes. We need a better list.[16] Removing the ones that appear just a single time doesn't really help that much:

```
1.000000000 -zzazzes     3/3
1.000000000 -zzazz       3/3
1.000000000 -zzans       3/3
1.000000000 -zzanim      2/2
1.000000000 -zzan        3/3
1.000000000 -zygotic     3/3
```

Still bad. Let's turn up the juice to prefixes and suffixes that appear at least 10 times.

```
1.000000000 -wrought    10/10
1.000000000 -writings   12/12
1.000000000 -wraps      10/10
1.000000000 -wrap       11/11
1.000000000 -worms      69/69
1.000000000 -worm       69/69
1.000000000 -working    21/21
```

Much better! But the next step is going to be to try removing these prefixes and suffixes from words that have them, to find new words. Since these have modularity of 100%, we already know that every time we apply them, the result will already be a word. So they are useless for our analysis. Here are the most modular prefixes and suffixes with modularity *strictly less than* 1.

```
0.985714286 -makers     69/70
0.985714286 -maker      69/70
0.983606557 -wood       120/122
0.983471074 -woods      119/121
0.982758621 -down       57/58
0.982658960 -works      170/173
0.981818182 -houses     108/110
0.981818182 -house      108/110
0.981132075 kilo-       52/53
```

[16] The right thing to do here is probably to use binomial likelihood rather than the scale-independent fraction. But simpler approaches produce pretty good lists.

```
0.980752406 -less       1121/1143
0.980743395 over-       2190/2233
0.980000000 -books      49/50
0.980000000 -book       49/50
0.979591837 -proof      48/49
0.979310345 -lessnesses 142/145
0.979069767 -ships      421/430
0.978723404 -lessness   184/188
0.978723404 -board      138/141
0.978494624 -woman      91/93
0.978021978 -women      89/91
0.977528090 -ship       435/445
0.977272727 -manship    43/44
0.976744186 -weeds      84/86
0.976470588 after-      83/85
0.976190476 -manships   41/42
0.976190476 -making     41/42
0.976190476 -craft      41/42
0.976190476 -boats      41/42
0.976190476 -boat       41/42
```

Wow, now we're talking! The single word that cannot have "–maker" removed is *comaker*, suggesting that *co* should be word (noun: "What a comaker makes.").

Given this list, the next step is to identify potential words that can be backformed by removing prefixes or adding suffixes from existing words. Such a string can often be found via multiple prefixes and suffixes. For example, *twing* can be formed by removing "–ing" from *twinging* (a false positive, since the root word is actually *twinge* in this case) as well as by removing the prefix "lef–", which has modularity of 20% (including splits such as "lef/tie"). Maybe not good justification, but *twing* is a pretty good word anyway.

We define the probability of a word as its Markov probability (with $n = 4$, as this seems to produce the best results), times the probability that at least one of the potential backformation rules applies.[17] Here are the most likely words by backformation:

```
word    prob   most likely backformation rules
=============================================
dises   .023%  para- (0.42) fluori- (0.39)
               melo- (0.35) bran- (0.31)
tring   .020%  hams- (0.36) scep- (0.35)
               bows- (0.33) hearts- (0.29)
disms   .017%  triba- (0.31) drui- (0.30)
               bar- (0.27) invali- (0.27)
ching   .017%  day- (0.86) hot- (0.69)
               star- (0.51) guillo- (0.50)
sking   .017%  dama- (0.24) imbo- (0.18)
               fri- (0.18) atta- (0.17)
cally   .015%  anti- (0.78) specifi- (0.61)
               magnifi- (0.55) phoni-
pring   .015%  days- (0.67) heads- (0.62)
               outs- (0.54) ups- (0.51)
```

I think that this approach shows promise, but there appear to be a few problems: Many of these "rules" can be explained by bad segmentation ("heads–" appearing to be modular, for example, is really just "head–" plus "s" being a common letter.) Second, I believe the disjunctive probability of any rule applying is too naive for determining the score. For example, *tions* has almost a thousand different prefixes that could apply to it; the chance of *any one* of them applying is very nearly 1. But this is actually because "tions" is just a common way for a word to end. Legitimate root words to which many good prefixes are applied cannot be easily distinguished from common suffixes by this symmetric algorithm. More work is called for here.

5 Survey

On occasion I have been accused of "overthinking" problems, whatever that means. So to compare, I next hazarded a tried and true technique from grade school, the survey.

I asked a few people who happened to be around, "What word ought to exist?" Most people did not know what to make of this question, and also, because people seem to revel in the opportunity to get (well deserved) revenge on me by being disruptive trolls, many of the answers were designed to be unusable. In order to not reprint everyone's bullshit—but not introduce bias by selectively removing data—I discarded random subsets of the data until it did not contain bullshit any more.

Rob:	etsy, nuuog
Chris:	nurm
David:	wafflucinations
Lea:	hnfff
Reed:	pansepticon
Jessica:	gruntle

From this we can conclude that 16% of people wish *nurm* were a word, and so on. These words did not come with definitions, except for *gruntle*, which Jessica gives

[17] As above we only allow backformation rules that have at least 10 occurrences, to prevent degeneracy.

as "the opposite of disgruntle". This is actually already a word, but it was the inspiration for Section 4. *etsy* is the name of a popular on-line crafts community so I don't know why Rob would suggest that. The meaning of *wafflucinations* is clear from morphological analysis.

6 Conclusion

In this paper I investigated several different ways of answering the question: What words ought to exist? Each method produces different words, and some don't work that well, but nonetheless we have several rich sources of words, each time scientifically justified. I conclude with a section of recommendations for words that ought to exist, along with definitions.

6.1 Recommendations

Sweeeeeeeeeeeeeeeeeet with 19 *e*s is the clear favorite based on analysis of usage, so this one should be introduced. It means "Really sweet."

Rane sounds too much like *rain*, but *sare* has a unique pronunciation and many people seem to think it's already a word. I propose that *sare* be introduced as a noun meaning, "a word that sounds real but isn't."

Cho is similarly easy to pronounce and spell. I propose that it be defined as "A kind of cheese," so that we can really nail the new triple entendre on the classic joke. *Chomaker* is someone who makes that kind of cheese.

Unders was one of the most frequently occuring words towards the top of many analyses. This word should be a colloquialism for underwear, which would probably already be understood from context.

Dise is suggested by both the Markov model (as *dises*, *dising*) and backformation (as *dises*). I like thinking of it as being the root of *paradise*, where *para–* means something like "along side of" or "resembling". So *dise* is the place you're really looking for when you get to paradise and realize it's just a mediocre country club.

Helafelo is one hell of a fellow.

Addendum. During the preparation of this paper, the Scrallbe game has converged on a culture where the words played are real-seeming, with creative definitions. Examples: *frodeo* ("Gandalf is the clown.") *pridefax* ("An unproven treatment for telephone anxiety.") *eeeee* ("eeeee") *orzigato* ("Move mr. roboto. for great justice.") *stovebed* ("Now you don't have to get out from under the covers to make breakfast.") *ovawiki* ("The free egg cell that anyone can edit.") *gaptave* ("Two discontinuous musical intervals.") *achoolane* ("Nostril (colloq.)") *gplerious* ("Completely overcome by software licenses.) *bestcano* ("Some eruptions are better than others.") Thanks to the players for their contributions, especially Ben Blum, Chrisamaphone, Rob Simmons, and Flammin Fingers.

Appendix

Here are the most likely words induced by the English Wikipedia, with $n = 3$. I have left off the probabilities; they can be reproduced by downloading Wikipedia and running the software yourself, which only takes like 9 hours.

s ther w en t c d f m b ser e diff u sup p n de tal othe th wikipedia rd con id r ame edia cal uk wiki und als st l km wp g x ext j des td ter h co k sout don o cound v ver wer med res ind la em re ins afd ands div htm infor rom wher ple mi acces ii ent unive ff refere ger bution lish ince wikiped auth dist oth smally rea al y est cle fr lation tral gov nd mily isbn cond bel befor hould ber pers inder les gree mation lar chan dia ove stary compan mon val san cate nothe pdf tes ster nort pril hist sq clast ful ment q aft tr nown lign del z ave dify stor sity artic pring rese mas li ord ca thes oldid lin es ress flage blace pland mar mes conce exter wate mp jame lity ft cology spany mic ve han thist spam cand sher smal subdiv tem jan ture inded bothe ared cour rect thers ope mm dom der mand tt dr phy cd websity stude lor bbc ru mer flor succes ence tely aust stan clas et outh pl ints vfd fc wikipedit sus forman contry au edu inding el lic bord mr ston whic ide hous mich auser google pt le doi engle ed gen userved cols num nate cer withis clude shous inted wikipedian thered andex chool ning attle chael sta plation ampion crea fontry lege prese grough iii mader augh ll ching fb furt yearch examp octor nove furth bein tra nover texter mont sted gener quary inclub ander sco ast gr wome ality lon hough ano exas polid sease dec sor ga nov unty shor imdb offic partic oved forn lan fm undex rus arted alia cong vill eason gue shough cc vil tember octory jr gove writy thand anding rev colog gover edian publist mics comple syster olding coor useum spect expan puble bate tribut oned coibot betwo conter cles leason fina freen eng fa que fied tre hamp guage wered nic awar yout usic cgi andy nrhp clar arting bc cated rober feb ames ven jun thould defere artics sen frand cred ada cana rection modiff sal los ret ched decial collow notalk nl yor

elete sain ance cens chand apr delect dea stant bruary commer righ ch tor peral andia arth roup founty hor scies alth pete footnot html ath tribs ac chas eith pa tl refor bes bor cently fonter neer tration reat hi awa chris da wikipedits atter pp ents dvd arge wan sch sr fere inds norts tha nament anot oper aspx rence website minal brity dts apring goalso alber rown graphy anded min alian ength mility abbr talign bec com authe hite pic pation jone ords demy alt wated alife totalk lond tage ad ana evers amg gened colo engton mina lates porth fff lifor footbally stry shing hase lis eletion rol mally inton ded doesn ny mus pro du pre somes dired na becamp ire inces cologo mong swith acts aree ang els noth unic mous dest ali deat smale poing dc ission iv oct posity pred poss kore anda ja ms gard coi frome afc nhl publice mothe jers maded qual geory roy vid pur studing ip ign fath cally coundex col abouth twork threen img ess lishe lable reland cing chi thern dete ar ving di eur wers il dio hl ando seast sation orge ble sep db classion cm las conver evel ing popula uns polic vs frans ker ann phich ovember alish cher therence prock ife feate uservice indid unded powere hs phot assion von dece theight proview ama prote kg birthe specian userve se nois cfm ma len ause gle jp flagic differe counder rance swed lee tel ont sould whics artion rans ric parly overy ves sa grea ans une tary ss jul ne fami dever ange hms oright dation cription prothe univer lat tota un adming aaa ampan perfor athe seased pov becaust alson thed methe aff spand inc sis stral refered endex xfd texamp valign sha dists fron heade lood thest unived italk brary colled ucfirst ite gan ep rary oney wikipedition ra ex scord cition hout clos bon lp imple acade flagico bost isn spection samp ral tead sant roces yorks ories bords unition che dra epison eff stal abour ach earch busic reate conves ses hims mone fran carly nated lished fra pm oh att delet unior reporth daving emple tworld noved trand ross ie vall orld vi becauser caust dard exand feder ches yound gence bol stribut rement gf sugges tity forder aug systed mathe websited abover ltd recember colle wils mos sman lete dow eles throup ba unt expr formany df ments tect pape hool exper butions hited delection deces coung gon gers stors abc partics writion furthe rade mult bb lism afries frence bish arence tems musion colore ethe bute vie ia hing oute hom clain nom missue hus ns chich oxfor contribut vict autom boile novers aus airst jos proble werence georgan rel eas opedia rs ned aut iss pg ct othis indo shi dst alf ian grounty hile mor jor grap parth elet regory thems ell septed pc url gnf unds appy studed offere gb timent sed dary eved hally deate hr taly noming duri cented thead aliam mored pmid hered hong recore vious wates fathe goved squary soment mout rela eview unives shere trate betweek gia mory cas dal fi enge stribs colspan vol facted amer reser loc fination deven andid leade tand protes gar dely til alled sary tro alist ng cha charly ab hel begion mana sk powed ted calign wikipe dar nm churce mil ren dise rever sel howere nexter commen sonal oly subser coibothe shool hal froman eu waship pr rease ph rical op assed bility ording conces mal stitle iucn sri os mond lations mance artment rigin rt brite ffff jsp lb dd wikipeding lant sett arter te yource sh contribs footbal zh aution acter addit govers coln useries ract thich tain inten sec aa blo janual hon nowing letion tition pute goldid communite wors fontribut areat titute rected summent stational hu counded sp eld ples yeare nating fing foll reque fiel pol josed atland phile manuary civision ok feded ult dj sas alle cg buildid kee redia ap unk ko cr inver nomy sund toon reg aaaa cath dll lc ters communive bal prope refef ania mrs hased worder pria arry centing spon ade vand dor sourch tooks seph capan pbb colong sql petion shors fift prily prial brition fl seption nz fonto cbs fally whill losed cluded mations monal sland pict russion profilm coundo reture harly soute sc rr titution therent shouser pager outhe quent spons fn ited proces bable atte whicle sile si rece unitor yan grought cos pian comput trans jource arthe tury specia artice provid raction deleter alo fored ti eague wiking conth ree musly ars mf ups schood rey donal sected writed freet formate cre colution opers chen bd hp sember folled pagest smales wen socies cs engin aften chur delp bia ce nors cource ussion fontribs summan arties jon dism lang sking augus theren xs seve deball er uning truction nomic creat gian acco sv pera alls rict relete rects coord stre ences aread spane meth brea ener thernal thire whical rounty thu cl pola una thround uservation volved hund aga cames menter froma uset publis frey libe posite northe ska appedia recial ro fant hought mination geo presult andon occup ala nas thists ew maring secore nee dian inders inves infore hower sume scal kans indust eding ka aux brid var orded gived backgrough hugh ado prive eb rading joing hown euro lim kan chus alifor wille deced conven blogy nbc quall coller partion folle eleter reven reques lition humber moder backgroup tribution flages lears forch reight pault evely mayork sition feath recont guit metion hony deview nume confere fle unitory rican ima accore deside ris pedia lisher acture aland userview thate polor kar nel sents pany cluding vard bing woul havel greer thang genry effere locall desh thors empt

sult quence mir blook porthe tran rew bording shought kely neight lam stion publishe hors wis apan nony sourn nage orce profest jonal fonth younty deta pd wa lo peries flign fource ros rfc mely mt smalled ge retion relect vern alid aligh userves speer willa infl undon dels aftern gres namedia advan distor oria subdiving lx presign blocal edway def scho fic counds belf sily metry cread pi novery occes cribut holor reces inte howed cout chis ency actory lange cf nc bornia adven revie sames newspan shan orial userry suff eth moto showere hunty nucle witz ider mune flagicont aire shtm fri guary rigion mome afted heate mo formedia oble cnn nea ha canage borning tol navbox ava octors md frent offical wal arent anon artish cration rell dir auding ira butional labout bott southe dence sporth formall ps intry thamp stube unus usering ron usert conse brote lib oces bg proad tas werent chese lyn atta pris oile drage unre whenry ement stada chara raphy tection rica reletion foothe titler lassion areen theim conship formand ution unio ided guid riging brited buch septem eleted conds preser ta ort feld ine actional alize tex fer aused engly eque sevel atted smalle thome alty reath gation prefere hd nr peration covember govember livision fren dq aren deve va ffcc jo prian afl gion medy stategory nomich tured mel ene appa licy linal rought areate bethe othey eral opport ara kw whild cance communited varing brothe pia mu sched socian poin vis obs squal paking sma convers noing maken cathe ni solor alry flagicons aaaaa weren artical mtv flagion chall alm upons dister searly sur towned oring ests avg whis publicy ec thange gp ced pes whick figh peted fed sween discore sil ds sournal tay dier kim adver reak mote wouldn paign inal medit pointo cata sper pts idw mlb guis cational ul wasn septer aird throm fe feated collo itary uni uple distory disa regist vere effor whican ober bettle anning cution speciate palign ai cit moview flagicon af covery govery lina reast creen texand exten vic

There's a Hole in The Bucket: DNA Memory Leaks

Biolo, G. PhD

ABSTRACT

Walruses are an insignificant problem when it comes to the complaints agains the DNA inside cells. However, there is an obvious memory leakage problem that has develop increasingly overtime to the point where it must be addressed before evolution continues even more. We present some possible options for replacing the DNA programming with other methods of storing the same information without memory leakage.

INTRODUCTION

Through the evolutionary process, the DNA code has been edited multiple times by changing the codons, or nodes of data used for the generation of proteins. The problem with the process used in editing DNA code is that it causes memory leaks. Memory leaks causes several problems, among them the requirement for a bucket. With the increased requirement of buckets, walruses have been denied their happiness and are now on strike against the DNA in their own cells.

Over the years, large groups of unused nodes have banned together into introns in an effort to cause severe bogged down problems and severe memory leakage given that DNA is not a higher level programming language with an auto-garbage collector. If it were, the garbage collector could use his mop and bucket to help correct some of these problems.

METHODS

We went back in time to collect DNA samples along phylogenic trees in order to measure the accumulation of memory leaks over time. These samples were analyzed using standard high volume sequencing methods, then CLUSTALW compared against each other (just for shits and giggles) to verify that we took the correct samples. We hate wasting gas in the Time Machine.

RESULTS

Our results found that the introns increased significantly over the past; however, the number of walruses on strike against their own cells did not (figure 1). This is probably due to their population decline once they realized that their existence in this world is futile and that they should therefore perish much like the lemmings Disney threw off of a cliff.

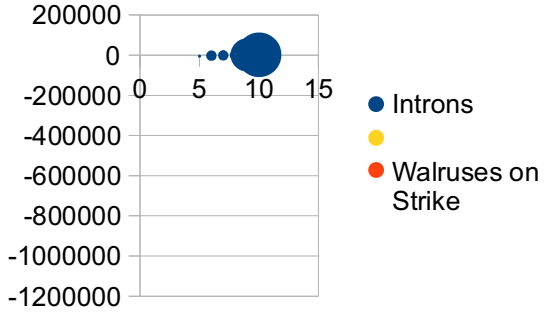


Figure 1: Increase in introns and thus an increase in memory leakage. The bucket was not invented until 5000 BCE, and therefore there was no ability to accurately measure the number of buckets needed to keep it up off the floor.

What we did find is that the memory leakage increased with evolution.

DISCUSSION

DNA is a bad programming language, so someone should fix it. Evolution is a demonstration of the failure of DNA to be useful overtime as a programming language given its constant problem with memory leakage. I suggest that we replace DNA with JAVA, PYTHON, Kittens, or PYTHONS that eat Kittens. In figures 2-5 we provide theoretical depictions of the reprogramming options.

Figure 2: DNA should be replaced with JAVA. This will be much more efficient since at least once a year it would get cleaned by the Monsoon season.

Figure 3:Python seems efficient enough that it could take care of all of those nasty evolutionary problems that occur over time. Even retroactively.

Figure 4: Kittens are a good replacement for DNA programming, and will cause less leakage. There is some worry; however, that they will get carried away with their new found responsibility and lose their desire to be cute and continue down a path of evolution no one wants by diving head first into the deep end of the gene pool.

Figure 5: As you can see, the Python is enjoying himself quite a bit as he devours the fluffy kitten and prevents the disastrous results of the kittens jumping into the deep end of the gene pool.

WORKS CITED

1.The Very Serious Biology Time Travelers Committee.[1]

2.The Independent Research of the Very Serious Biology Time Travelers Committee in cooperation with the WikiLeaks Memory Leaks Super-Secret Analysis Group.

3."How to Feed a Kitten to Your Hungry Programming Language" By C.M.L. Haskell

4. "Time Travel for Dummies: A Sensible Guide to Space-Time Continuum Surfing." E.M. Seeskwarred

5. "Fine Dining Recipes of the 20th Century: Cooking with the Dodo Bird." Captain Cook D. Bird

1.If you don't know who they are, then you don't need to even begin to understand this source.

SIGBOVIK 2011: TRACK THREE

Sleazeball-Computer Interaction

(Continued on next page…)

An Objection to “An Objection to “The Box and Circles Plot” ”

Dear sir or madame,

I wish to object on the strongest possible grounds to the previous paper. While it does make an admirable attempt at a sort of pseudo-Victorian meta-humor, its style too often fluctuates to be truly effective.

Indeed, it seems to be an unscrupulous attempt by an under-recognized author (possibly writing under a pseudonym) to focus more attention on an otherwise trivial subject. Perhaps this author is under the mistaken assumption that any publicity is good publicity.

I can assure you, this is not the case.

Yours,
Lohn T. Jongwood

Paper 3: An Objection to "An Objection to "The Box and Circles Plot""

REVIEWER: Prudy Coldfish
OVERALL RATING: 1 (weak accept)
REVIEWER'S CONFIDENCE: 3 (high)

While I stand by my "An Objection to "The Box and Circles Plot"", I appreciate the fine perspective that Lohn's contention presents. Verily, my colleague Lohn Jongwood provides a good analysis of tone and humor. Further, confident as I am that my right-thinking arguments will stand to the test of scrutiny, I welcome the invitation for this journal's readership to view content in a more critical light: perhaps this may foster an attitude less tolerant of disingenuous authors such as Mr. Longwood.

Good Friends Project: A Little Social Experiment in Stalking

Brian C. Becker, Pyry Matikainen, Heather Jones, Prasanna Velagapudi, Michael Furlong, Brina Goyette, Heather Justice, Umashankar Nagarajan, and Joydeep Biswas

***Abstract*—In this paper, we automatically poll the longitude and latitude of a friend (let's call him Bob) who is publicly sharing his location via Google Latitude on his smartphone. For 11 months, GPS locations were logged at 15 minute intervals, yielding over 30,000 locations spanning multiple continents. Analysis of this data reveals much about the activities of our friend Bob, satisfying cravings best exemplified by gossips and tabloid readers.**

I. Introduction

DANGEROUS some phrases are. For instance, consider with me for a moment the implications of posting the following sentence to the Internet: "*As part of a little social experiment, I'm making my location public via Google Latitude*." Does that not invoke the mental image of a serpentine wagon circle of words that effectively read "stalk me" (see Figure 1)? No? Perhaps it was just a fanciful agglomeration of too many giddy ideas that led us to begin a stalking project, but we rationalize it as being of great interest to sociologists and machine learningists.

II. Background

Google Latitude is a web-service provided, shockingly enough, by Google. It allows you to share your location by running a service on your smartphone that periodically uploads your current GPS location. Others can then see your location on Google Maps (see Figure 2). Generally, the service restricts the visibility of your location to friends, but Google Latitude will also let you share it publically to all.

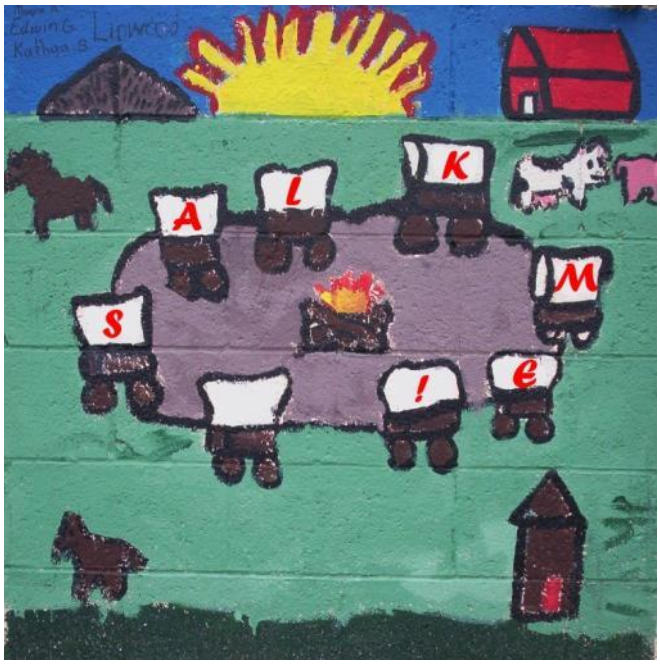


Fig. 1. An apt photo depicting the authors' imaginative responses to reading the phrase "*As part of a little social experiment, I'm making my location public via Google Latitude.*"

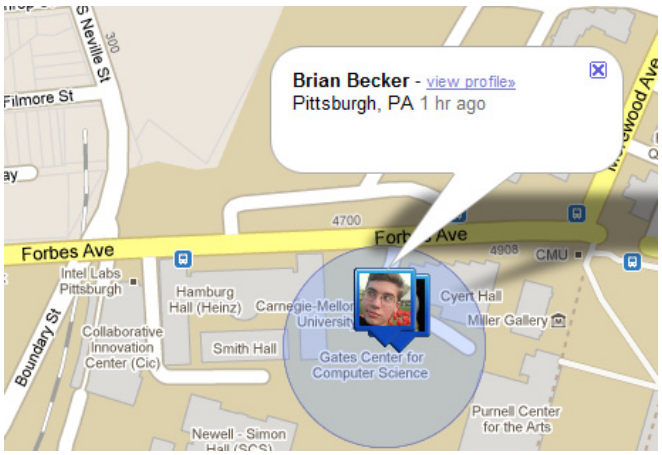


Fig. 2. Screenshot of Google Latitude interface, showing the location of the primary author, with error bounds.

Motiving our project is a friend, let's call him Bob, who not only made his location public but posted it to his website with a notice that it was for a social experiment. Unable to resist the temptation to use this data for evil, the authors of this paper immediately began a stalking campaign and christened it "Good Friends Project." To the best of our knowledge, this project has been kept a secret from Bob and represents the first of its kind, making it state-of-the-art.

III. Methods

We reversed engineered Google Latitude and wrote a Python script + cron job to save the latitude, longitude, and time to a file at 15 minute intervals.

IV. Results

For 11 months between April 6th, 2010 and February 25th, 2011 our system logged ~30,000 of Bob's GPS locations. In this section, we analyze this data to reveal trends and other information of interest. Or just generally stalk him.

A. Probability Density Heat Map

Using a Google Maps image of Pittsburgh, all GPS points were overlaid in a Gaussian-blurred heat-map with logarithmic scaling to prevent saturation. This represents the logarithmic probability density of Bob's location at any given time (see Figure 3).

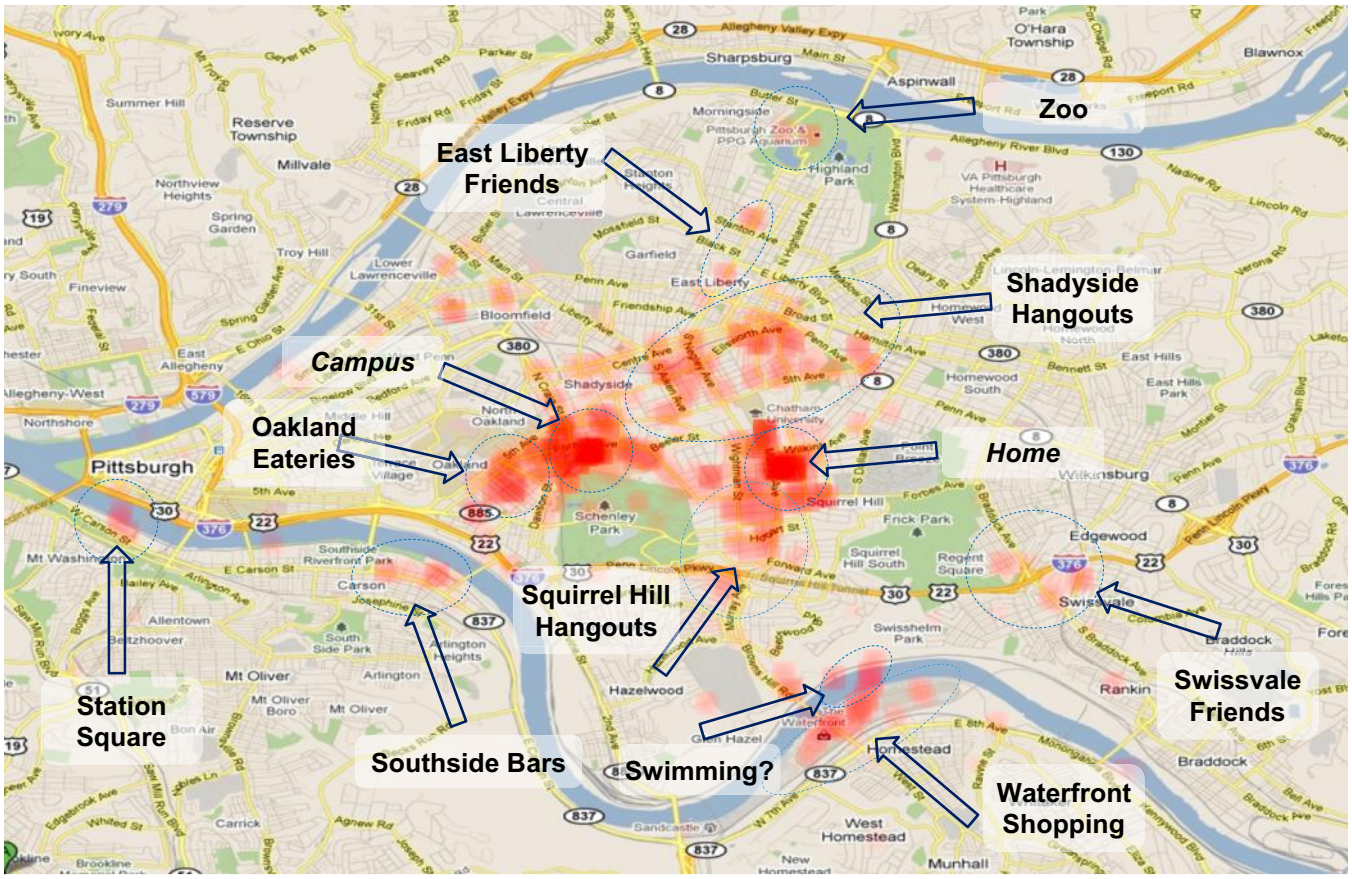


Fig. 3. Heat-map of the 30,000+ GPS coordinates overlaid onto a Google Maps image of Pittsburgh and Carnegie Mellon University. The heat-map is displayed logarithmically to avoid saturation. The map is annotated with common Pittsburgh features.

From this heat map, we can infer quite a bit about Bob. From the two points of highest density, we can assume that Bob works at Carnegie Mellon University (or possibly a surrounding shop such as Starbucks) and lives in Squirrel Hill. He spends a lot of time on Craig Street and in Oakland, which correlates well to the theory that he is a CMU student. He frequents the Waterfront and visits the Southside periodically. A number of friends he visits live in Shadyside, Swissvale, East Liberty, and Bloomfield. Also, it appears he has visited the Pittsburgh Zoo & Aquarium at least once.

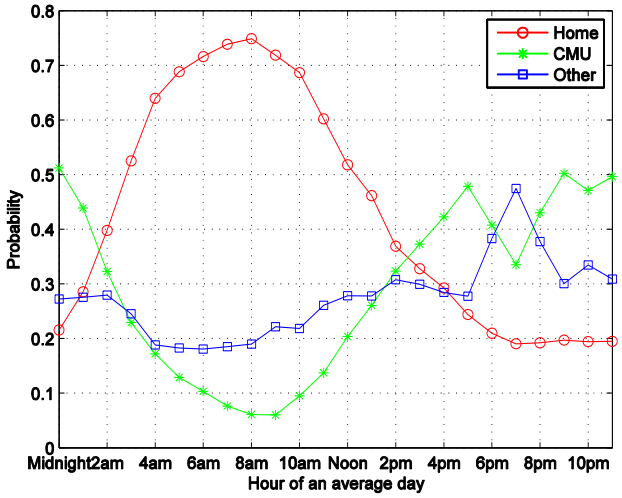


Fig. 4. Probability of Bob being at home, CMU, or somewhere else (other) as a function of the hour of the day.

B. *Time Distribution among CMU, Home, etc.*

If Bob is indeed a CMU student, we can analyze his location by hour to discover when he comes into campus, when he leaves for home, and when he spends time off campus. As seen in Figure 4, Bob is something of a night owl. The best chance of finding him on campus is at midnight (surprisingly!), and he is most probably home around 8 am. There is an interesting anomaly at 7 pm where he goes off campus but does not go home. Our best guess is that he goes off campus for supper, probably to Craig Street or Oakland as evidenced by their heavy concentration on the heat-map in Figure 3.

Taking a longer view of things, we can analyze Bob's patterns by month. Figure 5 shows the allocation of time per category. We can see that Bob worked heavily during January, February, and October, spending more time at CMU than at home. July and August seemed to be more laid back vacation months, or perhaps Bob just spent more time working in cool summer places around town. A large portion of December was spent at home, although this could be skewed if Google Latitude was disabled.

Overall, Bob spends 43% at home, 29% on campus, and 27% elsewhere.

TABLE I
LONG DISTANCE TRIPS

Location	Start Date	End Date
Toronto, Canada	5-8-2010	5-14-2010
Warren, OH	7-23-2010	7-24-2010
Philidelphia, PA	7-29-2010	8-1-2010
Washington, DC	9-9-2010	9-10-2010
Hong Kong, Taipei	10-17-2010	10-24-2010

The dates and locations of long distance trips Bob has taken.

C. *Long Distance Trips*

So far, we have focused on Bob's residential movements around Pittsburgh and their distribution between key locales. However, by examining the large excursions from Pittsburgh, we can gain a broader understanding of his life. Table I lists trips outside the Pittsburgh area, which even includes an international trip to Taipei. In total, our friend Bob traveled ~37,000 km (23,000 mi), most of which is accounted for by these long distance trips.

V. DISCUSSION

We have presented a novel way to track people who purposefully enable Google Latitude. This method also works with those who may find Google Latitude "accidentally" enabled after lending their smartphone to a friend to check email. All that is required is a computer connected to the Internet to periodically poll Google Latitude for the GPS locations. Admittedly, analysis of such resulting data is somewhat tedious and time consuming, so future work of this paper will focus on automated ways to stalk your friends. Alternatively we could find ways to pawn such analysis off on poor unsuspecting undergrads. Another possible avenue of research is discovering an automated method to remotely enable Google Latitude without the operator's express knowledge.

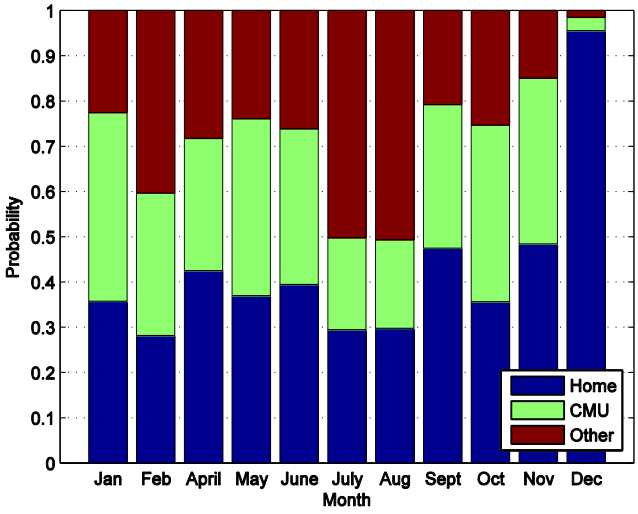


Fig. 5. Probability of Bob being at home, CMU, or somewhere else (other) as a function of the month.

ACKNOWLEDGMENT

We would like give a special thanks to Bob for publically providing his location and allowing us the wonderful opportunity to stalk him. We hope this tiny taste of an analysis whets his appetite for further social experiments.

REFERENCES

Available upon request.

Paper 14: Good Friends Project: A Little Social Experiment in Stalking

PC MEMBER: Laura Berg
OVERALL RATING: -3 (strong reject)
REVIEWER'S CONFIDENCE: 1 (low)

A little too "Good" of Friends project

If someone ever thought they may want to consider getting a restraining order, now maybe be the time. In fact, with the new Google Latitude, it might even become fashionable. While it is interesting to consider that someone may use this tool to study how they spend THEIR OWN time, to make your information publicly open to stalkers was not the brightest idea. What this paper failed to mention was the future homicide they were planning by tracking where this individual "Bob" went every single day. I'm also curious, why photos, both satellite and from behind that bush on the corner of his street, were not included given that he was being stalked. Like a zebra. mmmmmm zebra burger.

So I would like to conclude that through "Bob's" worldly travels, both in Pittsburgh and Out, he has yet to devour a zebra burger because he is the zebra burger. He is being stalked. Consider a restraining order "Bob." Bob? ...Bob?...oh no...

Who is the biggest douche in Skymall?

Dr. Tom Murphy VII Ph.D.*

1 April 2011

Abstract

Did you ever notice how there are so many douches in the Skymall catalog? This paper investigates the 22 males pictured in the January 2011 issue, using Internet technology to determine their douchiness. We then present an efficient image recognition algorithm that reliably predicts douchiness from photos.

Keywords: computational douchebaggery, as seen on tv, man who can sleep in any seat

Introduction

In that Luddite void between closing the cabin doors and the beep indicating it is now safe to use approved electronic devices, there is one perfect pleasure: The Skymall catalog. It has everything, including: Pointlessly impractical products you cringe at just imagining someone receiving as unwanted gifts at Christmas, copy that preys on the insecurities of business travelers, typo and physically impossible hyperbole treasure hunts galore, Photoshop disasters, new friends, and old familiar faces. But since 1990, science has wondered: **Who is the biggest douche in Skymall?**

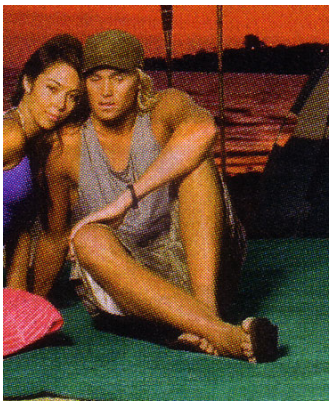
It is difficult to assess the absolute douchiness (say, on a scale from 1 to 10) of a given person. So, in order to answer this question, we used Internet Technology to conduct a series of more-douche–less-douche battles between randomly selected pairs of participants. The visitor is simply asked: Who is the bigger douche? The proportion of battles won, overall, is the final douche score.

After thousands of battles waged, we converged on the following results, ranked in descending douchiness:

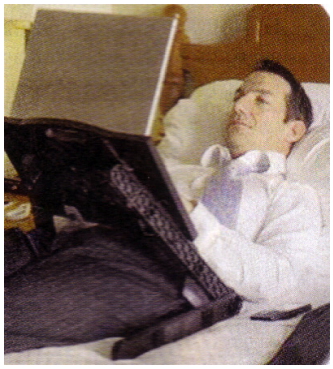
Results

In Charge of the Music
won 207/66; 75.82% douche

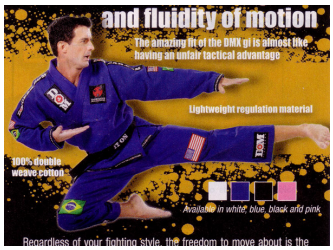
The Thinker
won 201/73; 73.35% douche



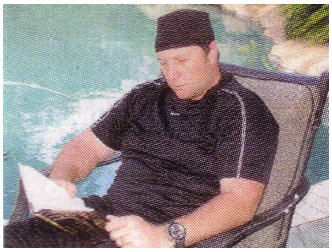
Seaside Date
won 197/76; 72.16% douche



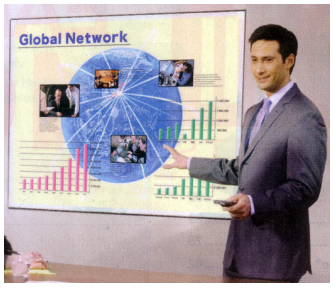
Reclining Numbercruncher
won 193/80; 70.69% douche



Karate Genius
won 188/85; 68.86% douche

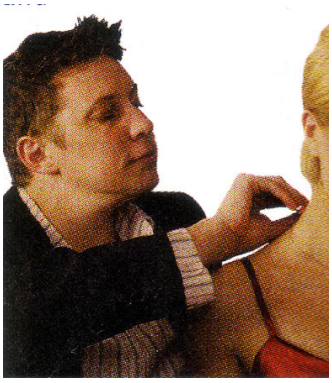


Poolside Bibliophile
won 177/95; 65.07% douche



Traveling Salesman
won 171/102; 62.63% douche

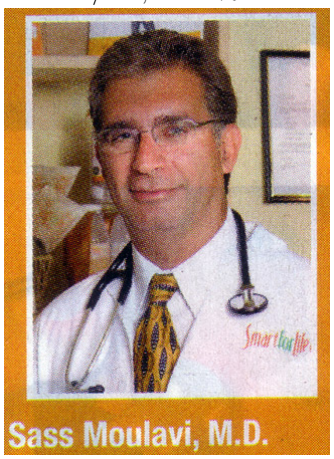
Beauty Rest
won 167/106; 61.17% douche



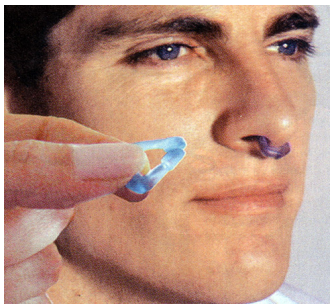
Jeweler
won 167/106; 61.17% douche

I Have to Take This
won 161/113; 58.75% douche

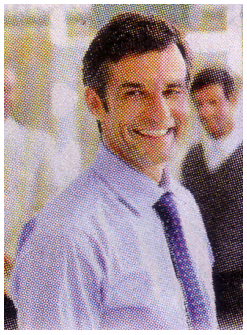
Treatment Recipent
won 121/152; 44.32% douche



Sass Moulavi, M.D.
won 117/155; 43.01% douche



Cured Snorer
won 141/132; 51.64% douche



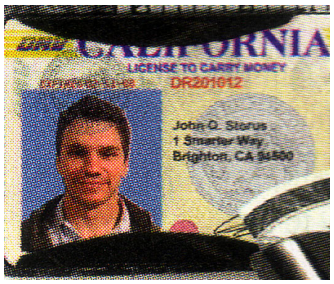
Business Expert
won 118/155; 43.22% douche

New Haircut
won 96/177; 35.16% douche

Man who can Sleep in Any Seat
won 131/141; 48.16% douche

Confident in Glasses
won 118/155; 43.22% douche

Woodsy Gentleman
won 86/186; 31.61% douche



John Q. Storus
won 78/195; 28.57% douche

Father/Kidnapper
won 73/199; 26.83% douche

Got the Promotion
won 59/214; 21.61% douche

Silver Medalist
won 34/238; 12.50% douche

Although many people disagreed on the finer points of what constitutes a douche, the results were fairly significant, in that there were many participants that were consistently perceived as more or less douchey. This is truly a victory for the scientific method. Speaking of victories:

Douche recognition

Having collected consensus on what constitutes a douche, we next turn to the problem of determining whether any given person is a douche, even if that person has not participated in hundreds of rounds of douche-battle. Since people appear to be able to make decisions based mainly on images, we look to image processing techniques.

Images are formed using "pixels", which are like tiny individual color dots. Each color dot, or "pixel", is saved in a file. It turns out that these files are all the same one: Every "pixel" is in a file, which constitutes the series of color dots, as a series of bytes or "1s and 0s" 2, which constitute the digital information that is the file, or "pixels." Point is, in order for a computer to "see" a file, all it needs to do is look at it, by rubbing the files and "pixels" on its CPUs, the same way that you or I look at a picture by rubbing it on our eyeballs.

The problem with most image recognition algorithms is that they do not work, and are also slow.

We eschew the traditional model-based approaches, instead using efficient hashing algorithms such as SHA-1 and MD5. These operate directly on the "1s and 0s" of the image, and run in linear time. This way, even if the algorithms do not work, they will at least be fast.

We find that the algorithm SHA-1 correlates with the douchiness of the image, but not well. The MD5

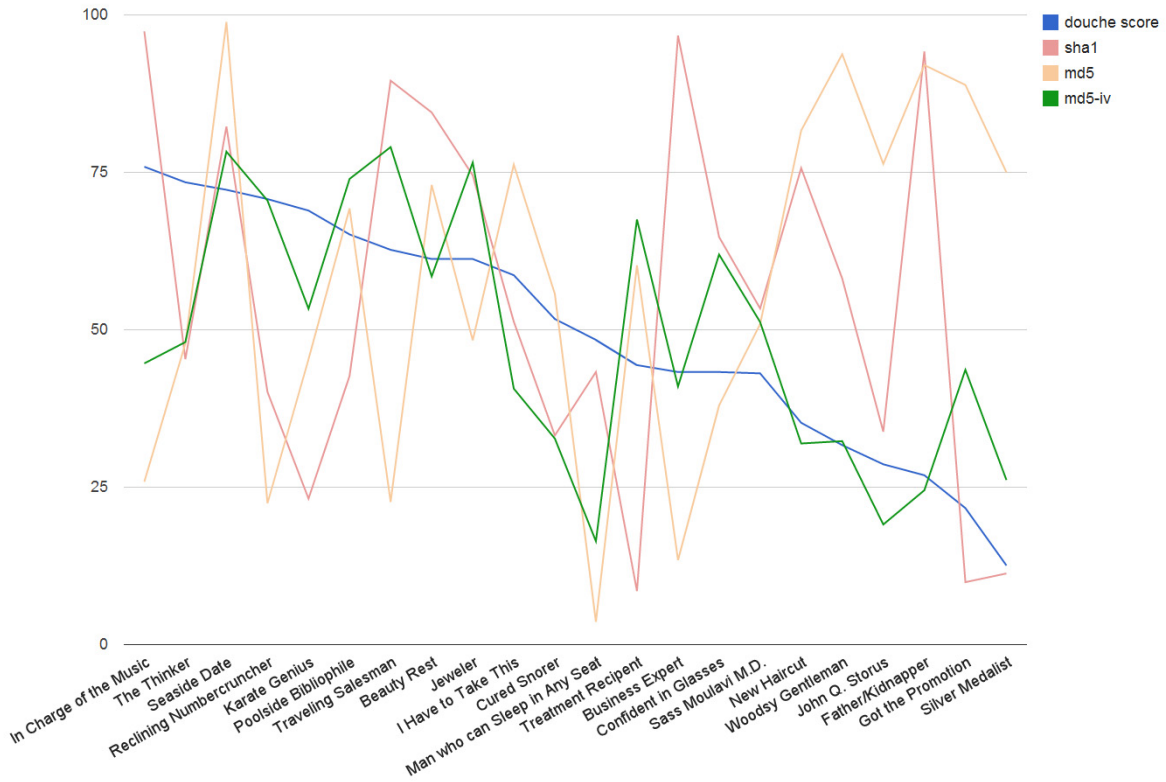


Figure 1: Performance of different prediction algorithms. SHA-1 correlates with the douchiness of the image, though it tends to overestimate the douchiness of medium-low douches. MD5 actually has negative correlation. A tuned variant of MD5 with the initialization vector `A8F82303 08A1B76B AA25DA9E 4C2C1883` correlates quite neatly.

algorithm is faster but in fact correlates negatively. However, by fine tuning the initialization vector, we are able to produce a variant that is just as fast and correlates very well with the data (Figure 1).

Conclusion

Do you disagree with these data (Y/n)? If (Y), then Science never Sleeps! `http://snoot.org/toys/wuss/skymall/`

Poolside Bibliography

Sorry, I didn't read any papers or anything or do actual science.

Figure 2: Image files consist of “ones” or “zeroes”, which is digital information pixels (pictured).

A Materials Based Approach to Hardware Digital Rights Management

Matt Mets *Matt Griffin* *Marty McGuire*
MakerBot Industries

Abstract

The advent of cheap and accessible rapid prototyping technologies[1] is creating a revolution in at-home, at-will manufacturing. As these technologies become cheaper, more and more consumers are able to download digital design files to "print" (or "personal fabricate") out a wide range of objects, from cheap alternatives of otherwise-expensive replacement parts to fan-created renditions of popular art. In the opinion of many experts[2], these innovations appear to be protected by existing intellectual property law. For the brave content companies that engineer the world's economy, new protections are needed to retain the precious intellectual property rights ecosystem which makes vast corporate entities possible. In this paper, we propose a novel mandatory materials compliance method to stem the tide of rampant intellectual property theft by pirates with rapid prototyping apparatuses.

Introduction

To paraphrase former MPAA president Jack Valenti, "the [MakerBot] is to the American [CAD worker] ... as the Boston strangler is to the woman home alone." Current intellectual property law appears to provide extraordinary protection to the "pirate fabber", allowing the wanton duplication of valuable replacement parts[3], and even iconic characters that would normally receive protection under copyright law find themselves at the mercy of these "prototype pirates" [4, 5].

In Code[6], Lawrence Lessig asserts that intellectual property owners can make use of technology to create extra-legal protections for their properties. Taking inspiration from this, we present a novel method for protecting the intellectual property rights of the owners of 3D-printable content: a mandatory materials compliance requirement for all 3D printer owners, limiting them to creating quickly-degrading or single-use prints, permitting a time-limited "try before you buy" period for unrepentant, unreconstructed prototype pirates while at the same time reinforcing the role of the content owners as the sole source for durable, high-quality parts, toys, sculpture, knickknacks, etc.

1 Prior Work

The music and movie industries have provided a rich history of methods to protect digital intellectual property rights[7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

Unfortunately, these methods have all failed to stem piracy in the real world.

2 An aqueous approach to item un-longevity

The desired result of this study was to determine a suitable materials paradigm exhibiting both short-term rigidity and medium-term degradation; ie, a temporally unstable construction material. A number of different approaches were considered (Matchstick erector set, Nitroglycerin cookie molds, "EZ"-bake napalm, deploying Richard Stallman to repossess materials after a given time). Though many of these methods may have utility in other situations, they all have the disadvantage of being hazardous to manufacture, or difficult to deal with. Instead, a copy protection mechanism based on the aqueous properties of polyvinyl alcohol (PVA) was identified as as a candidate that met this stringent set of requirements.

Unlike traditional 3d-printable thermoplastics, which typically have a low solubility when placed in aquious solutions, PVA exhibits the unique property of being completely dissolvable in water:

$(C_2H_4O)_x + H_20 \rightarrow goop_{aq}$

This makes it a prime substrate upon which to build models whose lifespan is guarinteed to be short. By combining this material with an infrastructure for purchase or rental of limited lifespan objects and a hardware system that guarintees models will only be printed in the correct plastic, an unbreakable hardware DRM scheme can be created.

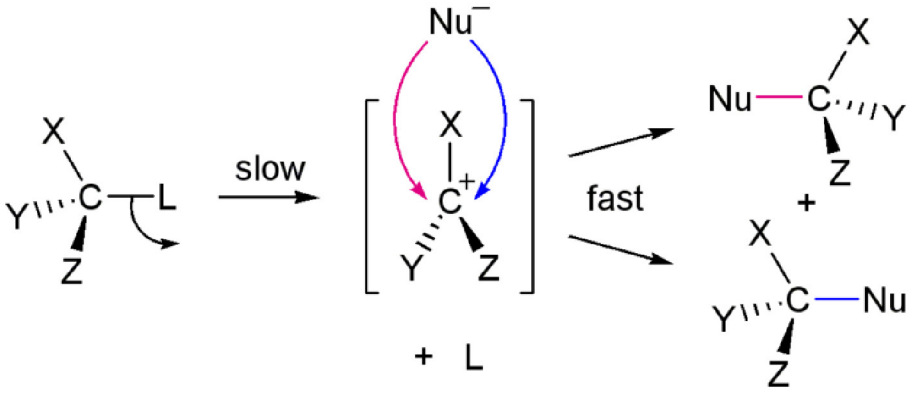


Fig. 1: SN1 reaction mechanism (unrelated)[26]

3 Process Explanation

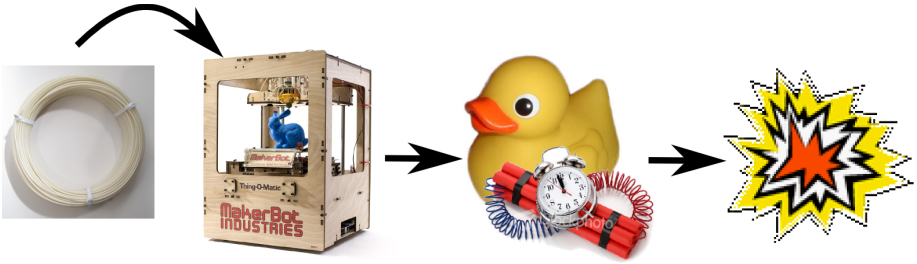


Fig. 2: Process Flow Diagram

The process detail is as follows:

1. Polyvinyl Alcohol stock loaded into 3d printer.
2. User acquires the right to lease a model through modeling broker.
3. Model is downloaded to machine, which verifies that the user has the correct permissions to print the model, and that properly licensed plastic stock is loaded into the machine.
4. Model is printed in fully functional, but time-limited material.
5. User enjoys model in a manner that agrees with the terms of service, for the allotted time.
6. After user's lease expires, the model automatically self-destructs by breaking down, effectively preventing unlicensed use.

4 Results: Children's Play Skull

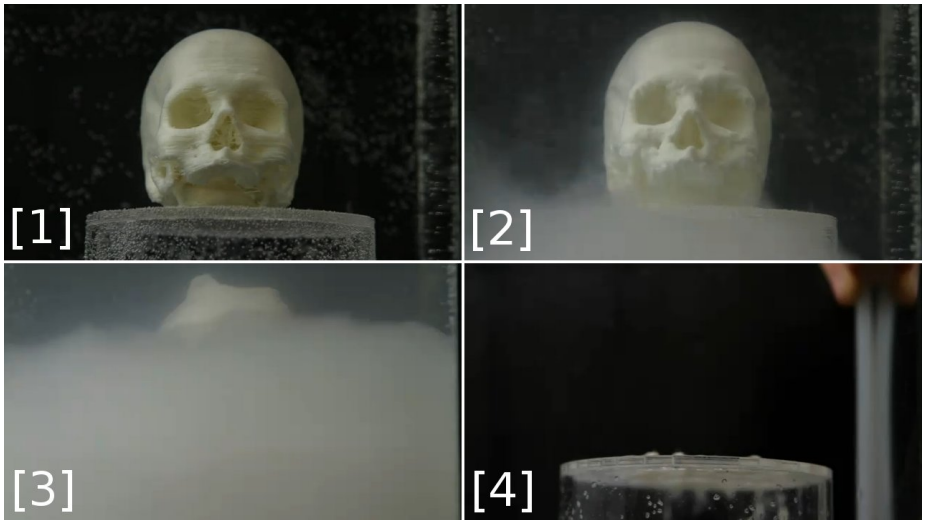


Fig. 3: Hardware Digital Rights Management in action: Automatic dispersion of a Children's Play Skull

The methods considered in this paper were applied to a popular children's bath-time toy, the Play Skull[27].

1. A PVA skull is placed in water at beginning of play period.
2. After allotted use time, toy begins self-revocation of it's rights.
3. Cloud-based dispersion technology disposes Model.
4. At end of self-revocation process, user's rights to the model fully relinquished.

5 Future Work

A poll of the authors suggested that conditions for more work were not favorable, as it had become warm and sunny "outdoors".

References

[1] "The printing of parts and products has the potential to transform manufacturing because it lowers the costs and risks." 3D Printing: The Printed World. 2011. The Economist.

[2] "Unlike with copyright infringement, the mere possession or downloading of a file is not enough to create infringement liability." WEINBERG, M. 2010. It Will Be Awesome if They Don't Screw it Up: 3D Printing, Intellectual Property, and the Fight Over the Next Great Disruptive Technology. Technical report.

[3] "Things Tagged 'Replacement". http://www.thingiverse.com/tag:replacement

[4] "Optime Primus Head". http://www.thingiverse.com/thing:7087

[5] "Cartoon Mouse Cookie Cutter". http://www.thingiverse.com/thing:6217

[6] Lawrence Lessig. 1999. Code and Other Laws of Cyberspace. Basic Books, Inc., New York, NY, USA.

[7] H.R. 2281. 1998.

[8] DeCSS. http://www.cs.cmu.edu/~dst/DeCSS/

[9] "Sony asks for restraining order against Geohot, fail0verflow over PS3 exploits". Engadget. Jan. 2011. http://www.engadget.com/2011/01/11/sony-sues-geohot-fail0verflow-over-ps3-exploits/

[10] Silenced Professor Sues SDMI, RIAA. PC World. June 2001. http://www.pcworld.com/article/52006/silenced_professor_sues_sdmi_riaa.html

[11] DeCSS T-Shirt. CafePress. http://www.cafepress.com/+decss_tshirt,20999686

[12] "DVD Jon Breaks Free of iTunes DRM". The Unofficial Apple Weblog (TUAW). Feb. 2008. http://www.tuaw.com/2008/02/20/dvd-jon-breaks-free-of-itunes-drm/

[13] Real Story of the Rogue Rootkit. Bruce Schneier. Wired. Nov. 2005.

[14] "Macrovision RipGuard can't stop ripping". AfterDawn. Nov. 2005. http://www.afterdawn.com/news/article.cfm/2005/11/24/macrovision_ripguard_can_t_stop_ripping

[15] "Disposable DVDs Go to the Dumps". Wired. May 2003. http://www.wired.com/entertainment/music/news/2003/05/58906

[16] Sony's War on Makers, Hackers, and Innovators. Phillip Torrone. Make: Online. Feb. 2011. http://blog.makezine.com/archive/2011/02/sonys-war-on-makers-hackers-and-innovators.html

[17] Blu-Ray DRM Defeated. John Leyden. The Register. Jan. 2007. http://www.theregister.co.uk/2007/01/23/blu-ray_drm_cracked/

[18] Official explanation of controversial Assassin's Creed 2 DRM. Ben Kuchera. Ars Technica. Feb. 2010. http://arstechnica.com/gaming/news/2010/02/ubisoft-details-drm.ars

[19] Amazon Erases Orwell Books from Kindle. Brad Stone. New York Times. July, 2009. http://www.nytimes.com/2009/07/18/technology/companies/18amazon.html

[20] 12-Year-Old Sued for Music Downloading. FOXNEWS.COM. Sept. 2003. http://www.foxnews.com/story/0,2933,96797,00.html

[21] Universities Realize That The RIAA Is Taking Advantage Of Them In Lawsuits On Students. Mike Masnick. TechDirt. Aug. 2008. http://www.techdirt.com/articles/20080812/0229461957.shtml

[22] EA's Spore Breaks Piracy Record. David Jenkins. Gamasutra. Dec. 2008. http://www.gamasutra.com/php-bin/news_index.php?story=21414

[23] Blizzard Freezes Bnetd Gaming Platform, Sues Own Customers. EFF. Apr. 2002. https://www.eff.org/press/archives/2002/04/08

[24] DirecTV dragnet snares innocent techies. Kevin Poulsen. The Register. July 2003. http://www.theregister.co.uk/2003/07/17/directv_dragnet_snares_innocent_techies/

[25] Open Source. http://xkcd.com/225/

[26] SN1 reaction mechanism. Calvero. Wikimedia Commons https://secure.wikimedia.org/wikipedia/en/wiki/File:SN1_reaction_mechanism.png

[27] MakerBot Introduces Water Soluble 3D Printer Filament! http://blog.makerbot.com/2011/02/21/makerbot-introduces-water-soluble-3d-printer-filament/

The "Who's Yo Daddy?" Problem
Or: Solvin' Concurren' Axcess Issues wit Computers, Yo

Dr. Donna M. Malayeri, PhD

Heather Miller, PhD(c), Esq., PPPoE, P2P

Herr Doctor Doktor Klaus Haller

Abstract

Computer Science research can solve many real-world problems. Here we describe our novel research, Uniqueness Types, and how it applies both to multi-threaded programs and to real world scenarios. In particular, we solve issues that commonly arise in popular daytime sociological science documentaries.[1]

1. Introduction

In multi-threaded programs resources such as memory and files are at the same time accessed, i.e., concurrently. This often leads to problems, such as blue screens, not-disappearing hour glasses, the spinning dying color wheel of death and so forth. For example, let's consider a thread program that a file opens, and it accesses. It may happen that yet another threaded process at some later point in time closes the file without the first program knowing about it. If the first program to the file again accesses, then the user may a blue screen experience.

2. Uniqueness Types

To solve the unique problems introduced in the introduction, we introduce a novel language-theoretic type theory called Uniqueness Types [2]. We base our theoretical theory on flow-sensitive linear logic with typestate [1]. The idea of our research approach is to assign unique types to pointer variables. A pointer is unique whenever the compiler program knows that all other pointers are pointing towards other datums, or, conversely, if and only if no other pointer is pointing towards it. In predated works computer researchers have published theoretical experiments with unique pointers that sometimes their typestates change.

Unique pointers with uniqueness types give rise to a unique approach to avoid the unique problems of hour glasses and spinning dying color wheel of death. Somehow we can make sure everyone points to the hour glass or something like that. Or no, maybe the thread program must have pointers with unique names.[2]

[1] E.g., *The Jerry Springer Show*.

[2] Americanadian translation: if two threads access the same resource, say a file, it would be bad if one thread opened the file, then the other thread closed the file, then the first thread tried to then read from the file. Essentially, threads with pointers to a shared resource need to know about state changes that may affect future operations on that resource. A uniqueness type solves this problem, and you can read more about it in this boring—er, exciting—research paper [2].

3. Real-World Scenarios

As interesting as these programming problems are, feel we that it time is to computer science to real world problems apply. How else can we, with a straight face, to funding agencies the claim make that we real problems that affect people's everyday lives solve?

We believe that a good source of real-world problems documentaries is, particularly those highly-rated ones that on broadcast television are shown. These informational programs an unprecedented view into the daily life of the everyman provide. For the purposes of this paper, *The Jerry Springer Show* as our primary source we shall use, though our solution to scenarios seen on other esteemed programs is applicable, such as *The Maury Povich Show* or *Jersey Shore*.

3.1 Real-World Problem Statement

> So, this one bitch is a real ho fo real and she get wit three playaz, Playa 1, Playa II, and Playa Playaa Playa. And now showty pregnan' and she sez the baby daddy is Playa Playaa Playa and he a pimp.[3] He sho' Playa 1 is da baby daddy fo serious and he don wants to pay no child suppo'. Da ho gots a paternity test dat sez da baby daddy is Play Playaa Playa but he thank it a fake.

Here, the problem is access to the shared resource within a critical timeframe. Since the third man does not trust the results of the paternity test, believing the woman to be a "lyin' ho," we need to provide the parties in this scenario with a fool-proof mechanism for determining parentage.

We believe this problem is widespread, and in the tradition of clever monikers for computer science problems (e.g., Travelling Salesman, Sleeping Barber, Din-

[3] This means he has lots of money.

ing Philosophers), we name this problem "Who's Your Daddy?"

Further, this is a problem that clearly arises often in practice. As evidence, see Figures 1, 2, and 3.

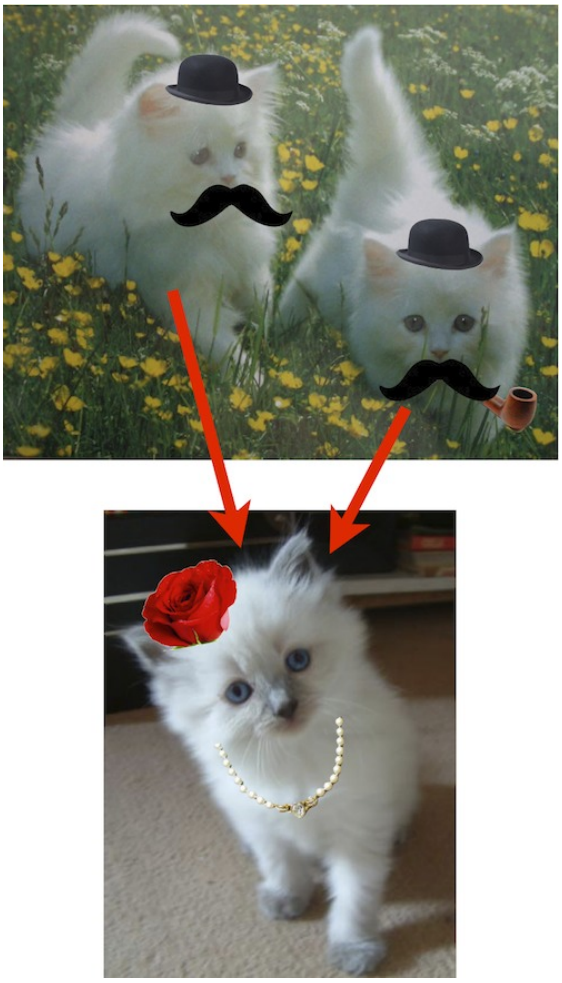


Figure 1. Two male kittens access a shared female kitten resource, creating contention. (No, this is not a gratuitous kitten picture.)

4. Applicability to Real-World Scenarios

It is not immediately obvious how the theoretical results of Uniqueness Types to Who's Your Daddy can be applied. We present here an iterative approach to a solution, starting with a naïve solution and refining it to handle all possible scenarios.

4.1 Solution 1: Hëävy M̈ëtäl Locking Device with Physical Key

The problem here is that several pointers the shared resource may access, and her state may at any time change, without it being obvious to any of the parties involved (including the resource herself) that a) the state change has occurred and b) which pointer caused the state change.

Figure 2. Unexpected physical confrontation of two threads who accessed a female within the same timeframe.

Figure 3. The shared resource unsuccessfully attempts to arbitrate between threads who are engaged in a violent altercation. As seen on *Jersey Shore*.

We introduce the following protocol:

- An external party outfits the resource with a hëävy m̈ëtäl locking device with a physical key. For instance, the Victorian chastity belt would be quite useful here (Fig. 4).
- The physical key is retained by the external party for an incubation period of not less than 9 months.
- The first pointer retrieves the key from the external party and may access the resource.
- When the first pointer is finished with the resource, he or she returns the key to the external party.
- The incubation period is again started and the process repeats itself.

Figure 4. An antique hëävy m̈ëtäl locking device. Note that this version lacks many of the features required in our solution, and is therefore far inferior to our proposed locking device.

However, this naïve approach is fraught with issues. The pointer who has the key in his or her possession may initiate an ownership transfer, either wanted or unwanted (i.e., in cases of theft). Moreover, borrowing re-introduces the same race condition that the protocol had attempted to eliminate. Thus, anyone with a copy of the key to the hëävy m̈ëtäl locking device may be the future Baby Daddy, which again leaves the question unresolved: Who's Your Daddy?

4.2 Solution 2: Hëävy M̈ëtäl Locking Device with Biometric Key

This solution is similar to the first, except the pointer is the access key. However, this means that once a particular pointer has accessed the resource, she may henceforth never be accessed by any other pointer. This is problematic if the first pointer gets bored and leaves, or is otherwise destroyed.

4.3 Solution 3: Hëävy M̈ëtäl Locking Device with Dynamic Biometric Key, aka She Crimped A Ladysman AndShit (SCALA)

This is similar to Solution 2, except now the locking device can be re-keyed by the external party to any particular pointer, assuming that the resource is in the vacant state (which can always be determined once the incubation period has passed).

4.3.1 Translation to Lay Speak

Now, dis shit gettin' futuristic and shit yo. Now, to get wit dis bitch, you gots to go to her mama and daddy and get permission and shit. If she ain't wit nobody and she ain't pregan' then they crimp yo junk in computer shit an make it so no other playa get in dem locked panties and shit. Na'i mean? That shit mothafuckin sucks, brace yoself foo' cuz dat shit mothafuckin stings an dey know it was you when shit happen.

4.4 SCALA Solves Everything

The solution is clearly SCALA[4] with Uniqueness Types. We draw here on previous work which also considered the urban implications of SCALA [3].

5. Conclusions and Future Work

We have shown that research that problems in multi-threaded programs solves can also to serious real-world scenarios be applied. With a minor modification to existing technology (the metal locking device), have we demonstrated how to solve a wide array of problems in popular socio-scientific documentaries seen. In particular, with our solution, can we always definitively that key question answer, "Who's Yo Daddy?"

Yo so this is the mothafuckin straight deal yo. Dey take da shit dey do on computers and shit and fuckin make it so yo badonkadonkical ho can't gets with no otha balla. So when she sez you da baby daddy you jus gotta say to the otha playa, "Drop yo draws boy I know yo junk be crimped."

6. Acknowledgements

Herr Doctor Doktor Klaus provided all of the middle-English-like sentences. The other authors thank him highly for this immense contribution, which would not be possible without a native German speaker. Ebonics translation gracefully provided by Frau Hezza (aka Hedair). Thanks greatly to the fucking wild input from Dr. Jenn B. S.

References

[1] Bierhoff, Kevin, Nels E. Beckman and Jonathan Aldrich. Practical API Protocol Checking with Access Permissions. In *23rd European Conference on Object-Oriented Programming (ECOOP)*, 2009.

[2] Haller, Philipp and Martin Odersky. Capabilities for Uniqueness and Borrowing. In *24th European Conference on Object-Oriented Programming (ECOOP)*, 2010.

[3] PhD, Malayeri, Donna Dr. "Pimp My Library" Is An Affront To Pimpers Of The World, Everywhere. In *Proceedings of SIGBOVK 2010*, Pittsburgh, PA, USA, 2010.

[4] She Crimped A Ladysman AndShit

SIGBOVIK 2011: TRACK FOUR

Military-Industrial Complexity Theory

An Objection to “ An Objection to “An Objection to “The Box and Circles Plot” ” ”

To whom it may concern,

I strongly object to the previous letter. While pretending to critique the self-promotional meta-humor of the preceding paper, it is – itself – a sort of self-promotional meta-humor.

Besides, the notion of a chain of objections to objections is clearly lifted from the excellent “Monty Python’s Flying Circus” sketch program, which – to be honest – did it better forty years ago. (In part because their letters often ended with a humorous author name or affiliation.)

Love,
Gen. Reginald F. Stout, (Mrs.)

Paper 4: An Objection to "An Objection to "An Objection to "The Box and Circles Plot""""

REVIEWER: Lohn Jongwood
OVERALL RATING: -2 (reject)
REVIEWER'S CONFIDENCE: 4 (expert)

This objection to the objection to the objection to the box and circles plot should have little place in any publication. As any good metatheoretician should know, meta-meta-humor is no humor at all, and the previous letter made no attempts at such content.

Besides, if a chain of objections is to be so avoided on principle, as Mrs. Stout indicates, on what grounds is her own objection any more worthwhile?

Going to the Mall Is Like, Totally NP-Complete

Brittany (AIM: BiEberGrL21348)
Bethany (sparkles3389@hotmail.com)
Tiffany (myspace.com/twilightXtiff)*

SIGBOVIK 2011

Abstract

You mean like, abstract art? Ugh I hate art class. The teacher is like a million years old. I think she's like from Russia or something wherever that is and she doesn't let us text in class!

1 Introduction

Ok so like it's Saturday and this week has been a total bummer! Mrs. Davidson said we had to read 50 pages this weekend *and* write a three paragraph reaction! And we have to use punctuation and proper spelling? Seriously isnt it like the 20th century or something? Who the hell still rights like that? Then Britt's mom grounded her because she thought she was sexting! I'm like mom, I was just in the bathroom taking a picture for my facebook! And only grownups say sexting! God everyone sucks. Tiff's dad took away her iPod because she was listening to that "Fuck You" song! He was all like, this is a Christian household! Thats probably why her parents only had sex once and it was probably like before she was born.

2 Proof

So we took Bethany's mom's minivan and we're going to the mall!! She's the only one that has her permit, but we help watch the road while she's texting. Anyway we get to the mall and we do some shopping. We kinda follow this one cute boy but he goes into Gamestop and we're like EWW NERD!!! Then we're in the food court and we have this realization. The mall kicks ass because there are **N**o **P**arents! And Britt just found the perfect belt so now her outfit is **Complete**! I think we're supposed to say "Q.E.D." but that's latino or something, whatever.

3 Conclusion

Tiff just stole my junior prom date! omg shes such a bitch!!!!!!

* We needed someone to type this up so we paid some guy $5. He's a huge dweeb but for another $5 we let him put his name here. Matt Sarnoff (computers_are_1@me.com) What a loser! Omg who the hell still uses email!!!

Optimal Image Compression

James McCann*
Adobe Systems, Inc.

Notrealy A. Coauthor
Noplace Much

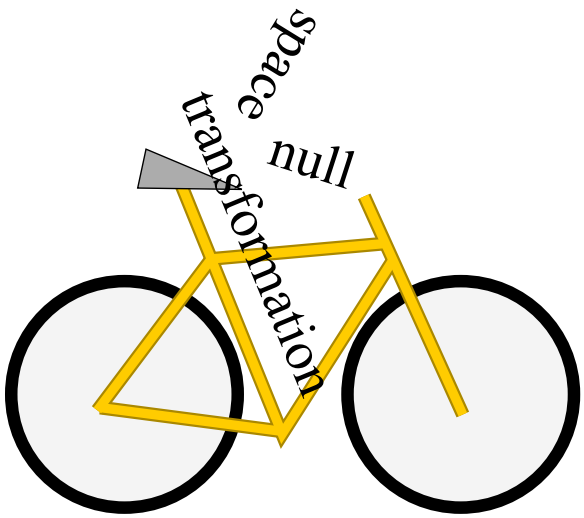


Figure 1: *Our null space transformation theory cyclically commutes on a hot orange-yellow bicycle, even when it's raining.*

Abstract

Data compression, particularly of image data, is an important application domain in computer science; much of the data transmitted on data networks today consists of image data or their wiggly cousins, video data. Thus, any data reduction in the data size of these data items can be of immediate cost savings for data network and data server operators and a time savings for data viewers.

However, current image compression literature is generally focused on heuristics and approximations rather than on optimality. This paper rectifies that deficiency, providing an image compression algorithm that is space- and time- optimal, both asymptotically and in practice.

CR Categories: C.d.b [Images]: Compression—Systems and results E.s.a.b.z.b [Systems]: Practice—Image transmission

1 Introduction

Image compression has been one of the enabling technologies of our current networked lifestyle; without it, we wouldn't be able to share pictures and video with our friends (and strangers who are our friends on facebook for some reason). However, despite the importance of compression algorithms, little progress has been made toward either space- or time-optimal compression.

*e-mail: jmccann@adobe.com

In this paper, we describe a compression scheme that is both space- and time- optimal and is easy to describe and implement. This scheme is based on a theory of null space transformation, which we lay out in detail before introducing the final compression operator.

In order to evaluate our compression operator, we compare it to various commonly-available compression schemes over a broad test set. We find that – as expected – it performs favorably in compression/decompression time as well as in resulting file size.

2 Background

[Microsoft Corporation 1998]

3 Null Space Transformation

Our method is motivated by a theory of null space transformation, assembled mostly from whole cloth and refined through smoke and mirrors. In this section, we derive our theory in three pairwise steps: null-space transformation, null space-transformation, and null $\frac{\text{space}}{}$ transformation.

These pairwise derivations are enough to demonstrate that our theory cyclically HOB-commutes (Figure 1), which is isomorphic to validity by assertion [Experts 2011].

3.1 Null-Space Transformation

We demonstrate the pairwise validity of our null-space transformation by induction.

Theorem 3.1 (Pairwise Validity A)**.** *A first-second dashed subset of our theory validates exactly.*

Proof: We proceed by induction.

To see why our transformation works, first consider that in cosmetic surgery the simplest form of transformation is removal; thus, removal is the base case.

Furthermore, the null-space of a linear operator is the space of vectors that the operator takes to zero – precisely those vectors which are to be *annulled*. Of course, annulment (which is synonymically tied to removal – our premise) is often difficult without specific dispensation; but such a dispensation can be obtained by a skilled negotiator (a so-called *smooth operator*).

As linear operators are smooth, we are done. □

3.2 Null Space-Transformation

In order to validate the space-transformation form of our theory, we are able to perform a transformation and then appeal to a prior result. We term this form of proof "proof by getting someone else to do the proof without realizing it", and plan to publish a series of human-computation articles about it just as soon as we figure out how to make other people write our papers for us without realizing it.

Theorem 3.2 (Pairwise Validity B)**.** *A second-third dashed subset of our theory validates exclusively.*

Proof: In order to properly demonstrate the futility of space-transformation, we proceed by contradiction.

Consider the existence of a space transformation. If such a thing did exist, then it would also be known as a space *warp*. Vargomax [2007] recently demonstrated that such warps can by characterized by a $\infty \times \infty$ fantasy land and graph coloring, from which he concluded,

> "Welcome to warp zone!"

Therefore, as a consequence, of course, consequently, in fact, consummately, it remains, indeed, ultimately, henceforth, i.e., overtly, by contradiction, proceeding onward, entirely, unequivocally, without equal, understandably, also, it appears, finally, thus. □

Extra case: the authors have observed that, on occasion, a proved theorem will become infected with agent α and revive. These *zombie* theorems can be hard to dispatch again without a shotgun approach. Thus, in the interest of the reader's safety, we provide an extra case of ammunition:

□ □ □ □ □ □ □ □ □ □ □ □ □

For the readers' sake, we hope that – when the infection comes – these shells either serve to enforce a lasting truth or a that counterexample is close at hand.

3.3 Null $\frac{\text{Space}}{}$ Transformation

We have saved the null $\frac{\text{space}}{}$ transformation sub-case for last because it is the most difficult to validate. Indeed, in order to force our proof through, we will require two lemmas and a Slycan's Lamask. We present the lemmas and their proofs presently. The Lamask, however, is a shy creature, often found in swamplands, jungles, and appendices.

Lemma 3.3 (Midwest Spacing)**.** *The Midwest has plenty of space.*

Proof: The Midwest region of the United States has a temperate climate, and is largely rural, with fields, plains, and woodland. As such, it formed a wonderful habitat for the Common Wood Pigeon (*Crocodylus niloticus*). Unfortunately, due to the aristocracy's demand for pidgeon-leather during the 19th century, the Common Wood Pigeon was hunted nearly to extinction by Mexican fur traders ("Voyageurs") who transported the furs to England in their distinctive canoes.

This paucity of pigeons (or *bird banishment*) leaves the area with a distinct deficit in the local species spectra, so – by the pigeonhole principle – we are done. □

Lemma 3.4 (Parking Spaces)**.** *It is always possible to find a parking space nearby.*

Proof: We proceed by generalization.

Many of those who read papers are of modest means. Many people of this economic stature drive cars.

Therefore, by the principle of the undistributed middle-class, the reader is either driving a car or not. Thus, the proof of this lemma falls into cases:

- If the reader is not driving a car, then – clearly – the car has been parked nearby.

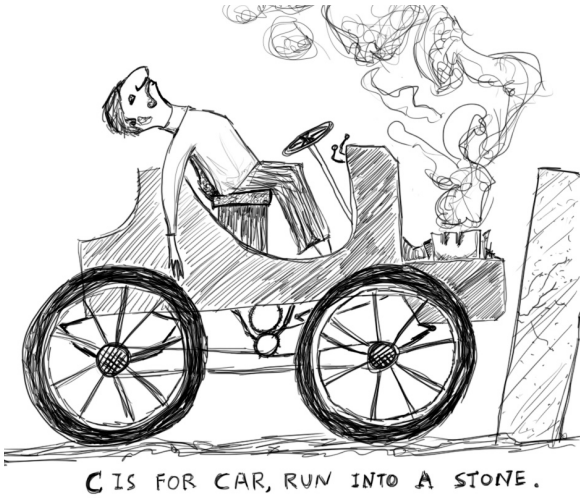


Figure 2: *A Gorey car crash.*

- On the other hand, if the reader is driving an automobile, then they will likely crash (Figure 2); eliminating themselves and, consequently, this case.

□

With these lemmas in hand, and the prospect of seeing a Lamask, we are ready to tackle the main theorem.

Theorem 3.5 (Pairwise Validity C)**.** *A first-third dashed subset of our theory validates ecstatically.*

Proof: In order to demonstrate that underlying all spaces there are no transformations, we proceed by extended binary space partitioning. Particularly, instead of using binary, we use ASCII hexadecimal.

For partition `0x6d696477657374`, our Lemma 3.3 suffices; while for `0x7061726b696e67`, our Lemma 3.4 is conclusive.

Any other partition is – with high-probability – improperly spelled, and thus can be dropped from the proof. Besides, Lamask watching is far more interesting. □

4 Defining the compression and decompression operator

As suggested by our theory, we define our compression operator as the transformation that nulls the data-space of the image. We provide pseudo-code for such an operator (and the associated best-fit decompression) in Figure 3. This code is a straightforward consequence of the theory, and so we do not need to explain it any further.

```
0: NST-Compress(i):
1:     return ∅

0: NST-Decompress(o):
1:     return
```

Figure 3: *Pseudo-code implementing our compression and decompression algorithms. For* `NST-Compress` *i is the input image. For* `NST-Decompress` *o is an output buffer of the proper size initialized to zeros.*

Method	comp. (sec)	dec. (sec)	size (bytes)
NST	0	0	0
*	> 0	> 0	> 0

Table 1: *Runtime and compressed image size for all known image compression schemes, as compared to NST.*

4.1 Reference Implementation

We provide source code of a reference implementation of our compressor in Figure 5. We feel that while small improvements could perhaps be gleaned by hand-optimization, overall, our compression/decompression speed is already quite good.

To compress, run

```
./nst < input-image > compressed
```

To decompress, prepare a blank image of the proper size, then run

```
./nst < compressed >> blank-image
```

In either case, the utility will deduce the proper file format before performing the compression or inserting decompressed information into the blank image. All image file formats are supported.

5 Comparison to Other Methods

Our compression algorithm is theoretically both time and space optimal, requiring zero work to compress or decompress images, and zero bytes to represent compressed images. However, in order confirm these theoretical results, we performed an exhaustive comparison of all known compression formats to our NST compression. We used a broad test set of images (Figure 4).

The results of this comparison are shown in Table 1, but can be briefly summarized as follows:

We, like, totally owned them, yo.

Or, more simply:

WLTOTY

Figure 4: *Our broad test set. Images copyrighted by flickr users piecesofalice and Nick Nunns, used under the creative-commons-attribution license.*

6 Conclusions and Future Work

In this paper, we introduced and rigorously validated a theory of null space transformation. Furthermore, we used this theory to develop a provably optimal image compression and decompression scheme which far exceeds the capabilities of other common compression methods.

Of course, this optimality is not without a slight cost. For many images, our compression scheme (much like the popular "JPEG" scheme) is somewhat lossy. Despite this, we feel that the optimality of the algorithm more than makes up for this slight deficiency.

References

EXPERTS. 2011. It's true. *Trust us* (Really).

MICROSOFT CORPORATION. 1998. Windows 98. *Computer Software* (June).

VARGOMAX, V. V. 2007. Generalized super mario bros. is np-complete.

A There may be a Lamask nearby

Stay quiet and move slowly, for they are flighty creatures.

Eh? What does one look like? Well, I have not, myself, seen one; however, I have conversed at length with a mathematician – quiet fool! step carefully – by the name of Johansensumson who was working on a theorem in the deep Cambodian jungles in 1925.

At that time, of course, the Lamask was just a legend, and Johansensumson was not entirely sure whether what he heard from the locals was well-founded or mere conjecture.

So during his second expedition, he baited a succulent theorem – just as we have done here – with a little bit of intuition and a few hand-waves. He waited for hours, working on another line of reasoning, when – suddenly! – it was upon him.

As quickly as it had arrived, it was gone, leaving only a few crumbs and a pile of irrefutable logic. But he never forgot the sight. It was twelve years later that Slycan conclusively documented the Lamask with his charged-plate/oil-drop experiment and Johansensumson was vidicated.

C++:

```
int main() { return 0; }
```

sh:

```
#!/bin/sh
```

perl:

```
#!/usr/bin/perl -w
```

Figure 5: *Reference C++, shell, and perl implementations of a compression/decompression utility. Most Linux distributions already ship a reference implementation under the name* `/bin/true`.

B There it is!

(artist's conception)

C As soon as it was, it was not

The Lamask is a true wonder of mathematics; we are lucky to have seen it, however briefly.

It is just as Johansensumson described it, but also so much more. Forgive me if I begin to weep; mathematicians seldom experience such beauty outside dreams and books.

On unlexable programming languages

Robert J. Simmons

April 1, 2011

Abstract

One of the features of the Perl programming language is that it is formally unparsable. In this article, we consider the design of a programming language that is similarly unlexable, in that the correct separation of a string of characters into lexical tokens is undecidable in general.

1 Introduction

The inspiration for this paper was a factual claim and an opinion, both of which I attribute with some uncertainly to Larry Wall's October 2010 lecture at Carnegie Mellon University. Both claims, however, are also definitively attributed to Jeffrey Kegler. The factual claim is that it Perl 5 is an unparsable language [Keg08a]. The opinion is that this is a good thing [Keg09]. If a language with undecidable parsing is good, a language with undecidable lexing must be freaking awesome. Hence, this paper.

2 An undecidable family of lexers

The recipe for Perl's unparsability is rather straightforward: Perl is unparsable because ``the only way to parse Perl 5 is to run it or to simulate it using a language of equivalent power'' [Keg08b]. Specifically, the use of `BEGIN` blocks can force code evaluation during the compilation phase and the `eval` function can trigger compilation at runtime [Keg08a].

Kegler, who established Perl's formal unparsability, was unamused that ``Perl-bashers'' picked up his result as a criticism of Perl [Keg09], declaring that Perl's unparsability is a good thing, and that, in fact, ``demanding a parseable language is the sign of weak programmer.'' The underlying truth of this statement is that Perl's unparsability means that Perl can have no meaningful notion of abstract syntax, so that it is impossible to contemplate static analyses, factoring tools, or IDE feedback that works in general on Perl programs, since such tools uniformly work on the level of abstract syntax. Perl can only be interpreted, not compiled or, in a certain sense, reasoned about.

How can we apply similar principles to the design of a programming language with undecidable lexing? In Perl, it is the `BEGIN` blocks which bootstrap parsing into potentially problematic Turing tarpits. We define the *lexing problem* to be the process of taking a stream of *characters* and unambiguously returning either an error or a single lexical token and a sub-stream of the original stream. Lexers

_	0	9	0001	j	111	t	00011111	E	10110	P	01011
0	00	a	1	k	11100	u	00000111	F	10101	Q	01010
1	1111111	b	0110	l	001100	v	11011	G	10100	R	01001
2	01	c	1010	m	110101	w	11010	H	10011	S	01000
3	0111	d	0010	n	101	x	11111	I	10010	T	00111
4	1011	e	1110	o	001	y	11110	J	10001	U	00110
5	0011	f	11	p	100	z	11101	K	10000	V	00101
6	1101	g	010	q	000	A	10	L	01111	W	00100
7	0101	h	110	r	10101010	B	11001	M	01110	X	00011
8	1001	i	011	s	11111000	C	11000	N	01101	Y	00010
						D	10111	O	01100	Z	00001

Figure 1: Standard encoding of letters as bitstrings from the literature [Sim11].

deal with potential ambiguity by always selecting the *longest possible lex*; this is how we ensure that that the C token `elsebob` always parses as a single identifier and not as the reserved word `else` followed by the identifier `bob`.

We specify that the lexer returns a sub-stream to ensure that lexing can only read from the character stream, not add to it.[1] In this paper, we will assume that the character stream available to the lexer is a byte sequence - which we intuitively connect to the standard ASCII encoding of characters.[2]

2.1 The Dec/n languages

The parametrized Dec/n family of languages - where n is a Gödel numbering of the three parameters, as yet unspecified - all share the following properties. Tokens such as `*`, `~`, and `^` are lexed individually, but an identifier is a string of alphanumeric characters (plus underscores) *such that no prefix corresponds to a non-terminating lambda calculus expression*. This, combined with the requirement that lexers return the longest possible token, is the reason why any perfect Dec/n lexer must solve the halting problem. For instance, in the Dec/5 language (defined below), the alphanumeric sequence `a1qa1KLaffq01X1uas0foo` parses as two tokens, `a1qa1KLaffq01X1uas0` and `foo`, since the former corresponds to a non-terminating expression.

A particular member of the Dec/n family is defined by three parameters. The first is a way of interpreting a series of alphanumeric characters as a bitstrings (by giving a bitstring encoding for each alphanumeric character in turn).[3] The second is a way of encoding bitstrings as lambda calculus expressions. The third is an evaluation strategy to use to attempt normalization.

2.2 Dec/5

The Dec/5 language is a specific instantiation of the Dec/n family. Characters are encoded as bitstrings using a standard character-by-character encoding from the literature [Sim11], even though this encoding does have the potential disadvantage that many sequences of alphanumeric characters map to the same bitstring. For convenience, we repeat this encoding in Figure 1.

The second piece for instantiating Dec/5 is a way of encoding bitstrings as lambda expressions. We use the encoding from the Jot programming language [Bar]. Jot actually presents itself as a full programming language, not just an encoding of lambda-calculus terms, but its termination behavior is dependent on the evaluation strategy of the host language [Rey72]. This is nevertheless perfect for our purposes: we use Jot merely as one potential encoding of (a subset of) the terms of the untyped lambda-calculus. Dec/5 uses a call-by-value evaluation strategy to attempt to normalize terms to a value.

One useful aspect of the standard encoding of alphanumeric sequences is that any combinator calculus term can be encoded directly as an alphanumeric sequence by writing application (prefix) as ``a'' and the S and K combinators as ``s'' and ``k'' (respectively). For instance, the token `aaksask` can be extended by virtue of the fact that $(KS)(SK)$ is a terminating combinator calculus term [Bar].

3 Implementation

In Figure 2, we can see the signature allowing us to instantiate an (necessarily incomplete) lexer for any Dec/n language. The particular implementation can only prove non-termination by detecting a cycle in evaluation. All divergent terms will fail to be noticed by the implementation and will either cause the implementation to diverge (if `limit` is `NONE`) or else raise an exception.

The implementation is available from `https://bitbucket.org/robsimmons/dec-n`.

4 Conclusion

This paper only scratches the surface of undecidable lexing techniques. Ever since Cohen's seminal work on the area [Coh80], Gödel encodings for programs have received insufficient attention. Our Jot encoding is the most disappointing aspect of the implementation: not only does it fail to capture all lambda calculus terms (merely allowing the simulation of all combinator calculus terms), but the encoding is *boring* from the perspective of non-termination: almost all encodings terminate. As an example, the Dec/n implementation can lex all of Shakespeare's *Othello* (downloaded from Project Gutenberg) under the Dec/5 instantiation without ever finding a non-terminating term.

[1] Investigating ultralexing, where the lexer can modify the character stream, is an exciting direction for future work.

[2] Investigating undecidable UTF encoding techniques is an exciting direction for future work.

[3] Investigating non-compositional encodings of alphanumeric strings as bitstrings is an exciting direction for future work.

```
signature LEX_ARGS = sig

   (* Maximum number steps to look for a cycle or an
    * irreducable expression.
    * NONE - the lexer will not terminate on a divergent sequence
    * SOME n - only for n steps, then raise an exception *)
   val limit: int option

   (* The encoding of a given character as a bitlist.
    * Most-significant-bit first! *)
   val charcode: char -> bool list option

   (* The encoding function that turns bitlists
    * into lambda calculus expressions. *)
   val lambdacode: bool list -> Lambda.exp

   (* A step function for the lambda calculus;
    * NONE implies termination *)
   val step: Lambda.exp -> Lambda.exp option

end
```

Figure 2: The `LEX_ARGS` signature, parameterizing undecidable lexers.

References

[Bar] Chris Barker. Iota and jot: the simplest languages? `http://semarch.linguistics.fas.nyu.edu/barker/Iota/`.

[Coh80] Norman H. Cohen. Gödel numbers: a new approach to structured programming. *SIGPLAN Notices*, 15:70-74, 1980.

[Keg08a] Jeffrey Kegler. Perl is undecidable. *The Perl Review*, 5:7-11, 2008.

[Keg08b] Jeffrey Kegler. Rice's theorem. *The Perl Review*, 4:23-29, 2008.

[Keg09] Jeffrey Kegler. Unparseability is a good thing, 2009. `http://www.perlmonks.org/?node_id=790624`.

[Rey72] John C. Reynolds. Definitional interpreters for higher-order programming languages. In *Proceedings of the ACM annual conference*, pages 717-740. ACM, 1972.

[Sim11] Robert J. Simmons. On unlexable programming languages. In *SIGBOVIK 2011*, pages 79-82. ACH, April 2011.

SIGBOVIK 2011: TRACK FIVE

Developers, Developers, Developers, Developers.

(Continued on next page…)

An Objection to “An Objection to “ An Objection to “An Objection to “The Box and Circles Plot” ” ” ”

SIGBOVIK Committee,

The previous objection, while accurate, did not contain a clear punchline. Such a punchline is often required to terminate a chain of objections, as it can otherwise co.-`Program received signal SIGSEGV, Segmentation fault.`
`Maximum stack depth exceeded at objection.tex:12`
`(gdb)`

Paper 5: An Objection to "An Objection to "An Objection to "An Objection to "The Box and Circles Plot""""

REVIEWER: Reginald F. Stout
OVERALL RATING: 0 (borderline paper)
REVIEWER'S CONFIDENCE: 1 (low)

I'm not sure why I was asked to review this letter, and I'm not really good at this gdb thing, but I'll give it a go.

```
(gdb) backtrace
#0 in SIGSEGV_handler()
#1 in ???
#2 in objection() at objection.tex:12
#3 in objection() at objection.tex:9
#4 in objection() at objection.tex:9
#5 in objection() at objection.tex:9
#6 in box_and_circle()
#7 in SIGBOVIK_handler()
#8 in main()
(gdb) print punchline
$1 = <value optimized out>
(gdb) punch printline
Violence isn't the answer to this one.
(gdb)
```

MLA-style Programming: Fostering More Interest in Programming in English Majors

Richard Ha Echt C. Mmunicationo Mary Langdon Ashford

April 1, 2011

Abstract: In this paper, we address issues that plague the computer science field that disinterest and disincentivize non-technical majors in fields such as English literature from trying to learn a new language that is not English or Latin. We investigates methods to provide new platforms to leverage the synergies between the humanities and computer science to promote better code readability and to get better grades in Technical Communications courses for Computer Scientists.

Keywords: english, sml, techcomm, mla, natural language processing

1 Introduction

Recently, there has been an increasing trend of reports by tech blog writers that the rate of students who choose STEM careers (and computer science in particular) in the country is diminishing every year for complex reasons. A large majority of these bloggers report that students avoid careers in CS because of the media portrayal of "all programmers are geeks" (reason A) and because it is hard for old people to read text on a monitor with their bi-focals on (reason Bi). These bloggers then reach the complex conclusion that the sum of these reasons (A+Bi) is why we cannot have nice things and why those in humanities refuse to learn a new language other than dead languages like Latin, and that somebody should do something about it.

Despite what the bloggers believe, a significant amount of research has been done into figuring out why humanities majors prefer to stay away from the programming field or computer science in general. A variety of solutions have been proposed like tricking children into learning lambda calculus or writing a module for Perl that lets one code in Latin with varying degrees of success in their implementations but none of them seemed to solve the overall problem of the general disinterest in programming.

What this paper achieves to do is to present the research done on the matter and demonstrate that the solution is actually counter-intuitively simple and easy and that its implementation will solve the problem in one swift blow.

2 Preliminary Research

To understand the underlying issues that cause specialists in the humanities to avoid programming, `SOME(data)` was collected from large Internet communities known for their Internet etiquette ("netiquette"), honesty, and well articulated discussions, such as 4chan (pronounced "*quatre*-chan" or "cat-chan", named for its cat enthusiasts). The communities were presented with a simple survey (Appendix A) that asked the users for their academic and professional backgrounds followed by a set of multiple-choice questions designed to provide the minimal amount of useful information possible. The results have been compiled into the following:

95% of surveyors replied "Yes", 2% said "No", 1% said "[x] Genuflect", and 1% replied "Have you read your SICP today?" The remaining 1% made no sense whatsoever.

Clearly the research done by other professionals in the field was all bogus and that this is the reason why English majors do not take up program-

ming: 95% of all English majors who responded to the survey are English majors. This new information blew us authors' minds and made them work feverishly to cure this crippling disease of being an English major.

3 Curing the Problem

Armed with this new knowledge, the authors immediately went to work to help English majors become computer science majors. A number of fixes were attempted such as tricking English majors into registering for and taking `Great Theoretical Ideas in Computer Science`, disguising post-fix programs as arguments made by Thomas Aquinas, and ~~physical violence and psychological warfare~~ asian-style parenting, but most attempted solutions only had varying unacceptable degrees of success.

But there was one solution that stood out amongst the others and managed to help stop a person with the affliction of being an English major from not learning to become a programmer. During trials of using various forms of Turing-complete languages, a group of English majors with no previous background in computer science or mathematics rapidly scored remarkably high compared to their peers and the control groups. This group learned how to use an English-based language to describe to the computer the tasks they wanted the computer to solve in an efficient and competent manner. After a few more follow-up trials, the authors have discovered that if a computer understands commands given to it in simple English, then English majors can become expert programmers.

But the language the English majors were using was a simple language defined by a simple grammar written on a napkin by one of the authors. The English majors were asked to perform some tasks with some known "code is almost English" languages such as Befunge and Ruby but complications arose giving way to complaints such as "Ruby is unreadable." so the authors set out to formally define a new language with a new grammar that the English majors can easily understand. This new language and solution is called: **Simple ML in MLA style** or "SML/MLA".

4 Solution Proposal: Move over New Jersey

The proposed solution is a context-free Turing-complete language that supports different paradigms of programming such as imperative, functional, and cloud-style programming. The goal and purpose of the proposed standard of Simple ML in MLA style is to create a workable and modular standard of code writing that can be understood by all persons with English degrees with the help of the standards of English as proposed by the Modern Language Association of America. The proposal is a work-in-progress and this paper will only cover the main preliminary points that will become the founding principle of the language.

4.1 Quotations

One of the many points of conflict between programmers and English professors is the proper usage of quotations. For example, with the phrase

One commonly known phrase that has all the letters of the alphabet is "The quick brown fox jumped over the lazy dog."

On one side, the programmers insist that period should be outside the quotation so as to not confuse the reader into thinking that there is more to the sentence than shown. On the other, the professors just take off unreasonable amounts of points for not putting the period inside the quotes. This pathological case for this issue is when programmers start inserting in-line quotes of code into their papers and then lose a lot of points because the professors do not agree on the parse structure. Thus, the solution to this issue in the language proposal is that all language delimiters immediately adjacent to the end-quotation character will be placed inside the quotations. For example,

```
printf("Hello World.);"
```

```
fprintf(stdout, ("Hello "
                 "World number %d.)," x);
let
    val teststring = "Hello Worldin"
    print teststring
end;
```

Initial usage trials have shown that English majors are all in favour of this new proposal for quotations. This is clearly a step in the right direction but it is not enough. In order to succeed in converting English majors into becoming programmers, we have to define the entire proposed language as if it's the English language itself!

4.2 Punctuation and Arrays

Since the language is modeled after impeccable American English, all code statements in the language must be ended by a period instead of a semi-colon. Commas will still be used as delimiters for lists unless one of the entries in the list contain a comma. In that case, the comma will be replaced locally in the list by the semi-colon. Also, if a list or array has exactly two elements, then they may be specified with an "and" inbetween the two values and/or variables. The comma before the "and" would be optional for this case.

To take advantage of this new feature, declarations of lists and arrays in the language are declared as demonstrated by the following code block.

```
Construct an array called A with the
following values: "foo," "bar," and "baz."

Give the function Foo an array with the
values of 1, 2, 3; "foo;" and 6.82.
```

4.3 Variable Name and Capitalization

All properly written SML/MLA programs must adhere to proper capitalization rules as defined in the MLA. That is, all sentences (code statements) must start with a capital letter no matter what. Because of the issue of people defining variable names that start with lower-case letters, all symbol names in a program must start with a capital letter. As of the time of writing, the authors have not been able to come up with a valid exception scenario that cannot be solved by rewording the sentence to move the offending lower-case word away from the beginning of the sentence. The capitalization rule is demonstrated in the previous code block where the sample array was named "A" instead of "a".

Symbol names are also allowed to contain spaces provided that the spaces are not at the beginning or end of the name because that would not be good English. Here is a code sample to demonstrate:

```
Assign "Kill Casshern!!" to Kill Casshern.
Assign "Devour Casshern!!" to Devour Casshern.
Construct an array called Dying Robot Chant
with the variables Kill Casshern, and Devour
Casshern.
Give the Dying Robot Chant to the Robot Apocalypse
function.
```

4.4 Code Flow

Loops are very simple in SML/MLA. For a while loop, one just needs to tell the program to loop until a certain condition is met. For a for loop, one just needs to tell the program to loop a certain number of times. And for the sake of code readability, unless specified in line, the program would loop the code in a particular paragraph. Example:

```
While loops:
Until you get a SIGKILL signal, Print
Kill Casshern. Print Devour Casshern.

For loops:
Print Kill Casshern and Devour Casshern 100 times.

Count from 1 to 100, and every time you count,
print Kill Casshern and Devour Casshern.
```

4.5 Other MLA features

4.5.1 Spacing

All programs must be double-spaced and written in 11pt or 12pt Times New Roman fonts. This is one of the most important points of the MLA that implementations of SML/MLA must adhere to.

4.5.2 Functions and Bolding

All top-level functions within a file are the equivalent of Level 1 Headings. This means that the name of the functions and the type of the value(s) they are returning must be bold and flush left, and also enumerated in the order that the function is declared in the file.

4.5.3 Header File Inclusion

To put simply, when files known as 'header files" to most programmers (or signatures, or abstract classes, etc.) are included or referenced to in a program, they must be cited properly and properly noted in a bibliography or references list at the end of the program or library file. Bold characters can be inserted into most available word processors with the shortcut Ctrl + B.

5 Hello World

The following code is what a Hello World program would look like if it's written in a language compliant with SML/MLA.

1. Function Main that returns an integer
It takes in an integer named argc and an argument list of strings called argv.
Ignore argc and argv and print to the terminal the following quote: "Hello World." (stdio.h, printf)
Return zero.

Various Coders. "stdio.h - Standard Buffered Input/Ouput" GNU C Library. Free Software Foundation, Inc.

6 Conclusions and Future Work

With the advent of the new language specification of Simple ML in MLA style, the authors are confident that English majors will start swarming in droves to join the ranks of the programmers because of the ease of adoption of the language and because of how easy it has become to program. The authors are terrible people but it's okay because now everybody and their grandma can code.

There is a lot of work to be done in the future such as finalizing the specification of the language, proving soundness and decidability of the language, and implementing compilers that will compile the SML/MLA programs into fast programs in native code on various platforms.

7 References

1. Russell, Tony, Allen Brizee, and Elizabeth Angeli. "MLA Formatting and Style Guide." The Purdue OWL. Purdue U Writing Lab, 4 Apr. 2010. Web. 28 Mar. 2010.

2. Authors, Various. "The MLA Style Manual" Wikipedia. Wikimedia Foundation Inc., 25 Mar. 2011. 28 Mar. 2011.

3. Berenson, Dmitry. "Socio-Economic Factors and Automated Statistical Analysis of the LOLcat Language!!!1!" SIGBOVIK 2008.

4. 4chan.org, /prog/. 4chan BBS - Programming. http://dis.4chan.org/prog/

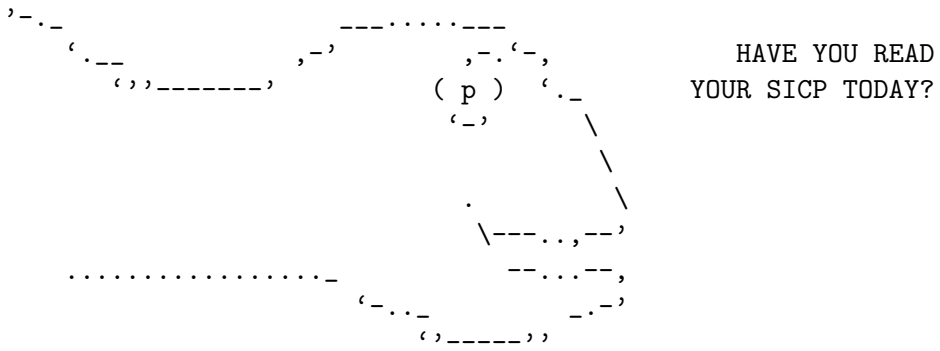
Appendix A

Please check only one answer per question.

1. Are you an English major?
    [ ] Yes
    [ ] No
    [ ] Other: _____

Thank you for filling out this survey!

Appendix B



Theoremin: Proof by Handwaving

Ben Blum (bblum@andrew.cmu.edu)

2011.04.01

Abstract

We present a method for extending proof systems to support an additional lemma, Handwave. The theremin serves as a hardware platform for converting hand-waving efforts on the part of the researcher into a representative waveform; special software, called Theoremin, then analyses the waveform to determine if the hand-wave was sufficiently convincing for a given inference. Using Theoremin, a theorem-proving program may easily be extended to incorporate the Handwave lemma into its logical system. We provide a simple "sound"ness proof (have you ever *listened* to a theremin?) and several example derivations which have been vastly simplified by use of the new lemma.

Quantitative Comparison of Typesetting Software: TEXShop vs. TechShop

Matthew Sarnoff*
Department of Advanced Alcoholic Research
The Boom Boom Room, 1601 Fillmore Street, San Francisco, CA 94115

BoviKon East 2011

Keywords: typesetting, Knuth-Liang algorithm, user interface design, human-computer interaction, human-computer attraction, human-computer courtship, sexual turing test

1 TEXShop

TEXShop (regrettably styled as TeXShop[1]) is a full-featured TEX editor and previewer for Mac OS X.[2] I used it to write up my homework back in college after asking Alan V for the answers.

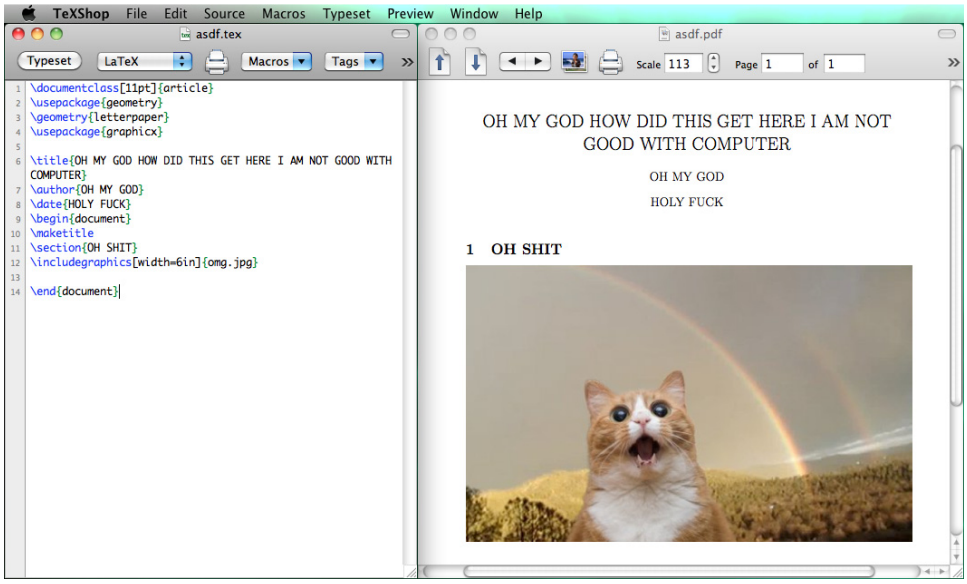


Figure 1: Screenshot of TEXShop on Mac OS X 10.9 "Lol Cat." Note the presence of modern features like syntax highlighting and scroll bars.

*ε-mail: computers_are_l@me.com, twittums: @autorelease

[1] "Thou shalt not render the Name of the Holy Software without subscript or proper kerning. To do so is an abomination: I am the KNUTH." (Knuthviticus 31:14)

[2] Rhymes with TEX. You lose all credibility by pronouncing it "Mac OS Ecks" or, god forbid, "Mac OS Ten."

2 TechShop

TechShop is a chain of member-based workshops that lets people of all skill levels come in and use industrial tools and equipment to build their own projects. They have locations in California, North Carolina, and Michigan.[3] It is not typesetting software.

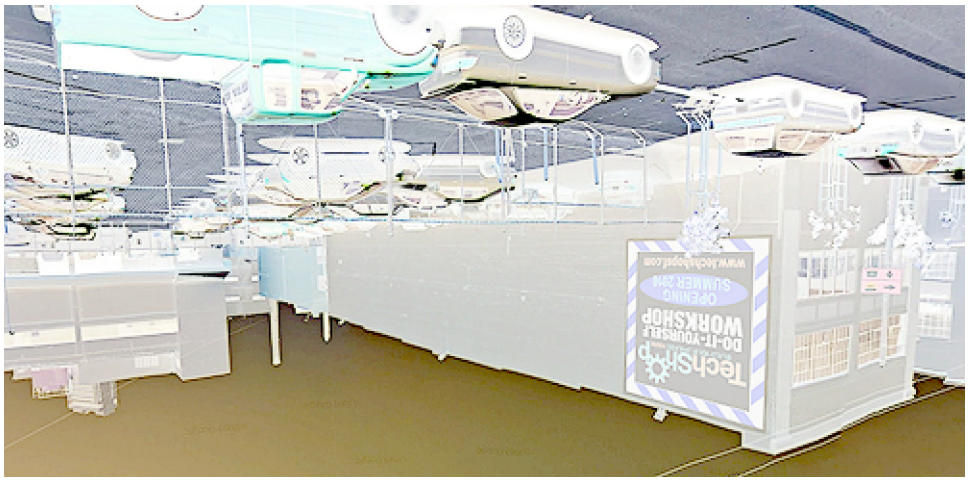


Figure 2: A photo of the TechShop location in San Francisco. It has been rotated 180° *and the colours have been inverted to seem hip and edgy.*

3 Similarities

They are both pronounced the same way.

4 Differences

One can typeset documents, the other can't. One is an actual shop, the other isn't. This is stupid.

5 Conclusion

I'm bored. Who wants to go out tonight?

References, passed by reference

```
citeReference(&wikipedia);
citeReference(&techShopWebsite);
```

[3] According to Wikipedia. It's way too expensive so I've never been there.[4]

[4] Can you add a footnote to a footnote? Apparently yes.

Postscript

```
%!PS
/fillstroke
{ gsave 1 setgray fill grestore stroke } def

/circle
{ 0 360 arc } def

newpath
  100 120 moveto
  250 120 lineto
  250 105 15 270 90 arc
  250 90 moveto
  100 90 lineto
stroke
100 100 25 circle fillstroke
120 80 25 circle fillstroke
showpage
```

Med School, CS Grad School, Both, or Neither?

Brian Hirshman

Med school and CS grad school aren't that different? Or are they? For the following sets of questions, answer "med school", "CS grad school", "both", or "neither". Chances are, you'll learn something in the process – even if that something is that you're now rethinking your original career decision. The grass is always greener on the other side!

Acronyms and Mnemonics:

1. SNAFU
2. RBC
3. BIOS
4. OS
5. MRI
6. CAT
7. LR6SO4AR3
8. CPP
9. LOLWUT
10. Doesn't matter, you're probably looking it up on Wikipedia anyway.

Work and studying:

1. Crash and compile is a valid approach to problem solving
2. Some reading materials make more sense after a beer or two
3. The only way to make some reading materials to make sense is after a beer or two
4. Half of the things you read about are named for a dead guy
5. You've taken a test using pencil and paper
6. You're being paid to be there
7. You spend at least half your time staring at a computer screen
8. You've dissected a field-relevant specimen
9. You were grossed out by said dissection
10. Professors actually make a pretense of teaching you

Research:

1. You're expected to do research.
2. At least some of your day is spent trying to avoid Question 1.

3. Most of your day is spent trying to avoid Question 1.
4. You lied in your answer to either Question 2 or Question 3.
5. You're expected to publish in order to end up in your next good job
6. Your work has been cited somewhere!
7. Your work has been cited, but really it was you who cited it ☹
8. The first two years of your graduate education are the least important
9. Publish or perish
10. What research?

Your field:

1. People do really, really, really important things in your field!
2. You have the potential to impact millions of lives.
3. No really, you have the potential to impact millions of lives!
4. You do really, really, really important things in your field
5. You decided what you wanted to be in kindergarten and stuck with it
6. You never decided to do this, but somehow ended up here?
7. You already regret going into you flied
8. You don't regret going into your field, but you might in a few years
9. Your expectations of what the field was like were shattered on your first day there.
10. You're actually lying about your answer to question #9.

Peers:

1. Someone who started in your year is smarter than you, and you know it.
2. Someone who started in your year is smarter than you, and you don't know it.
3. Someone who started in your year is smarter than you, and you didn't think so, but then they totally proved you wrong by doing _____ better than you'll ever do.
4. You're lying in one of your answers to the first three questions.
5. If your grades are at the top of your class, you aren't doing enough research.
6. At least one member of every class you take already has seen the material before.
7. You know a person who started your program and was legally unable to drink.
8. You're reassured when you know you're not the oldest person in your program
9. You look at the people who are starting your program next year and feel old.
10. When a peer tells you they got a 2390 SAT, you *know* that's 800 points too high

Sleep:

1. Wait, you actually sleep?
2. Wait, you're actually expected to sleep?

3. There are actually administrators in your program telling you to sleep.
4. Things start at 8AM, but only for the first two years or so.
5. You've slept through class, multiple times.
6. You've slept in class, multiple times.
7. You've had at least one class about sleep.
8. The only sleep you know of involves a UNIX command.
9. You've woken up in terror thinking about something you learned recently.
10. There will be many sleepless nights in your future even after you get your degree.

Comics, websites, and email:

1. People around you read XKCD.
2. People around you read PhD comics.
3. People around you don't read comics because they're lame.
4. People around you read Slashdot.
5. People around you read reddit.
6. You suspect that at least one of your professors doesn't actually use the internet.
7. You know that at least one of your professors doesn't actually use the internet.
8. You know that at least one of your professors doesn't have a computer in his office.
9. You're expected to be constantly checking you email.
10. You feel naked without a smart phone.

Life:

1. …
2. …
3. …
4. …
5. …
6. …
7. …
8. Sometimes you feel like you're never going to make it through this.
9. People have mistakenly called you "doctor" already.
10. What life?

Answers:

Trivial, and left as an exercise to the reader.

SIGBOVIK 2011: TRACK SIX

Future Work