More on prefix normal words

Péter Burcsi, Gabriele Fici, Zsuzsanna Lipták,
Frank Ruskey, and Joe Sawada

LSD/LAW 2015
London, 5-6 Feb. 2015

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more **1**s than the prefix of the same length.

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more $\mathbf{1}$s than the prefix of the same length.

Example

- $\mathbf{11100000}$ p.n.?

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more $\mathbf{1}$s than the prefix of the same length.

Example

- $\mathbf{11100000}$ p.n.? YES

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more $\mathbf{1}$s than the prefix of the same length.

Example

- $\mathbf{11100000}$ p.n.? YES
- $\mathbf{11011100}$ p.n.?

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more $\mathbf{1}$s than the prefix of the same length.

Example

- $\mathbf{11100000}$ p.n.? YES
- $\mathbf{11011100}$ p.n.? NO 111 has more 1s than 110

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word *w* is prefix normal (w.r.t. 1) if no substring has more **1**s than the prefix of the same length.

Example

- **11100000** p.n.? YES
- **11011100** p.n.? NO 111 has more 1s than 110
- **11001101** p.n.?

Prefix normal words

Fici, L. (DLT 2011)

Definition

A binary word w is prefix normal (w.r.t. 1) if no substring has more **1**s than the prefix of the same length.

Example

- **11100000** p.n.? YES
- **11011100** p.n.? NO 111 has more 1s than 110
- **11001101** p.n.? NO 1101 has more 1s than 1100

Prefix normal words

So how can we tell whether w is prefix normal?

Let $F_1(w, k)$ = max # 1s in all k-length substrings

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4

Definition (again)

w prefix normal $\Leftrightarrow$ $\forall k$: $w_1 \ldots w_k$ has $F_1(w, k)$ many 1s

Prefix normal forms

Definition

Two words v, w are prefix normal equivalent if $F_1(v, \cdot) = F_1(w, \cdot)$.

Example

1100, 0110, 0011 are p.n. equivalent.

Prefix normal forms

Definition

Two words v, w are prefix normal equivalent if $F_1(v, \cdot) = F_1(w, \cdot)$.

Example

1100, 0110, 0011 are p.n. equivalent.

Lemma

Each equivalence class contains exactly one prefix normal word.

Definition

The prefix normal form $\mathrm{PNF}_1(w)$ of w is the unique p.n. word in w's equivalence class.

Prefix normal forms

Recall: $v \equiv w$ if same F_1-function
$F_1(w,k) = \max \#$ 1s in k-length substrings

PNF_1	class	cardinality
1111	{1111}	1
1110	{1110, 0111}	2
1101	{1101, 1011}	2
1100	{1100, 0110, 0011}	3
1010	{1010, 0101}	2
1001	{1001}	1
1000	{1000, 0100, 0010, 0001}	4
0000	{0000}	1

- w and w^{rev} are in the same class
- if $[w]$ has size 1 then w p.n. and palindrome

Prefix normal forms

Thus, $\mathrm{PNF}_1(w)$ is an encoding of the function $F_1(w, \cdot)$.

Prefix normal forms

Thus, $\mathrm{PNF}_1(w)$ is an encoding of the function $F_1(w, \cdot)$.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4

Prefix normal forms

Thus, $\mathrm{PNF}_1(w)$ is an encoding of the function $F_1(w, \cdot)$.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4

- Note that F_1 increases by at most 1.

Prefix normal forms

Thus, $\mathrm{PNF}_1(w)$ is an encoding of the function $F_1(w, \cdot)$.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4

- Note that F_1 increases by at most 1.
- Construct $\mathrm{PNF}_1(w)$: put a 1 if F_1 increases, 0 otherwise

Prefix normal forms

Thus, $\mathrm{PNF}_1(w)$ is an encoding of the function $F_1(w, \cdot)$.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4

- Note that F_1 increases by at most 1.
- Construct $\mathrm{PNF}_1(w)$: put a 1 if F_1 increases, 0 otherwise

$$\mathrm{PNF}_1(w) = \mathbf{1101001}$$

Where do prefix normal words come from?

Binary Jumbled Pattern Matching (BJPM)

Does $\mathbf{w} = \mathbf{1010011011000111001 0}$ have a substring of length 11 containing exactly 5 ones?

Where do prefix normal words come from?

Binary Jumbled Pattern Matching (BJPM)

Does $\mathbf{w} = \mathbf{1010011011000111001 0}$ have a substring of length 11 containing exactly 5 ones? (Online: easy, $\mathcal{O}(|w|)$. Indexed: ?)

Where do prefix normal words come from?

Binary Jumbled Pattern Matching (BJPM)

Does $\mathbf{w} = \mathbf{10100110110001110010}$ have a substring of length 11 containing exactly 5 ones? (Online: easy, $\mathcal{O}(|w|)$. Indexed: ?)

Recent papers on this problem:

PSC 2009, FUN 2010, IPL 2010, JDA 2012, ToCS 2012, IJFCS 2012, CPM 2012, SPIRE 2012, IPL 2013, IPL 2013, ESA 2013 x 2, SPIRE 2013, TCS 2014, PhTRS-A 2014, CPM 2014, CPM 2014, FUN 2014, Arxiv 2014, ISIT 2014, SPIRE 2014, ICALP 2014, ...

(red ones have an intersection with current authors)

Binary Jumbled Pattern Matching (BJPM)

Interval property

If w has a substring of length k with x 1s, and another with y 1s, then it also has one with z 1s, for every $x \leq z \leq y$.

Binary Jumbled Pattern Matching (BJPM)

Interval property

If w has a substring of length k with x 1s, and another with y 1s, then it also has one with z 1s, for every $x \leq z \leq y$.

Therefore, it is enough to store, for each k, max and min no. of 1s.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
max	0	1	2	2	3	3	3	4
min	0	0	0	1	2	2	3	4

Binary Jumbled Pattern Matching (BJPM)

Interval property

If w has a substring of length k with x 1s, and another with y 1s, then it also has one with z 1s, for every $x \leq z \leq y$.

Therefore, it is enough to store, for each k, max and min no. of 1s.

Example

$\mathbf{w} = \mathbf{1001101}$

k	0	1	2	3	4	5	6	7
max	0	1	2	2	3	3	3	4
min	0	0	0	1	2	2	3	4

This index (table) has size $O(n)$ and allows constant time queries.

BJPM: Construction of index

- $O(n^2)$ time—Cicalese, Fici, L. (PSC 2009)
- $O(n^2/\log n)$ time
  —Burcsi, Cicalese, Fici, L. (FUN 2010); Moosa, Rahman (IPL 2010)
- $O(n^2/\log^2 n)$ time in word-RAM model
  —Moosa, Rahman (JDA 2012)
- approximate index with one-sided error in $O(n^{1+\epsilon})$ time
  —Cicalese, Laber, Weimann, Yuster (CPM 2012)
- Corner Index: construction time, index size, query time depend on
  $r =$ runlength enc. of s
  —Badkobeh, Fici, Kroon, L. (IPL 2013); Giaquinta, Grabowski (IPL 2013)
- $n^2/2^{\Omega(\log n/\log\log n)^{1/2}}$—Hermelin, Landau, Rabinovich, Weimann (unpubl.)
- $O(n^{1.859})$ time—Chan & Lewenstein (unpubl.)

BJPM using prefix normal forms

The prefix normal form is just a different way of looking at this index.

Example

$\mathbf{w} = \mathbf{1001101}$, define analogously $F_0(w, k)$ and $\mathrm{PNF}_0(w)$:

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4
$F_0(w, k)$	0	1	2	2	2	3	3	3

BJPM using prefix normal forms

The prefix normal form is just a different way of looking at this index.

Example

$\mathbf{w} = \mathbf{1001101}$, define analogously $F_0(w, k)$ and $\mathrm{PNF}_0(w)$:

k	0	1	2	3	4	5	6	7
$F_1(w, k)$	0	1	2	2	3	3	3	4
$F_0(w, k)$	0	1	2	2	2	3	3	3

$\mathrm{PNF}_1(w) = \mathbf{1101001}$,

BJPM using prefix normal forms

The prefix normal form is just a different way of looking at this index.

Example

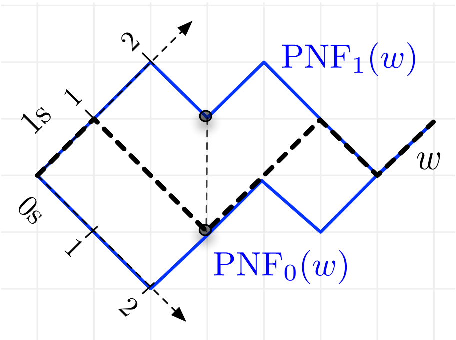
$\mathbf{w} = \mathbf{1001101}$, define analogously $F_0(w,k)$ and $\text{PNF}_0(w)$:

k	0	1	2	3	4	5	6	7
$F_1(w,k)$	0	1	2	2	3	3	3	4
$F_0(w,k)$	0	1	2	2	2	3	3	3

$\text{PNF}_1(w) = \mathbf{1101001}$, $\text{PNF}_0(w) = \mathbf{0011011}$

BJPM using prefix normal forms

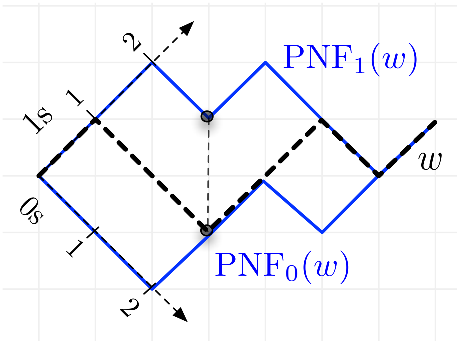
Drawing: $\diagup = 1$, $\diagdown = 0$



$\mathbf{w} = \mathbf{1001101}$, $\text{PNF}_1(w) = \mathbf{1101001}$, $\text{PNF}_0(w) = \mathbf{0011011}$

BJPM using prefix normal forms

w = **1001101**

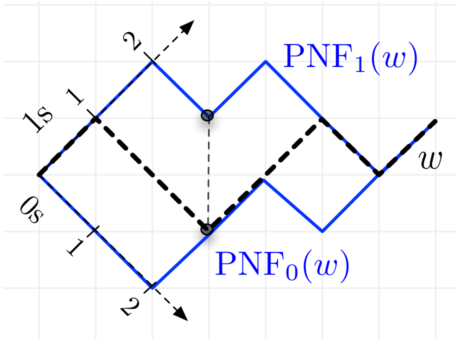


k	0	1	2	3	4	5	6	7
max	0	1	2	2	3	3	3	4
min	0	0	0	1	2	2	3	4

vertical line: $k = 3$

BJPM using prefix normal forms

w = 1001101



Thus, it suffices to query the two PNFs for answering BJPM queries (bitstring rank queries in constant time).

Some questions about prefix normal words

1. Compute prefix normal forms: Given $w \in \{0,1\}^n$, compute $\mathrm{PNF}_1(w)$ (interesting for Binary Jumbled Pattern Matching).
   – maybe faster directly?

Some questions about prefix normal words

1. Compute prefix normal forms: Given $w \in \{0,1\}^n$, compute $\mathrm{PNF}_1(w)$ (interesting for Binary Jumbled Pattern Matching).
   – maybe faster directly?
2. Testing: Given w, is w prefix normal? – naïve: $O(n^2)$, lit (on BJPM): $O(n^2/polylog\, n)$, our algos: $O(n^2)$ w-c, $O(n)$ expected conjectured

Some questions about prefix normal words

1. Compute prefix normal forms: Given $w \in \{0,1\}^n$, compute $\mathrm{PNF}_1(w)$ (interesting for Binary Jumbled Pattern Matching).
   – maybe faster directly?
2. Testing: Given w, is w prefix normal? – naïve: $O(n^2)$, lit (on BJPM): $O(n^2/polylog\, n)$, our algos: $O(n^2)$ w-c, $O(n)$ expected conjectured
3. Generation (listing): List all prefix normal words of length n.
   – our CPM 2014 paper, based on p.n. words being a bubble language presented at LSD/LAW 2014

Some questions about prefix normal words

1. Compute prefix normal forms: Given $w \in \{0,1\}^n$, compute $\mathrm{PNF}_1(w)$ (interesting for Binary Jumbled Pattern Matching).
   – maybe faster directly?
2. Testing: Given w, is w prefix normal? – naïve: $O(n^2)$, lit (on BJPM): $O(n^2/polylog\, n)$, our algos: $O(n^2)$ w-c, $O(n)$ expected conjectured
3. Generation (listing): List all prefix normal words of length n.
   – our CPM 2014 paper, based on p.n. words being a bubble language presented at LSD/LAW 2014
4. Enumeration (counting): number of p.n. words of length n?
   – partial results in our FUN 2014 paper

Some questions about prefix normal words

1. Compute prefix normal forms: Given $w \in \{0,1\}^n$, compute $\mathrm{PNF}_1(w)$ (interesting for Binary Jumbled Pattern Matching).
   – maybe faster directly?
2. Testing: Given w, is w prefix normal? – naïve: $O(n^2)$, lit (on BJPM): $O(n^2/polylog\, n)$, our algos: $O(n^2)$ w-c, $O(n)$ expected conjectured
3. Generation (listing): List all prefix normal words of length n.
   – our CPM 2014 paper, based on p.n. words being a bubble language presented at LSD/LAW 2014
4. Enumeration (counting): number of p.n. words of length n?
   – partial results in our FUN 2014 paper
5. In general, properties of p.n. words.

Some properties of prefix normal words

Recall: w is prefix normal (w.r.t. 1) if no substring has more **1**s than the prefix of the same length.

- w is p.n. $\Rightarrow$ $w0$ is p.n.
- w is p.n. $\Rightarrow$ every prefix of w is p.n.
- for any w of length n, $v = 1^n w$ is p.n.
- ...

Enumerating p.n. words

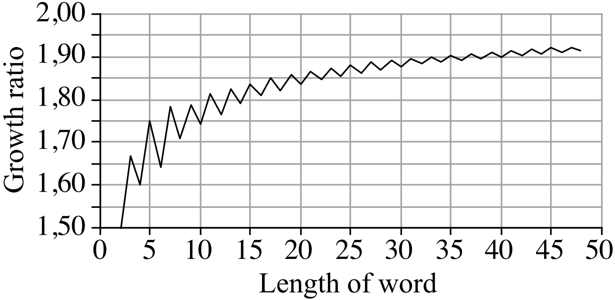
OEIS sequence no. A194850

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$pnw(n)$	2	3	5	8	14	23	41	70	125	218	395	697	1273	2279	4185	7568

Enumerating p.n. words

OEIS sequence no. A194850

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$pnw(n)$	2	3	5	8	14	23	41	70	125	218	395	697	1273	2279	4185	7568



Enumerating p.n. words

Let $pnw(n) = \#$ prefix normal words of length n.

Easy:

$pnw(n)$ grows exponentially.
Proof: $\forall w : 1^{|w|}w$ is p.n. $\Rightarrow$ $pnw(2n) \geq 2^n$.

Enumerating p.n. words

Let $pnw(n) = \#$ prefix normal words of length n.

Easy:

$pnw(n)$ grows exponentially.
Proof: $\forall w : 1^{|w|}w$ is p.n. $\quad \Rightarrow \quad pnw(2n) \geq 2^n$.

Theorem
Exists $c > 0$: $pnw(n) = \Omega(2^{n-c\sqrt{n \ln n}}) = \Omega((2-\epsilon)^n) \quad \forall \epsilon > 0$.

Theorem
$pnw(n) = O\left(\frac{2^n (\ln n)^2}{n}\right) = o(2^n)$.

Fixed-density p.n. words

For a word w, its density d is the number of 1s in w,
$pnw(n, d)$ = no. of p.n. words of length n and density d.

Fixed-density p.n. words

For a word w, its **density** d is the number of 1s in w,
$pnw(n, d)$ = no. of p.n. words of length n and density d.

Theorem

Let $f_d(x) = \sum_{n \geq 0} pnw(n, d)x^n$, the generating function of $pnw(n, d)$.

gen.func.	coefficients	closed form
$f_0(x) = \frac{1}{1-x}$	$1, 1, 1, 1, 1, 1, 1, 1 \ldots$	1
$f_1(x) = \frac{x}{1-x}$	$0, 1, 1, 1, 1, 1, 1, 1 \ldots$	$[n > 0]$
$f_2(x) = \frac{x^2}{(1-x)^2}$	$0, 0, 1, 2, 3, 4, 5, 6 \ldots$	$(n-1)$
$f_3(x) = \frac{x^3}{(1-x^2)(1-x)^2}$	$0, 0, 0, 1, 2, 4, 6, 9, \ldots$	$pnw(2n) = n(n-1)$, $pnw(2n+1) = (n-1)^2$

. . . and more (but for larger d becomes less and less manageable) . . .

Fixed-density p.n. words

For the proof, we use the following encoding of a binary word w: Set $r_j =$ distance between consecutive 1s, and r_d s.t. $\sum r_j = n$.

Example

$\mathbf{w} = \mathbf{110100010}$, then $d = 4$, $r_1 = 1$, $r_2 = 2$, $r_3 = 4$ and $r_4 = 2$.

Fixed-density p.n. words

For the proof, we use the following encoding of a binary word w: Set $r_j =$ distance between consecutive 1s, and r_d s.t. $\sum r_j = n$.

Example

$\mathbf{w} = \mathbf{110100010}$, then $d = 4$, $r_1 = 1$, $r_2 = 2$, $r_3 = 4$ and $r_4 = 2$.

Lemma

A word w is prefix normal if and only if the following inequalities hold:

$$\begin{array}{rcll} r_1 & \leq & r_j & j = 2, 3, \ldots, d-1 \\ r_1 + r_2 & \leq & r_j + r_{j+1} & j = 2, 3, \ldots, d-2 \\ r_1 + r_2 + r_3 & \leq & r_j + r_{j+1} + r_{j+2} & j = 2, 3, \ldots, d-3 \\ & \vdots & & \vdots \\ r_1 + r_2 + \cdots + r_{d-2} & \leq & r_j + r_{j+1} + \cdots + r_{d-1} & j = 2 \end{array}$$

Fixed-density p.n. words

Corollary

pnw(n) *grows exponentially.*

Proof.

If $r_1 \leq r_2 \leq \ldots \leq r_d$ holds, then the Lemma is satisfied. Therefore,

$$pnw(n) \geq \text{ number of integer partitions of } n.$$

□

(But we already knew this.)

Words with fixed prefix

For a word w, let $ext(w, m)$ denote the number of prefix normal words of length $|w| + m$ with prefix w.

w	$ext(w, 4)$	$ext(w, \|w\|)$
1111	16	2^n
1110	15	$2^n - 1$
1101	11	$2^n - 5$
1100	11	$2^n - (n + 1)$
1010	8	$F(2\lfloor \frac{n}{2} \rfloor + 2)$
1001	3	3
1000	5	$n + 1$
0000	1	1

$F(n) = n$'th Fibonacci number

Words with fixed prefix

In general ($ext(w,m)$: # p.n. words of length $|w|+m$ with prefix w):

Lemma

$$
\begin{array}{lcl}
ext(1^n, n) & = & 2^n \\
ext(1^{n-1}0, n) & = & 2^n - 1 \\
ext(1^{n-2}01, n) & = & 2^n - 5 \\
ext(1^{n-2}00, n) & = & 2^n - (n+1) \\
ext((10)^{\frac{n}{2}}, n) & = & F(n+2) \textit{ if } n \textit{ even.} \\
ext((10)^{\lfloor \frac{n}{2} \rfloor}1, n) & = & F(n+1) \textit{ if } n \textit{ odd.} \\
ext(10^{n-2}1, n) & = & 3 \\
ext(10^{n-1}, n) & = & n+1 \\
ext(0^n, n) & = & 1
\end{array}
$$

$F(n) = n$'th Fibonacci number

Critical prefix length

For a word $w = 1^s 0^t v$, the critical prefix length is $cr(w) = s + t$, the length of the first 1-run plus the first 0-run (in this order! note that $s = 0$ is possible).

Lemma

$\mathbb{E}(cr(w)) = 3$ *for a random binary word. (For fixed-length, tends to* 3 *for* $n \to \infty$*.)*

Lemma

$\mathbb{E}(cr(PNF_1(w)) = \Theta(\log n)$ *for a random binary word* w *of length* n*.*

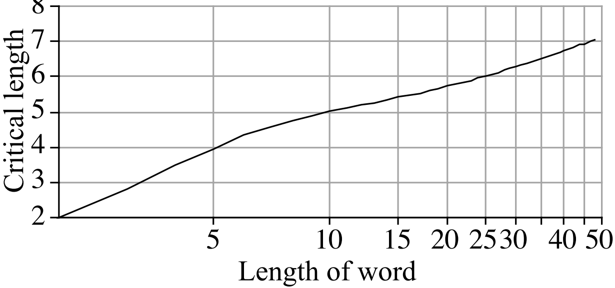
Conjecture

$\mathbb{E}(cr(w)) = \Theta(\log n)$ for a random p.n. word of length n.

Critical prefix length

Conjecture

$\mathbb{E}(cr(w)) = \Theta(\log n)$ for a random p.n. word of length n.



We need this conjecture for our generating algorithm! If conjecture true, then it runs in amortized $O(\log n)$ time.

References

If you got interested . . .

- Fici, L. (DLT 2011)
- Burcsi, Fici, L., Ruskey, Sawada (CPM 2014): combinatorial generation of all p.n. words, based on them being a bubble language.
- Burcsi, Fici, L., Ruskey, Sawada (FUN 2014): partial results on enumeration of p.n. words
- OEIS seq. no. A194850: no. of p.n. words of length n
- OEIS seq. no. A238109 (added by Sloane): prefix normal words over $\{1, 2\}$ in lex. order

THANK YOU!