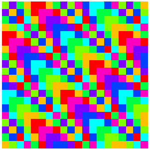
Computing k-binomial equivalence & avoiding binomial repetitions

Michel Rigo

http://www.discmath.ulg.ac.be/
http://hdl.handle.net/2268/187305
28th October 2015



The notion of binomial coefficient of words is classical in COW. See, for instance, Sakarovitch & Simon, Lothaire.

$\binom{w}{x}$ number of times x appears as a (scattered) subword of w

i.e., x occurs as a subsequence of w

We count the number of increasing maps $\varphi : \{1, \ldots, |x|\} \to \{1, \ldots, |w|\}$ such that

$$\varphi(1) < \cdots < \varphi(|x|)$$

$$w_{\varphi(1)} \cdots w_{\varphi(|x|)} = x$$

$$\binom{aabbab}{ab} = 7$$

It generalizes the usual binomial coefficients for integers

$$\binom{a^m}{a^n} = \binom{m}{n}, \quad m, n \in \mathbb{N}$$

Observe that $\displaystyle\binom{w}{a} = |w|_a, \quad a \in A$

We can easily compute coefficients:

$$\binom{w}{\varepsilon} = 1, \qquad \binom{w}{x} = 0, \quad \text{if } |w| < |x|$$

$$u, v \in A^*, a, b \in A, \quad \binom{ua}{vb} = \binom{u}{vb} + \delta_{a,b}\binom{u}{v}$$

```
coeff[u_, v_] := coeff[u, v] =
   If[Length[v] == 0, 1,
    If[Length[u] < Length[v], 0,
     coeff[Drop[u, -1], v]
      + ((Last[u] == Last[v]) /. {True -> 1, False -> 0})
         coeff[Drop[u, -1], Drop[v, -1]]
      ]
     ]
```

Definition

Let $k \geq 1$. Two words u, v are *k-binomially equivalent*

$$u \equiv_k v$$

if and only if

$$\binom{u}{x} = \binom{v}{x} \quad \forall x \in A^{\leq k}.$$

Remark: 1-binomial equivalence $=$ *abelian equivalence*.

One also finds the notion of *k-spectrum* of a word u which is the (formal) polynomial in $\mathbb{N}\langle A^* \rangle$ of degree k

$$\mathsf{Spec}_{u,k} = \sum_{x \in A^{\leq k}} \binom{u}{x}\, x.$$

Two words are k-binomially equivalent iff they have the same k-spectrum. $\rightsquigarrow$ full information.

Example

The 2-spectrum of the word $u = abbab$ is

$$\mathsf{Spec}_{u,2} = 1\varepsilon + 2a + 3b + aa + 4ab + 2ba + 3bb.$$

The 3-spectrum of this word is

$$\mathsf{Spec}_{u,3} = \mathsf{Spec}_{u,2} + aab + 2aba + 3abb + 2bab + bba + bbb.$$

Note that the k-spectrum contains

$$\frac{(\#A)^{k+1} - 1}{(\#A) - 1} \text{ (possibly zero) coefficients.}$$

$\rightsquigarrow$ grows exponentially with k.

$$2 + 3 = \binom{5}{1},\ 1 + 4 + 2 + 3 = \binom{5}{2},\ 1 + 2 + 3 + 2 + 1 + 1 = \binom{5}{3}$$

In COW, there is a zoo of equivalence relations :

- abelian equivalence (since Erdős in 1961)

$$abbacba \sim_{ab} cababba$$

- k-abelian equivalence (Karhumäki *et al.*)

$$|u|_x = |v|_x \quad \forall x \in A^{\leq k}$$

- k-binomial equivalence
- (Parikh) matrix equivalence (Salomaa *et al.* 2000)
- Simon's congruence (1975, Karandikar *et al.* 2015)

$$Supp(\mathsf{Spec}_{u,k}) = Supp(\mathsf{Spec}_{v,k})$$

applications to piecewise testable languages

Link with Parikh matrices.

$A = \{a_1, \ldots, a_k\}$. The *Parikh matrix mapping*

$$\psi_k : A^* \to \mathbb{N}^{(k+1)\times(k+1)}$$

is the morphism defined by the condition:
if $\psi_k(a_q) = (m_{i,j})_{1\le i,j\le k+1}$, then for each $i \in \{1, \ldots, k+1\}$,

$$m_{i,i} = 1, \quad m_{q,q+1} = 1,$$

all other elements of the matrix $\psi_k(a_q)$ being 0.

DEFINITION

Two words are M-*equivalent*, or *matrix equivalent*,
if they have the same Parikh matrix.

Example, $\#A = 2$

Consider $A = \{a, b\}$. We have

$$\psi_2(a) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \psi_2(b) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$\psi_2(abbab) = \psi_2(a)\psi_2(b)\psi_2(b)\psi_2(a)\psi_2(b) = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Parikh matrices for an alphabet of cardinality k encode

$$k(k+1)/2$$

of the binomial coefficients of a word w for subwords of length $\leq k$.

Theorem (A. Mateescu, A. Salomaa, K. Salomaa, S. Yu 2001)

Let $A = \{a_1, \ldots, a_k\}$ be an (ordered) alphabet.
Let w be a finite word and $\psi_k(w) = (m_{i,j})_{1 \leq i,j \leq k+1}$.
Then

$$m_{i,j+1} = \binom{w}{a_i \cdots a_j}$$

for all $1 \leq i \leq j \leq k$.

$\rightsquigarrow$ partial information : $\mathcal{O}(k^2)$ vs. $\Omega((\#A)^k)$

Example over $A = \{a, b, c\}$

$$\psi_3(w) = \begin{pmatrix} 1 & \binom{w}{a} & \binom{w}{ab} & \binom{w}{abc} \\ 0 & 1 & \binom{w}{b} & \binom{w}{bc} \\ 0 & 0 & 1 & \binom{w}{c} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\psi_3(wb) = \begin{pmatrix} 1 & \binom{w}{a} & \binom{w}{ab} & \binom{w}{abc} \\ 0 & 1 & \binom{w}{b} & \binom{w}{bc} \\ 0 & 0 & 1 & \binom{w}{c} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For instance,

$$\binom{wb}{ab} = \binom{w}{a} + \binom{w}{ab}$$

Also **generalized** *Parikh mappings* ψ_u, for all words $u \in A^*$, can be defined.

Let $u = u_1 \cdots u_\ell$.
If $\psi_u(a) = (m_{i,j})_{1 \leq i,j \leq \ell+1}$, then for each $i \in \{1, \ldots, \ell+1\}$, $m_{i,i} = 1$, and for each $i \in \{1, \ldots, \ell\}$,

$$m_{i,i+1} = \delta_{a,u_i},$$

all other elements of the matrix $\psi_u(a)$ being 0.

Remark

We get back to the 'classical' Parikh matrices with

$$u = a_1 a_2 \cdots a_k$$

if $A = \{a_1, \ldots, a_k\}$.

We have

$$\psi_{abba}(a) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \ \psi_{abba}(b) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Natural generalization of the theorem of Mateescu et al.

Theorem (Şerbănuţă 2004)

Let $u = u_1 \cdots u_\ell$ and w a word. Let $\psi_u(w) = (m_{i,j})_{1\le i,j\le \ell+1}$. Then, for all $1 \le i \le j \le \ell$,

$$m_{i,j+1} = \binom{w}{u_i \cdots u_j}.$$

In particular, the first row of $\psi_u(w)$ contains the coefficients corresponding to the prefixes of w:

$\binom{w}{\varepsilon}, \binom{w}{u_1}, \binom{w}{u_1 u_2}, \ldots, \binom{w}{u_1 \cdots u_{\ell-1}}, \binom{w}{u}.$

Similarly, the last column of $\psi_u(w)$ contains the coefficients corresponding to the suffixes:

$\binom{w}{u}, \binom{w}{u_2 \cdots u_\ell}, \ldots, \binom{w}{u_1}, \binom{w}{\varepsilon}.$

Example

$$\psi_{abba}(w) = \begin{pmatrix} 1 & \binom{w}{a} & \binom{w}{ab} & \binom{w}{abb} & \binom{w}{abba} \\ 0 & 1 & \binom{w}{b} & \binom{w}{bb} & \binom{w}{bba} \\ 0 & 0 & 1 & \binom{w}{b} & \binom{w}{ba} \\ 0 & 0 & 0 & 1 & \binom{w}{a} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Link between k-binomial equivalence and matrix equivalence

PROPOSITION

Over a 2-letter alphabet, two words are 2-binomially equivalent if and only if they have the same Parikh matrix.

$$\psi_2(w) = \begin{pmatrix} 1 & \binom{w}{a} & \binom{w}{ab} \\ 0 & 1 & \binom{w}{b} \\ 0 & 0 & 1 \end{pmatrix}$$

$\Rightarrow$ clear !

$\Leftarrow$

$$\binom{w}{aa} = \binom{|w|_a}{2}$$

$$\binom{w}{aa} + \binom{w}{ab} + \binom{w}{ba} + \binom{w}{bb} = \binom{|w|}{2}$$

Unfortunately, we do not have more.

Two words over $\{a, b, c\}$,

$$u = abcbabcbabcbab \text{ and } v = bacabbcabbcbba$$

- not 3-binomially equivalent: $\binom{u}{abb} = 34$ and $\binom{v}{abb} = 36$,
- BUT with the same Parikh matrix $\psi_3(u) = \psi_3(v)$.

Note: they do not have the same *generalized* Parikh matrix

$$\psi_{abb}(u) \neq \psi_{abb}(v).$$

Erasing the c's, we get two words over $\{a, b\}$

$$u' = abbabbabbab \text{ and } v' = baabbabbbba$$

- not 3-binomially equivalent : $\binom{u'}{abb} = 34$, $\binom{v'}{abb} = 36$
- BUT with the same Parikh matrix
$$\begin{pmatrix} 1 & 4 & 16 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{pmatrix}$$

Indeed, 3-binomial equivalence is a strict refinement of 2-binomial equivalence.

Finally, two words over $\{a, b, c\}$

$$u = bccaa \text{ and } v = cacab$$

- not 2-binomially equivalent: $\binom{u}{ca} = 4$ and $\binom{v}{ca} = 3$,
- BUT with the same Parikh matrix $\psi_3(u) = \psi_3(v)$.

$$\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Theorem (A. Salomaa 2010)

Over a 2-letter alphabet A, two words have the same Parikh matrix if and only if one can be obtain from the other by a finite sequence of transformations of the form

$$x\,ab\,y\,ba\,z \to x\,ba\,y\,ab\,z$$

where $a, b \in A$ and $x, y, z \in A^*$.

Recall, it also works for 2-binomial equivalence.

$$1011001001011 \equiv_2 1101001000111 \equiv_2 1100110000111$$

$$\#[0\cdots01\cdots1]_{\equiv_2} = 1$$

$$\#(\{a, b\}^n/{\equiv_2}) = \frac{n^3 + 5n + 6}{6}$$

Remark

If $x \equiv_{k-1} y$, then

$$p\,x\,q\,y\,r \equiv_k p\,y\,q\,x\,r$$

But it is not clear that the previous result can be generalized.

Over a 3-letter alphabet:

$$2100221 \equiv_2 0221102$$

but 2100221 cannot be factorized into $pxqyr$ with $x \equiv_{ab} y$.

Questions

Avoidance is a classical topic in COW (back to Thue early 1900).

- $\#A = 2$, any word of length ≥ 4 contains a square uu
- $\#A = 2$, cubes (even overlaps) can be avoided

$$abbabaabbaababbabaababbaabbabaab\cdots$$

- $\#A = 3$, squares can be avoided

$$(abb)(ab)(a)(abb)(a)(ab)(abb)(ab)(a)(ab)(abb)(a)(abb)(ab)\cdots$$

$$0 \mapsto 012,\ 1 \mapsto 02,\ 2 \mapsto 1$$

- $\#A = 3$, abelian squares are unavoidable
- $\#A = 4$, abelian squares can be avoided (V. Keränen)
- $\#A = 3$, abelian cubes can be avoided (F. M. Dekking)

QUESTIONS

We can define a 2-binomial square uv where $u \equiv_2 v$

"abelian square $\prec$ 2-binomial square $\prec \cdots \prec$ square"

- squares are avoidable over a 3-letter alphabet
- abelian squares are avoidable over a 4-letter alphabet

$\rightsquigarrow$ are 2-binomial squares avoidable over a 3-letter alphabet?

$$0 \mapsto 012, \;\; 1 \mapsto 02, \;\; 2 \mapsto 1$$

Remark: k-binomial squares avoidable over a 3-letter alphabet, $\forall k \geq 2$.

Questions

We can define a 2-binomial square uv where $u \equiv_2 v$

"abelian square $\prec$ 2-binomial square $\prec \cdots \prec$ square"

- squares are avoidable over a 3-letter alphabet
- abelian squares are avoidable over a 4-letter alphabet

$\rightsquigarrow$ are 2-binomial squares avoidable over a 3-letter alphabet?

$$0 \mapsto 012,\ \ 1 \mapsto 02,\ \ 2 \mapsto 1$$

Remark: k-binomial squares avoidable over a 3-letter alphabet, $\forall k \geq 2$.

Questions

We can define a 2-binomial cube uvw where $u \equiv_2 v$, $v \equiv_2 w$

$$abbabaabbaab$$

"abelian cube $\prec$ 2-binomial cube $\prec \cdots \prec$ cube"

- cubes are avoidable over a 2-letter alphabet
- abelian cubes are avoidable over a 3-letter alphabet

$\rightsquigarrow$ are 2-binomial cubes avoidable over a 2-letter alphabet?

$$0 \mapsto 001, \;\; 1 \mapsto 011$$

M. Rao, M. Rigo, P. Salimov, Avoiding 2-binomial squares and cubes, *Theoret. Comput. Sci.* **572** (2015), 83-–91.

Questions

We can define a 2-binomial cube uvw where $u \equiv_2 v$, $v \equiv_2 w$

$$abbabaabbaab$$

"abelian cube $\prec$ 2-binomial cube $\prec \cdots \prec$ cube"

- cubes are avoidable over a 2-letter alphabet
- abelian cubes are avoidable over a 3-letter alphabet

$\rightsquigarrow$ are 2-binomial cubes avoidable over a 2-letter alphabet?

$$0 \mapsto 001, \;\; 1 \mapsto 011$$

M. Rao, M. Rigo, P. Salimov, Avoiding 2-binomial squares and cubes, *Theoret. Comput. Sci.* **572** (2015), 83-–91.

Questions

Sakarovitch and Simon already asked how to better characterize or evaluate $\#(A^n/\sim_k)$ where $\sim_k$ is the Simon congruence.

- Given $k \geq 1$ and two words u, v of length n

  decide, in polynomial time w.r.t. n, k, *whether or not* $u \equiv_k v$.
- Given $k \geq 1$ and two words w, x

  find, in polynomial time, all occurrences of factors of w *which are* k*-binomially equivalent to* x.
- Given two u, v of length n,

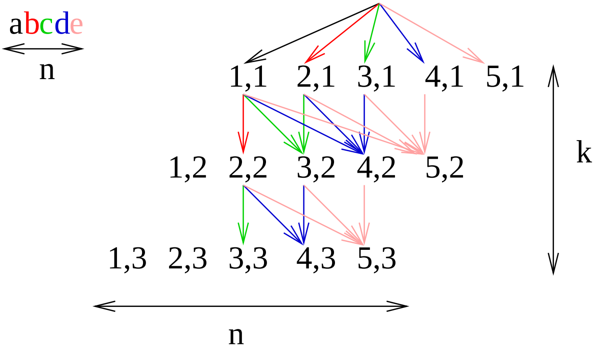
  find the largest k *such that* $u \equiv_k v$.

Also, see *k-abelian pattern matching*, T. Ehlers, F. Manea, R. Mercas, D. Nowotka, DLT 2014. (in linear time)

Main ideas of the paper
'Testing k-binomial equivalence'
`arXiv:1509.00622`
D. Freydenberger *et al.*

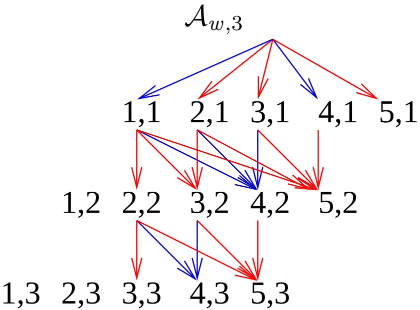
We consider the first question.

First answer, given a word w of length n and an integer k
$\rightsquigarrow$ build a NFA $\mathcal{A}_{w,k}$ with $nk+1$ states



- All states are final,
- accepts exactly the subwords of w of length $\leq k$
- a subword x is accepted $\binom{w}{x}$ times !

$w = abbab,\ k = 3$



$$\binom{w}{abb} = 3, \quad \binom{w}{ba} = 2$$

Two automata are **equivalent** if
they accept the *same language* with *the same multiplicities*.

Given two words u, v

- build $\mathcal{A}_{u,k}$ and $\mathcal{A}_{v,k}$
- $u \equiv_k v$ reduces to 'are $\mathcal{A}_{u,k}$ and $\mathcal{A}_{v,k}$ equivalent ?'

W. Tzeng, SIAM J. Computing 1992

$\rightsquigarrow$ polynomial algorithm, at least in n^3...

From Tzeng's paper abstract:

Two probabilistic automata are equivalent if for any string x, the two automata accept x with equal probability. This paper presents an $\mathcal{O}((n_1 + n_2)^4)$ algorithm for determining whether two probabilistic automata U_1 and U_2 are equivalent, where n_1 and n_2 are the number of states in U_1 and U_2, respectively.

• S. Kiefer, A. S. Murawski, et al. *On the complexity of the equivalence problem for probabilistic automata*, LNCS **7213** (2012), 467–481.

• M.-P. Schützenberger, *On the definition of a family of automata*, Inf. and Control, 245–270, 1961. (about the minimization of weighted automata)

Second answer, a randomized algorithm

Definition

Given a word $w \in \{0,1\}^*$ of length n and an integer k,

$$Q_{w,k}(X) := \sum_{v \in A^{\leq k}} \binom{w}{v} X^{val_2(1v)}$$

$$Q_{0010,2}(X) = X + 3X^2 + X^3 + 3X^4 + X^5 + X^6$$

Similar to the k-spectrum, it contains full information.

Example

The 2-spectrum of the word $abbab$ is

$$1\underbrace{\varepsilon}_{1}+2\underbrace{a}_{10}+3\underbrace{b}_{11}+\underbrace{aa}_{100}+4\underbrace{ab}_{101}+2\underbrace{ba}_{110}+3\underbrace{bb}_{111}.$$

$$Q_{01101,2}(X) = X + 2X^2 + 3X^3 + X^4 + 4X^5 + 2X^6 + 3X^7.$$

Remark

$Q_{w,k}$ is of degree

$$val(1\underbrace{1\cdots 1}_{k \text{ times}}) = 2^{k+1} - 1$$

$\rightsquigarrow$ grows exponentially with k.

Remark

Two words u, v are k-binomially equivalent if and only if

$$Q_{u,k}(X) = Q_{v,k}(X).$$

At first glance, we need to compute all the coefficients !

Let p be a (well-chosen) large prime,
$Q_{u,k}(X)$ and $Q_{v,k}(X)$ can be seen as polynomials over $\mathbb{F}_p[X]$

If $u \not\equiv_k v$, then $Q_{u,k}(X) - Q_{v,k}(X)$ is a non-zero polynomial of degree d and has at most d roots. If we randomly choose $\alpha \in \mathbb{F}_p$,

$$\mathbb{P}((Q_{u,k} - Q_{v,k})(\alpha) = 0) \leq d/p.$$

If $u \equiv_k v$, then $Q_{u,k}(X) - Q_{v,k}(X) = 0$.
For all $\alpha \in \mathbb{F}_p$, $Q_{u,k} - Q_{v,k}(\alpha) = 0$

Remark

Two words u, v are k-binomially equivalent if and only if

$$Q_{u,k}(X) = Q_{v,k}(X).$$

At first glance, we need to compute all the coefficients !

Let p be a (well-chosen) large prime,
$Q_{u,k}(X)$ and $Q_{v,k}(X)$ can be seen as polynomials over $\mathbb{F}_p[X]$

If $u \not\equiv_k v$, then $Q_{u,k}(X) - Q_{v,k}(X)$ is a non-zero polynomial of degree d and has at most d roots. If we randomly choose $\alpha \in \mathbb{F}_p$,

$$\mathbb{P}((Q_{u,k} - Q_{v,k})(\alpha) = 0) \leq d/p.$$

If $u \equiv_k v$, then $Q_{u,k}(X) - Q_{v,k}(X) = 0$.
For all $\alpha \in \mathbb{F}_p$, $Q_{u,k} - Q_{v,k}(\alpha) = 0$

A Monte–Carlo algorithm

Assume that d/p is '*small*', then randomly pick $\alpha \in \mathbb{F}_p[X]$.
Assume that we can '*easily*' compute $Q_{u,k}(\alpha)$ and $Q_{v,k}(\alpha)$.

- If $Q_{u,k}(\alpha) \neq Q_{v,k}(\alpha)$, then $u \not\equiv_k v$.
  $\rightsquigarrow$ The algorithm returns $u \not\equiv_k v$.
- If $Q_{u,k}(\alpha) = Q_{v,k}(\alpha)$, then *almost surely* $u \equiv_k v$.
  $\rightsquigarrow$ The algorithm returns $u \equiv_k v$.

We have $Q_{u,k}(\alpha) = Q_{v,k}(\alpha)$ and $u \not\equiv_k v$, only when we have picked a root of the non-zero polynomial $(Q_{u,k} - Q_{v,k})(X)$.

$\rightsquigarrow$ We could have a wrong conclusion $u \equiv_k v$ when $u \not\equiv_k v$, with probability at most d/p.

Choice of p ?

The coefficients in $Q_{w,k} \in \mathbb{F}_p[X]$ are less than n^k, indeed

$$\binom{a^n}{a^k} = \binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!} < n^k$$

$$\boxed{\text{Take a prime } p \in [n^k, 2n^k]}$$

This is not an issue for polynomial running time:

- AKS polynomial in $\log(n)$
- probabilistic test of Miller–Rabin,
  deterministic if Riemann hypothesis holds.

$Q_{w,k}(X)$ is of degree $2^{k+1}-1$ and $p \geq n^k$

$$\text{probability of error : } \frac{d}{p} \leq \frac{2^{k+1}-1}{n^k} \overset{n \to +\infty}{\longrightarrow} 0$$

For long enough words u, v, we are fairly sure of the result of the algorithm when it returns '$u \equiv_k v$'.

Main result for this algorithm

Let w be a word of length n. Let $\alpha \in \mathbb{F}_p$.
The value $Q_{w,k}(\alpha)$ can be computed in $\mathcal{O}(k^2 n)$ time.

$$Q_{w,k}(X) = \sum_{|v| \leq k} \binom{w}{v} X^{val_2(1v)} = \sum_{\ell=1}^{k} X^{2^\ell} \left(\underbrace{\sum_{|v|=\ell} \binom{w}{v} X^{val_2(v)}}_{=: R_{w,\ell}(X)} \right)$$

$\rightsquigarrow$ We need to determine the $R_{w,\ell}(\alpha)$ for all $\ell \in \{1, \ldots, k\}$

$w = w_1 \cdots w_n \quad w[i,n] = w_i \cdots w_n$

Use dynamic programming to compute the following $k \times n$ table and the values

$$R_{w[i,n],t}(\alpha), \quad i \in \{1,\dots,n\}, t \in \{1,\dots,k\}$$

$R_{w,k}$	$R_{w[2,n],k}$	$R_{w[3,n],k}$	$\cdots$	$\cdots$	$R_{w[n,n],k}$	0
$R_{w,k-1}$	$R_{w[2,n],k-1}$	$R_{w[3,n],k-1}$	$\cdots$	$\cdots$	$R_{w[n,n],k-1}$	0
$R_{w,k-2}$	$R_{w[2,n],k-2}$	$R_{w[3,n],k-2}$	$\cdots$	$\cdots$	$R_{w[n,n],k-2}$	0
$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$
			$R_{w[i,n],t}$	$R_{w[i+1,n],t}$		
				$R_{w[i+1,n],t-1}$		
$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$
$R_{w,1}$	$R_{w[2,n],1}$	$R_{w[3,n],1}$	$\cdots$	$\cdots$	$R_{w[n,n],1}$	0
1	1	1	$\cdots$	$\cdots$	1	1

$R_{\underbrace{w[n+1,n]}_{=\varepsilon},t} = 0$ if $t > 0$; $R_{w[i,n],0} = 1$ for all $1 \le i \le n+1$

$$R_{w[i,n],t},\ i \le n,\ t \ge 1,$$

depends only on $R_{w[i+1,n],t}$ and $R_{w[i+1,n],t-1}$

Let $i \leq n$, $t \geq 1$, we have

$$R_{w[i,n],t}(X) = R_{w[i+1,n],t}(X) + R_{w[i+1,n],t-1}(X), \text{ if } w_i = 0$$

$$R_{w[i,n],t}(X) = R_{w[i+1,n],t}(X) + X^{2^t} R_{w[i+1,n],t-1}(X), \text{ if } w_i = 1$$

Recall that

$$R_{w[i,n],t}(X) = \sum_{|v|=t} \binom{w_i \cdots w_n}{v} X^{val_2(v)}$$

$$\underbrace{R_{w[i,n],t}(X)}_{\downarrow} = R_{w[i+1,n],t}(X) + R_{w[i+1,n],t-1}(X), \text{ if } w_i = 0$$

$$\sum_{|v|=t} \binom{0w_{i+1}\cdots w_n}{v} X^{val_2(v)} \qquad v \text{ starts with } 0 \text{ or } 1$$

$$= \sum_{|u|=t-1} \binom{0w_{i+1}\cdots w_n}{0u} X^{val_2(0u)} + \sum_{|u|=t-1} \binom{0w_{i+1}\cdots w_n}{1u} X^{val_2(1u)}$$

$$= \sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{u} X^{val_2(u)} + \sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{0u} X^{val_2(0u)} + \sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{1u} X^{val_2(1u)}$$

$$= \overbrace{\sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{u} X^{val_2(u)}}^{R_{w[i+1,n],t-1}(X)} + \underbrace{\sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{0u} X^{val_2(0u)} + \sum_{|u|=t-1} \binom{w_{i+1}\cdots w_n}{1u} X^{val_2(1u)}}_{R_{w[i+1,n],t}(X)}$$

Summary

- Computing one element $R_{w[i,n],t}(\alpha)$ of the table is just *one addition* in $\mathbb{F}_p$ and $p \sim n^k$.
  It requires $\mathcal{O}(\log p) = \mathcal{O}(k \log n)$ — classical finite field arithmetic
- We have to compute $k \times n$ such elements
  $\rightsquigarrow \mathcal{O}(k^2 n \; \log n)$
- Finally, we compute

$$Q_{w,k}(\alpha) = \sum_{\ell=1}^{k} \alpha^{2^\ell} R_{w,\ell}(\alpha)$$

k products, each one needs $\mathcal{O}(\log^2 p) = \mathcal{O}(k^2 \log^2 n)$
$\rightsquigarrow \mathcal{O}(k^3 \; \log^2 n)$

References

- P. Karandikar, M. Kufleitner, Ph. Schnoebelen. On the index of Simon's congruence for piecewise testability, *Information Processing Letters* **15** (2015), 515–519.
- J. Maňuch, Characterization of a word by its subwords, in: G. Rozenberg, W. Thomas (Eds.), Developments in Language Theory, World Scientific Publ. Co., Singapore, 2000, pp. 210–219.
- A. Mateescu, A. Salomaa, K. Salomaa, Yu Sheng, A Sharpening of the Parikh Mapping, *RAIRO-Theoretical Informatics and Applications* **35** (2001), 551–564.
- M. Rigo, P. Salimov, Another generalization of abelian equivalence: Binomial complexity of infinite words, *Theoret. Comput. Sci.* **601** (2015), 47—57.

References

- J. Sakarovitch, I. Simon, Subwords, in: M. Lothaire (Ed.), Combinatorics on Words, Addison-Wesley, Reading, MA, 1983, pp. 105–142.
- A. Salomaa, Counting (scattered) subwords, *EATCS Bull.* **81** (2003) 165–179.
- A. Salomaa, Connections between subwords and certain matrix mappings, *Theoret. Comput. Sci.* **340** (2005) 188–203.
- A. Salomaa, Criteria for the matrix equivalence of words, *Theoret. Comput. Sci.* **411** (2010) 1818–1827.
- T.-F. Şerbănuţă, Extending Parikh matrices, *Theoret. Comput. Sci.* **310** (2004), 23–246.

16th Mons TCS Days — Liège

September 5th – 9th, 2016

http://www.cant.ulg.ac.be/jm2016/

4th CANT School & Conference — CIRM, Marseille

Combinatorics, Automata and Number Theory

November 28th – December 2nd, 2016

http://www.cant.ulg.ac.be/cant2016/

http://scientific-events.weebly.com/1502.html