

Optimal degeneracy through OwO based variable names

Fwans Skawman, OwOlav

March 31, 2022

Abstwact

Wwiting untouchabwe code can be vewy benefwishial fow exampew fow job secuwity (UwU). One waw of wwiting unweadabwe and untouchabwe cowde is thwough insewting unweasonabwe amounts of UwUs and OwOs in wawiwabwe names. In this papew we pwesent a novew method fow genewating degenewate wawiwabwe names fow most wawiwabwes of length $n > 3$. We show optimawity of ouw method using the owo degenewacy theowem.


1 Introduction

If for some reason, you would like to keep others away from your code, for example, for job security purposes, there are several different methods to choose from. There are several techniques one can use for this related to writing unmaintainable code[1], however, in writing unmaintainable code, one runs the risk of being fired for doing just that.

As an alternative method, we propose writing *degenerate code*, code which is so offputting that others will not want to interact with the code, or the person who wrote it. A very effective method for doing this is through *OwOisation* as has been previously shown [2] experimentally.

For maximum effect, it is clear that one would like to use an optimal amount of degeneracy for each variable used in the code. The rest of this paper introduces methods for maximizing degeneracy in variable names. We prove that our method is optimal for many lengths of identifiers.

2 Theory

The use of OwO is typically used in furry communities to denote a surprised facial expression[3]. Several variations exist, in particular where the Os are replaced with other characters to signify other emotions. Examples of this include: unimpressed (UwU), mischevious (ÖwÖ), pirate (ØwØ), and of course, cursed OwO ()¹

Previous work has shown that for identifiers of length $n = 3$ the maximum possible degeneracy is achieved using the strings OwO, UwU or other variations of the eye-mouth concept[4].

3 Repeated OwO

A natural extension to the work presented in [4] is to repeat the OwOs to reach longer degenerate variable names, for example, for $n = 6$ one might use the variable name OwOUwU, or OwOOwO. Using the OwO-degeneracy-theorem also presented in [4], it is easy to show that this is the optimal amount of degeneracy that can be achieved for $n = 6$.

Naturally, $n = 6$ is no special case; we can extend this method to any n where $n \bmod 3 = 0$.

4 Advanced OwO

For cases where n is not divisible by three, more creativity is needed. While one could argue that variable names such as OwOO are quite degenerate, no proof

¹Unfortunately, this is hard to use in languages which do not support emoji in identifiers. Pick your language carefully

for optimal degeneracy exists. In order to reach optimality, we must turn to more exotic areas of the animal kingdom². It is well known that spiders have multiple eyes, therefore one could form a new form of OwO: OOwOO. In day to day use, this variation might be used by a spider furry to exclaim excitement over a new insect stuck in its net, i.e. “OOwOO what’s this?”.

We claim that such OwO extensions also satisfy the OwO-degeneracy-theorem and through this, it is trivial to prove the optimality of the spider-OwO for $n = 5$.

Luckily, the animal kingdom provides more exotic eye configurations. For example, scorpions have 4 groups of eyes, with 3, 2, 2, and 3 eyes respectively[5]. This allows the construction of a scorpion OwO: OOO_OOwOO_OOO. Again, extending the OwO-degeneracy-theorem, we trivially show that this reaches an optimal amount of degeneracy for a variable of length 13.

5 Advanced OwO combinations

Naturally, we can also combine these new OwOs in the same manner as described in Section 3. For example, we can construct a degenerate variable of length 8 using UwUOOwOO.

Using the previously defined degenerate variables, we can create optimally degenerate variables of $n = 12$ as OwOOwOOwOOwO, and for $n = 11$ as UwUOOwOOUwU. As we now have optimal degeneracy for $n = 11, 12$ and 13 , we can construct optimal degeneracy for any $n \geq 11$ by simply appending more 3 character OwOs. \square

6 Conclusions and future work

We present a novel method for maximizing degeneracy in variable names through combinations and extensions of OwO. We show methods for generating optimally degenerate variable names for $n = 3$, $n = 5$ and $n = 13$. We also show how these can be combined to form optimal degenerate identifiers

for n when n is divisible by the above numbers, as well as for arbitrary sums of those numbers. Finally, we show how to generate optimal degeneracy for any variable of length $n \geq 11$

Unfortunately, we do not yet have any known optimal degenerate variable names for other n . $n = 4$ in particular is of great interest as finding an optimally degenerate name for it would give a constructive method for optimal OwO for all $n \geq 3$.

References

- [1] R. Green, “unmaintainable code,” online. [Online]. Available: <https://www.mindprod.com/jgloss/unmain.html>
- [2] “The abstract of this paper,” did you really want to continue reading after reading the abstract? Of course not.
- [3] M. Rivers and et.al., “Notices bulge / OwO what’s this?” online, 2016. [Online]. Available: <https://knowyourmeme.com/memes/notices-bulge-owo-whats-this>
- [4] You, “Optimal degeneracy for variables of length $n = 3$,” The future, writing the previous work is left as an exercise to the reader.
- [5] Veritasium, “Why are scorpions fluorescent?” Sep. 2021. [Online]. Available: <https://www.youtube.com/watch?v=f-Nr2z5X7Rs>

²and thus also the furry kingdom