

Remarks on Privileged Words

Michael Forsyth, Amlesh Jayakumar, and Jeffrey Shallit

School of Computer Science

University of Waterloo

Waterloo, ON N2L 3G1

Canada

{mforsyth,a3jayakumar,shallit}@uwaterloo.ca

June 1, 2018

Abstract

We discuss the notion of privileged word, recently introduced by Peltomäki. A word w is privileged if it is of length ≤ 1 , or has a privileged border that occurs exactly twice in w . We prove the following results: (1) if w^k is privileged for some $k \geq 1$, then w^j is privileged for all $j \geq 0$; (2) the language of privileged words is neither regular nor context-free; (3) there is a linear-time algorithm to check if a given word is privileged; and (4) there are at least $2^{n-5}/n^2$ privileged binary words of length n .

1 Introduction

We say that a word x is a *border* of w if it is both a prefix and a suffix of w .

Peltomäki [4, 5] recently introduced the notion of *privileged word*. A word w is privileged if

- (a) it is of length ≤ 1 , or
- (b) it has a privileged border that appears exactly twice in w .

Here are the first few privileged words over a binary alphabet:

0, 1, 00, 11, 000, 010, 101, 111, 0000, 0110, 1001, 1111, 00000, 00100, 01010, 01110, 10001,

10101, 11011, 11111, 000000, 001100, 010010, 011110, 100001, 101101, 110011, 111111.

An easy induction shows that a^i is privileged for any letter a and $i \geq 0$.

We now recall two results of Peltomäki [4].

Theorem 1. *Let w be privileged.*

- (a) If t is a privileged prefix (resp., suffix) of w , then t is also a suffix (resp., prefix) of w .
- (b) If v is a border of w then v is privileged.

Define the *number of leading a's in w* to be the largest integer n such that a^n is a prefix of w , and similarly for the number of trailing a's. Then we have

Corollary 2. *If w is privileged, then the number of leading a's in w equals the number of trailing a's.*

Proof. Write $w = a^i z a^j$ where z neither begins nor ends in a . Then by Theorem 1 (a) we see that $i \geq j$ and $j \geq i$. \square

We now state a useful lemma.

Lemma 3. *Let w be a nonempty word. Then w is privileged if and only if its longest proper privileged prefix is also a suffix of w .*

Proof. \implies : follows from Theorem 1 (a) above.

\impliedby : Let u be the longest proper privileged prefix of w . Let v be the shortest prefix of w containing exactly two occurrences of u ; this is well-defined since u is a suffix of w . Then v itself is privileged. So either $v = w$, or $|u| < |v| < |w|$ and v is a longer proper privileged prefix of w , a contradiction. \square

We now prove a result on powers and privileged words.

Theorem 4. *Let w be any word and k an integer ≥ 1 . If w^k is privileged, then w^j is privileged for all integers $j \geq 0$.*

Proof. Suppose $k \geq 2$. Then w is a border of w^k , and hence by Theorem 1 (b) we know w is privileged.

It remains to show that if w is privileged, then so is w^j for all $j \geq 0$. We prove this by induction on j . The result is clearly true for $j = 0$ or $j = 1$, so assume $j \geq 2$ and w^{j-1} is privileged.

Let u be the longest proper privileged prefix of w^j . If $|u| \leq |w^{j-1}|$, then u is also a privileged prefix of w^{j-1} . Then Theorem 1 (a) and induction together imply that u is a suffix of w^{j-1} . Then u is also a suffix of w^j , and by Lemma 3 we know w^j is privileged.

Otherwise $|u| > |w^{j-1}|$. Write $u = w^{j-1}y$ for some y , where y is a proper prefix of w . Since $j \geq 2$, we see that y is also a proper prefix of w^{j-1} and hence a proper prefix of u . Thus y is a border of u , and hence, by Theorem 1 (b), y is privileged. Since y is a privileged prefix of w , by Theorem 1 (a), it is also a suffix of w . Write $w = zy$ for some z . By induction we know that w^{j-1} is privileged. Since w^{j-1} is a prefix of u , by Theorem 1 (a), it is also a suffix of u , so there exists x such that $u = xw^{j-1}$. Since $u = w^{j-1}y = xw^{j-1}$, we see that $|x| = |y|$ and x is a proper prefix of w . Thus in fact $x = y$. So $u = yw^{j-1}$. Then

$$w^j = w w^{j-1} = (zy)w^{j-1} = z(yw^{j-1}) = zu,$$

and it follows that u is a suffix of w^j . By Lemma 3, we conclude that w^j is privileged. This completes the induction. \square

2 The set of privileged words

Let Σ be a fixed alphabet and consider \mathcal{P} , the set of privileged words over Σ . We prove here that \mathcal{P} is neither regular nor context-free.

Proposition 5. *If $|\Sigma| \geq 2$, then \mathcal{P} is not regular.*

Proof. Let $0, 1$ be distinct letters in Σ . Assume \mathcal{P} is regular, and consider $L = \mathcal{P} \cap 0^+10^+$. By Corollary 2 we have $L = \{0^n10^n : n \geq 1\}$. By the pumping lemma, L is not regular, and hence neither is \mathcal{P} . \square

Proposition 6. *If $|\Sigma| \geq 2$, then \mathcal{P} is not context-free.*

Proof. Assume \mathcal{P} is context-free, and consider the regular language $R = 0^+10^+110^+$. By a well-known closure property of the context-free languages, $L := \mathcal{P} \cap R$ is context-free. We will now use Ogden's lemma [3] to show that L is not context-free, a contradiction.

We claim that

$$L = \{00^a100^b1100^c : a = c \text{ and } a > b\}.$$

To see this, note that $L \subseteq R$. Thus it suffices to show that a word w of the form $0^{a+1}10^{b+1}110^{c+1}$ word is privileged if and only if $a = c$ and $a > b$.

(\Rightarrow) Since w begins and ends with 0, by Corollary 2, we know that $a + 1 = c + 1$ and so $a = c$. Suppose $b \geq a$. Then $0^{a+1}10^{a+1}$ is a privileged prefix of w , yet it is not a suffix of w . By Theorem 1 (a), w is not privileged. Thus $a > b$.

(\Leftarrow) Let $w = 00^a100^b1100^a$ where $a > b$. Then the longest proper privileged prefix of w is 0^{a+1} , which appears again as a suffix of w . Thus w is privileged.

Now let n be as in Ogden's lemma, and let $w = \underline{0}^n10^{n-1}110^n$, where the first block of n zeros is marked as required by Ogden's lemma. Then there exists some decomposition $w = uxvyz$ where xvy contains at most n 'marked' characters, xy contains at least 1 'marked' character, and $ux^ivy^iz \in L$ for all $i \geq 0$.

We see that if either x or y contain a 1, then ux^0vy^0z will have too few ones, and thus will not be in L . Otherwise, we know x lies entirely in the first block of zeros. If y does not lie in the last block of zeros, then if $i = 0$, we will have $a < c$, so $ux^0vy^0z \notin \mathcal{P} \cap R$. If y does lie in the last block of zeros, then $ux^0vy^0z = 00^{n-j}100^{n-1}1100^{n-k}$ for some $j, k > 0$. Since $n - j \leq n - 1$, we see that $w \notin L$.

Hence no decomposition for w exists with $ux^0vy^0z \in L$, and thus $\mathcal{P} \cap R$ is not context-free. Thus, the language of privileged words is not context-free. \square

3 A linear-time algorithm for determining if a word is privileged

In this section we present an efficient algorithm for determining if a given word is privileged.

Algorithm P:

function CHECK-PRIVILEGED(w)

```

if  $|w| \leq 1$  then
  return True
else
   $T[0] \leftarrow 0$ 
   $p \leftarrow 1$ 
  for  $i = 1$  to  $|w| - 1$  do
     $j \leftarrow T[i - 1]$ 
    while true do
      if  $w[j] = w[i]$  then
         $T[i] \leftarrow j + 1$ 
        if  $T[i] = p$  then
           $p \leftarrow i + 1$ 
        end if
        exit while loop
      else if  $j = 0$  then
         $T[i] \leftarrow 0$ 
        exit while loop
      end if
       $j \leftarrow T[j - 1]$ 
    end while
  end for
  if  $p = |w|$  then
    return True
  else
    return False
  end if
end if
end function

```

Our algorithm is a slightly modified version of the algorithm for building a failure table in the well-known Knuth-Morris-Pratt linear-time string-matching algorithm [2].

Theorem 7. *Algorithm P returns “true” if and only if w is privileged.*

Proof. It is easy to see that if $|w| = 0$ or $|w| = 1$, then w is privileged and the algorithm returns “true”. Otherwise, we consider the value for p at each iteration of the for-loop.

We now claim that at the end of each iteration of the for-loop, p equals the length of the longest privileged prefix of the first $i + 1$ characters of w .

To see the claim, observe that, when entering the first loop we have $p = 1$, and is the longest privileged prefix of the first character of w . This establishes our base case. Otherwise, we assume p is the longest privileged prefix of the first i characters of w at the beginning of the for loop, and prove our claim for the end of this iteration. We note that $T[i]$ represents the length of the longest subword u which is both a prefix and suffix of the first $i + 1$ characters of w (the word “read so far”). If $T[i] = p$, we know u is privileged, and p is increased to

$i + 1$. Since p is increased as soon as this equality is found, this is the first time u is repeated in w , and thus the word read so far is privileged. This proves our claim.

After w has been completely read by our algorithm, p represents the length of the longest privileged prefix of w . The algorithm returns “true” if and only if $p = |w|$, in which case w is privileged. \square

Next, we have

Theorem 8. *Algorithm P runs in $O(n)$ time, where $n = |w|$.*

Proof. Starting with the KMP algorithm, we have added one extra *if* statement in the main loop, allowing this algorithm to run in the same $O(|w|)$ time bound as the original algorithm.

More formally, we consider the number of times the inner while loop is executed, as all else takes constant time. The first time the while loop is executed, $i = 1$ and $j = 0$. Upon each iteration, we see that either

1. i is incremented by 1, and j is incremented by at most 1;
2. j decreases

We see i is incremented by exactly 1 when $w[j] = w[i]$ or $j = 0$, due to moving to the next iteration of the for loop. When $j = 0$, then j will remain 0 beginning the next execution of the while loop. When $w[i] = w[j]$, then j will be set to $j + 1$ in the next execution of the while loop.

If neither of the above cases are fulfilled, we see j is set to $T[j - 1]$, which is known by a property of the failure array to be strictly less than j .

With these cases, we see that either i increases or $i - j$ increases. Since the algorithm terminates when $i = |w| - 1$, i will increase exactly $n - 2$ times, where $n = |w|$. Also, since $j < i$ at each stage of the algorithm, $i - j$ can increase at most $n - 3$ times. Since these are the only possible cases, the while loop will execute no more than $2n - 5$ times. Thus, Algorithm P takes $O(n)$ time to complete. \square

4 A lower bound on the number of privileged binary words

Let $B(n)$ denote the number of privileged binary words of length n .

We observe that if $x = 0^t 1 w 1 0^t$, and w contains no occurrences of 0^t , then x is privileged. By choosing the appropriate value of t , we get our lower bound. First, though, we need a detour into generalized Fibonacci sequences.

We need to count the number of words of length n that contain no occurrence of 0^t . As is well-known [1, p. 269] and easily proved, this is $G_n^{(t)}$, where

$$G_n^{(t)} = \begin{cases} 2^n, & \text{if } 0 \leq n < t; \\ G_{n-1}^{(t)} + G_{n-2}^{(t)} + \cdots + G_{n-t}^{(t)}, & \text{if } n \geq t. \end{cases}$$

We point out that in the case where $t = 2$, this is F_{n+2} , the $(n + 2)$ 'nd Fibonacci number, where $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$.

It is well-known from the theory of linear recurrences that

$$G_n^{(t)} = \Theta(\gamma_t^n),$$

where $1 < \gamma_t < 2$ is the root of the equation $x^t - x^{t-1} - \dots - x - 1 = 0$. Since $\gamma_t^t - \gamma_t^{t-1} - \dots - \gamma_t - 1 = 0$, multiplying by $\gamma_t - 1$ we get $\gamma_t^{t+1} - 2\gamma_t^t + 1 = 0$, so $\gamma_t = 2 - \gamma_t^{-t}$.

The next step is to find a good lower bound on γ_t .

Lemma 9. *Let $s \geq 2$ be an integer and let β be a real number with $0 \leq \beta \leq \frac{6}{s}$. Then*

$$2^s - \beta s 2^{s-1} \leq (2 - \beta)^s.$$

Proof. For $s = 2$, the claim is $4 - 4\beta \leq (2 - \beta)^2 = 4 - 4\beta + \beta^2$. Otherwise, assume $s \geq 3$. The result is clearly true for $\beta = 0$, so assume $\beta > 0$. By the binomial formula, we have

$$\begin{aligned} (2 - \beta)^s &= \sum_{0 \leq i \leq s} 2^{s-i} (-\beta)^i \binom{s}{i} \\ &= 2^s - \beta s 2^{s-1} + \sum_{2 \leq i \leq s} 2^{s-i} (-\beta)^i \binom{s}{i} \\ &= 2^s - \beta s 2^{s-1} + \sum_{1 \leq j \leq (s-1)/2} \left(2^{s-2j} \beta^{2j} \binom{s}{2j} - 2^{s-2j-1} \beta^{2j+1} \binom{s}{2j+1} \right) \quad (1) \\ &\quad + \begin{cases} \beta^s, & \text{if } s \text{ even;} \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

It therefore suffices to show that each term of the sum (1) is positive, or, equivalently, that

$$2^{s-2j} \beta^{2j} \binom{s}{2j} \geq 2^{s-2j-1} \beta^{2j+1} \binom{s}{2j+1}.$$

for $1 \leq j \leq (s-1)/2$.

Now $\beta \leq \frac{6}{s}$ by hypothesis, so $\beta \leq \frac{6}{s-2}$. Hence $\beta s - 2\beta \leq 6$. Adding $2\beta - 2$ to both sides we get $\beta s - 2 \leq 4 + 2\beta$, and so $\frac{\beta s - 2}{2 + \beta} \leq 2$. If $i \geq 2 \geq \frac{\beta s - 2}{2 + \beta}$ then $(2 + \beta)i \geq \beta s - 2$, so $2(i + 1) \geq \beta(s - i)$, and

$$\frac{2}{\beta} \geq \frac{s - i}{i + 1} = \frac{\binom{s}{i+1}}{\binom{s}{i}}.$$

Thus $2 \binom{s}{i} \geq \beta \binom{s}{i+1}$. Let $i = 2j$, and multiply both sides by $2^{s-2j} \beta^{2j}$ to get $2^{s-2j} \beta^{2j} \binom{s}{2j} \geq 2^{s-2j-1} \beta^{2j+1} \binom{s}{2j+1}$, which is what we needed. \square

Theorem 10. Let $t \geq 2$ be an integer and define

$$\alpha_t = 2 - \frac{1}{2^t - \frac{t}{2} - \frac{t^2}{2^t}}.$$

Then $\alpha_t \leq 2 - \alpha_t^{-t}$.

Proof. It is easy to verify that

$$\frac{3t^2}{4} \geq \frac{t^3}{2^t} + \frac{t^4}{2^{2t}}$$

for all real $t \geq 2$. Hence

$$0 \leq \frac{3t^2}{4} - \frac{t^3}{2^t} - \frac{t^4}{2^{2t}},$$

and, adding $t2^{t-1}$ to both sides, we get

$$\begin{aligned} t2^{t-1} &\leq t2^{t-1} + \frac{3t^2}{4} - \frac{t^3}{2^t} - \frac{t^4}{2^{2t}} \\ &= \left(\frac{t}{2} + \frac{t^2}{2^t}\right) \left(2^t - \frac{t}{2} - \frac{t^2}{2^t}\right). \end{aligned}$$

Setting $\beta_t = \frac{1}{2^t - \frac{t}{2} - \frac{t^2}{2^t}}$, we therefore have

$$\beta_t t 2^{t-1} \leq \frac{t}{2} + \frac{t^2}{2^t},$$

or

$$-\beta_t t 2^{t-1} \geq -\frac{t}{2} - \frac{t^2}{2^t}.$$

Add 2^t to both sides to get

$$2^t - \beta_t t 2^{t-1} \geq 2^t - \frac{t}{2} - \frac{t^2}{2^t}.$$

Now it is easily verified that $\beta_t \leq 6/t$ for $t \geq 2$, so we can apply Lemma 9 with $s = t$ to get $2^t - \beta_t t 2^{t-1} \leq (2 - \beta)^t$. It follows that

$$(2 - \beta)^t \geq 2^t - \frac{t}{2} - \frac{t^2}{2^t},$$

and so

$$\beta_t \geq (2 - \beta_t)^{-t}.$$

It follows that

$$2 - \beta_t \leq 2 - (2 - \beta_t)^{-t}.$$

Since $\alpha_t = 2 - \beta_t$, we get

$$\alpha_t \leq 2 - \alpha_t^{-t},$$

as desired. □

We can now apply this to get a bound on $G_n^{(t)}$.

Corollary 11. *Let $t \geq 2$ be an integer and $n \geq 0$. Then $G_n^{(t)} \geq \alpha_t^n$, where $\alpha_t = 2 - \frac{1}{2^{t-\frac{t}{2}} - \frac{t^2}{2^t}} < 2$.*

Proof. By induction on n . Clearly $G_n^{(t)} = 2^n \geq \alpha_t^n$ for $0 \leq n < t$ by definition. Otherwise we have

$$\begin{aligned} G_n^{(t)} &= G_{n-1}^{(t)} + \cdots + G_{n-t}^{(t)} \\ &\geq \alpha_t^{n-1} + \cdots + \alpha_t^{n-t} \\ &= \frac{\alpha_t^n - \alpha_t^{n-t}}{\alpha_t - 1}. \end{aligned}$$

However, $\alpha_t \leq 2 - \alpha_t^{-t}$ by Theorem 10, so

$$\alpha_t - 1 \leq 1 - \alpha_t^{-t}.$$

Hence $(\alpha_t - 1)\alpha_t^n \leq (1 - \alpha_t^{-t})\alpha_t^n = \alpha_t^n - \alpha_t^{n-t}$, so from above we have

$$G_n^{(t)} \geq \frac{\alpha_t^n - \alpha_t^{n-t}}{\alpha_t - 1} \geq \alpha_t^n.$$

□

Now we state and prove our lower bound on the number of binary privileged words of length n .

Theorem 12. *There are at least*

$$\frac{2^{n-5}}{n^2}$$

privileged binary words of length n .

Proof. Each word of the form $0^t 1 w 10^t$ is privileged, where $|w| = n - 2t - 2$ and w contains no factor 0^t . The number of such w , as we have seen, is $G_{n-2t-2}^{(t)}$. So it suffices to pick the right t to get a lower bound on $G_{n-2t-2}^{(t)}$.

It is easy to check, using the data in the next section, that our bound holds for $n \leq 10$. So assume $n \geq 11$.

Now

$$\begin{aligned} G_{n-2t-2}^{(t)} &\geq \alpha_t^{n-2t-2} \\ &= (2 - \beta_t)^{n-2t-2} \\ &\geq 2^{n-2t-2} - \beta_t(n-2t-2)2^{n-2t-3} \\ &= 2^{n-2t-2}(1 - \beta_t(n/2 - t - 1)), \end{aligned}$$

by Lemma 9 with $s = n - 2t - 2$, provided $\beta_t \leq 6/(n - 2t - 2)$.

We now choose $t = \lfloor \log_2 n \rfloor + 1$, so that

$$2^{t-1} \leq n < 2^t. \quad (2)$$

It is now easy to verify that $\beta_t \leq 6/(n - 2t - 2)$ for $n \geq 11$.

On the other hand, it is easy to verify that

$$\frac{3t}{4} \geq \frac{t^2}{2^{t+1}}$$

for all real $t \geq 0$, so

$$\frac{3t}{4} + 1 - \frac{t^2}{2^{t+1}} > 0.$$

Adding 2^{t-1} to both sides, and using (2), we get

$$\frac{n}{2} < 2^{t-1} < 2^{t-1} + \frac{3t}{4} + 1 - \frac{t^2}{2^{t+1}},$$

which implies

$$\frac{n}{2} - t - 1 \leq \frac{1}{2} \left(2^t - \frac{t}{2} - \frac{t^2}{2^t} \right)$$

and so $\beta_t(n/2 - t - 1) \leq 1/2$.

It follows that

$$B(n) \geq G_{n-2t-2}^{(t)} \geq 2^{n-2t-2} (1 - \beta_t(n/2 - t - 1)) \geq 2^{n-2t-3} \geq \frac{2^{n-5}}{n^2}.$$

□

Open Problem 13. What is the true asymptotic behavior of $B(n)$ as $n \rightarrow \infty$?

Define the function f as follows:

$$f(n) = \begin{cases} n, & \text{if } n \geq 2; \\ nf(\lfloor \log_2 n \rfloor), & \text{otherwise.} \end{cases}$$

It should be possible to improve Theorem 12 to $B(n) = \Omega(2^n c^{\log^*(n)} / f(n))$, where c is a constant and, as usual, $\log^*(n)$ is the number of times we need to apply \log_2 to n to get a number ≤ 1 . We sketch the outline of an incomplete argument here:

We generalize our argument above to count the number of privileged words of length n having any privileged border of length $\lfloor \log_2 n \rfloor$. We can use our previous argument provided the count for arbitrary patterns is larger than the count for 0^t .

More precisely, if $x(p, n)$ is the number of strings of length n beginning with the pattern p , ending with p , and having no other occurrence of p , then $x(p, n)$ satisfies a linear recurrence of order $t = |p|$. By analyzing this carefully, it should be possible to show that, provided n is in a certain range with respect to $|p|$, we have $x(p, n) \geq x(0^t, n)$.

Then we can imitate our analysis above, setting $t = \lfloor \log_2 n \rfloor$, to get

$$\begin{aligned} B(n) &\geq \sum_{\substack{p \text{ privileged} \\ |p| = \lfloor \log_2 n \rfloor}} x(p, n) \\ &\geq cB(\lfloor \log_2 n \rfloor) \cdot \frac{2^n}{n^2}, \end{aligned}$$

for a constant c . By iterating this relationship $\log^*(n)$ times, we would get the claimed bound.

5 Explicit enumeration of privileged words

We finish with a table giving the number $B(n)$ of privileged binary words of length n for $0 \leq n \leq 38$. It is sequence A231208 in Sloane's *On-line Encyclopedia of Integer Sequences* [6].

n	$B(n)$	n	$B(n)$	n	$B(n)$
0	1	13	328	26	875408
1	2	14	568	27	1649236
2	2	15	1040	28	3112220
3	4	16	1848	29	5888548
4	4	17	3388	30	11160548
5	8	18	38576	31	21198388
6	8	19	71444	32	40329428
7	16	20	133256	33	76865388
8	20	21	248676	34	146720792
9	40	22	466264	35	280498456
10	60	23	875408	36	536986772
11	108	24	1649236	37	1029413396
12	176	25	3112220	38	1975848400

References

- [1] D. E. Knuth. *The Art of Computer Programming*. Volume 3: Sorting and Searching. Addison-Wesley, 1973.
- [2] D. E. Knuth, J. H. Morris, and V. Pratt. Fast pattern matching in strings. *SIAM J. Comput.* **6** (1977), 323–350.
- [3] W. Ogden. A helpful result for proving inherent ambiguity. *Math. Systems Theory* **2** (1968), 191–194.

- [4] J. Peltomäki. Introducing privileged words: privileged complexity of Sturmian words. *Theoret. Comput. Sci.* **500** (2013), 57–67.
- [5] J. Peltomäki. Privileged factors in the Thue-Morse word — a comparison of privileged words and palindromes. Preprint, June 28 2013, <http://arxiv.org/abs/1306.6768>.
- [6] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. Available at <http://oeis.org>.