A lower bound on the length of the shortest superpattern

Jay Pantone

October 22, 2018

Abstract

This is a rewriting of the proof posted anonymously here:

http://mathsci.wikia.com/wiki/The_Haruhi_Problem

1 A warmup — $|W| \geq n! + (n-1)$

For a word given word w over the alphabet $[n] = \{1, \ldots, n\}$,

$$\begin{aligned} |w| &= \text{the length of } w, \\ N_0(w) &= \text{the number of unique permutations of } [n] \text{ that appear (consecutively) in } w, \\ D(w) &= |w| - N_0(w) \end{aligned}$$

For any word w of length n we have $N_0(w) \leq 1$. Rearranging this gives

$$D(w) \geq n - 1.$$

Let $a \in [n]$ and define the concatenation $w' = wa$. Clearly, $|w'| = |w| + 1$ and $N_0(w') \leq N_0(w) + 1$ and so

$$D(w') \geq D(w).$$

Therefore, if W is a superpermutation, we must have

$$n - 1 \leq D(W) = |W| - N_0(W) = |W| - n!$$

and hence

$$|W| \geq n! + (n-1).$$

2 A brisk walk — $|W| \geq n! + (n-1)! + (n-2)$

A cycle of permutations of $[n]$ is a set of n permutations, each of which can be formed by taking the first letter of a word and moving it to the end. For example, $\{12345, 23451, 34512, 45123, 51234\}$ is a cycle. The $n!$ permutations of $[n]$ can be partitioned disjointly into $(n-1)!$ cycles, each of size n.

We deviate slightly from the proof of the lower bound that appears online in order to emphasize the similarity between this case and the next one. Define a 1-loop to be a cycle with a distinguished permutation called its *entry point*. There are $n!$ 1-loops, each of size n.

In the process of building a word w over the alphabet $[n]$ letter-by-letter, a *k-step* consists of concatenating some word u of length k to the end of w such that the last n letters of wu form a permutation of $[n]$ but the same is not true for any proper prefix of u. For example, given the word $w = 21345$, we can take a 1-step to form $w = 213452$, or a 2-step to form $w = 2134512$ (note that $w = 2134521$ *cannot* be formed by taking a 2-step), or we can take three different 3-steps, etc.

Taking a 1-step keeps you in the same 1-loop, while taking a 2-step must move you to a different 1-loop, as the resulting permutation cannot be in the same cycle as the previous one.[1]

For a word w over the alphabet $[n]$, define

$$\begin{aligned}
|w| &= \text{the length of } w, \\
N_0(w) &= \text{the number of unique permutations of } [n] \text{ that appear (consecutively) in } w, \\
N_1(w) &= \text{the number of 1-loops that have been encountered so far}, \\
D(w) &= |w| - (N_0(w) + N_1(w))
\end{aligned}$$

Note that $N_1(w)$ doesn't just count the 1-loops that are completely traversed, just those that are entered. Any permutations that were not traversed must be part of the superpermutation later (otherwise it's not a superpermtuation)—either the same 1-loop is later re-entered (by definition, at the same entry point), or (more likely) the copy of the untraversed permutations residing in another 1-loop is traversed.

Now, for a word w of length n, we clearly have that $|w| = n$, $N_0(w) \leq 1$, and $N_1(w) \leq 1$, and so

$$D(w) \geq n - 2.$$

The induction step requires us to show that if w' is formed from w by taking a single k-step, then $D(w') \geq D(w)$. First observe that $|w'| = |w| + k$ and that $N_0(w')$ and $N_1(w')$ can each be at most one more than $N_0(w)$ and $N_1(w)$, respectively. Put plainly, $N_0 + N_1$ can go up by at most 2 for each k-step taken.

If $k \geq 2$ then clearly $D(w') \geq D(w)$, leaving us only to check that when $k = 1$ we cannot have that both N_0 and N_1 increase by 1 simultaneously. This is obvious, because 1-steps cannot move us to a new 1-loop.

As a result, $D(w') \geq D(w)$, and so for any superpermutation W,

$$n - 2 \leq D(W) = |W| - (N_0(W) + N_1(W)) = |W| - n! - N_1(W).$$

How many 1-loops must a superpermutation visit? We don't know for sure, but since each 1-loop contains only n permutations, any superpermutation must visit at least $(n-1)!$. Thus,

$$n - 2 \leq |W| - n! - (n-1)!$$

and so

$$|W| \geq n! + (n-1)! + (n-2).$$

3 A light jog — $|W| \geq n! + (n-1)! + (n-2)! + (n-3)$

We first need to define a 2-loop of permutations over $[n]$, and the definition is very carefully calibrated to make this whole argument work. A 2-loop consists of n 1-loops linked together such that the entry point of the $(i+1)$st 1-loop is reached by a 2-step from the permutation in the ith 1-loop which is exactly $n-1$ 1-steps from its own entry point. A picture is much more helpful, so we've drawn below a 2-loop over $[5]$ with 20 elements.

$$\begin{tikzcd}
\mathbf{12345} \arrow[r] & 23451 \arrow[r] & 34512 \arrow[r] & 45123 \arrow[r] & 51234 \arrow[d] \\
52341 \arrow[d] & 15234 \arrow[l] & 41523 \arrow[l] & 34152 \arrow[l] & \mathbf{23415} \arrow[l] \\
\mathbf{34125} \arrow[r] & 41253 \arrow[r] & 12534 \arrow[r] & 25341 \arrow[r] & 53412 \arrow[d] \\
54123 \arrow[uuu, bend left] & 35412 \arrow[l] & 23541 \arrow[l] & 12354 \arrow[l] & \mathbf{41235} \arrow[l]
\end{tikzcd}$$

[1]We could be more formal here by declaring that the *state* of a word is defined to be the permutation of its last five letters together with the entry label of the current 1-loop. For now we err in favor of intuition and readability.

The *entry points* of a 2-loop are the entry points of its constituent 1-loops, and so as in the case of 1-loops, two distinct 2-loops may have entries (permutations) in common. In the picture above, the entry points are drawn in blue, the 1-steps are represented by black arrows, and the 2-steps are represented by orange arrows.

While attempting to create a superpermutation, we say that we have *finished a cycle* if our word already contains all permutations in the cycle. (Recall that a 1-loop is a cycle together with a designated entry point.) For a word w over the alphabet $[n]$, define

$$\begin{aligned}
|w| &= \text{the length of } w, \\
N_0(w) &= \text{the number of unique permutations of } [n] \text{ that appear (consecutively) in } w, \\
N_1(w) &= \text{the number of completed cycles} + \left\{ \begin{array}{ll} 1, & \text{if the current permutation is in an uncompleted cycle} \\ 0, & \text{otherwise} \end{array} \right\}, \\
N_2(w) &= \text{the number of 2-loops that have been encountered so far}, \\
D(w) &= |w| - (N_0(w) + N_1(w) + N_2(w))
\end{aligned}$$

Again, our aim is to show that as w is transformed to w' with a single k-step, the value D does not decrease. Actually, that's not true! But we'll see that it's true enough.

If w is transformed to w' with a 1-step, then the new permutation at the end of w' lies in the same cycle as the permutation at the end of w, and so N_1 and N_2 cannot increase. Therefore in this case, D does not decrease.

If a k-step is taken for $k \geq 3$, then since each N_i increased by at most 1, we still have that D does not decrease.

It remains only to consider a 2-step. If at least one of N_0, N_1, and N_2 does not increase, then D does not decrease and everything works out fine. Suppose instead that each of N_0, N_1, and N_2 increases. Define $\pi(w)$ and $\pi(w')$ to be the permutations formed by the last n letters of w and w' respectively. Since N_1 increases, the cycle containing $\pi(w)$ must have just finished. Since N_2 increases, $\pi(w')$ resides in a different 2-loop than $\pi(w)$.

It follows from these two facts that $\pi(w')$ has already been encountered while building the word w (although not necessarily in this 2-loop—we likely encountered $\pi(w')$ in another 2-loop). To see this, stare at 41523 in the image above, and imagine taking a 2-step to the permutation 52314 (not shown, as it's in a different 2-loop). Since the cycle containing 41523 is complete, either 41523 was just encountered for the very first time, or it's been encountered before. If it was just encountered for the very first time, that means that 15234, which must have been previously encountered, was reached not from 41523 but from some other 2-loop. As a result, the active copy of 41523 is in the same 2-loop as the copy of 15234 that was previously encountered, contradicting the fact that the 2-step from 41523 to 52314 visits a new 2-loop for the first time because the 2-loop containing 52314 as an entry point is the same as the one containing 15234 as an entry point.

So, finally, we can assume that 41523 been encountered before. This can happen, and in this case D does actually decrease. We'll show that the decrease (of at most 1) is compensated before by a necessary increase of at least 1 in the previous step. Let v be the word from which w was created by a single ℓ-step. Transitioning from v to w, N_0 and N_1 cannot increase, while N_2 could increase but only if $\ell \geq 2$. This shows that

$$D(w) - D(v) \geq 1.$$

and so

$$D(w') - D(v) \geq 0.$$

We finally conclude this tedious argument by pointing out that if W is a superpermutation then $N_0(W) = n!$, $N_1(W) = (n-1)!$ and $N_2(W) \geq (n-2)!$. Following the same logic as in the previous sections, it follows that

$$|W| \geq n! + (n-1)! + (n-2)! + (n-3).$$

4 A sprint? — $|W| \geq n! + (n-1)! + (n-2)! + (n-3)! + (n-4)$??

The concept of loops is a strange one, and could be generalized in several ways. I was unable to make the proof in the previous section work with the alternate definition

$$N_1(w) = \text{the number of 1-loops visited.}$$

The step in the proof dealing with the case where D can actually decrease—but not by more than it increased in the previous step—will certainly be trickier in the next generalization of the bound.

Another complication: while there is only a single 2-step from one permutation, there are always three valid 3-steps. So how would you define a 3-loop?