Data Compression by Substring Enumeration: Presentation and Recent Results

Danny Dubé

Quebec City, Quebec, Canada

Chōfu, Tōkyō, Japan — November 17, 2017

Thanks to JSPS and Université Laval

Collaborators

Work done in collaboration with:

- Vincent Beaudoin,
- Hidetoshi Yokoo, and
- Mathieu Béliveau.

Plan of the Presentation

Initial CSE proposal

- CSE informally
- Basic definitions
- Main algorithm
- Tools for an efficient implementation
- Earliest experiments
- Links to previous compression techniques

Subsequent work

- CSE in linear time and space
- Universality for Markovian sources
- Inducing phase awareness using synchronization codes
- Explicit phase awareness

Work by the community

- Universality for stationary and Ergodic sources
- Improved bounds for special cores
- Analysis of CSE's redundancy
- Link to anti-dictionary compression
- Faster and lightweight implementations
- Direct handling of non-binary alphabets
- Two-dimensional CSE

Future work

CSE informally (1)

Let $\mathbf{D} = 01000001$.
Let $N = |\mathbf{D}| = 8$.

We consider $\mathbf{D}$ to be *circular*.

The substring enumeration, from the compressor's point of view:

Length	**Substrings**							
0	$8\times\epsilon$							
1	6×0						2×1	
2	4×00				2×01		2×10	
3	3×000			1×001	2×010		1×100	1×101
4	2×0000		1×0001	1×0010	1×0100	1×0101	1×1000	1×1010
5	1×00000	1×00001	1×00010	1×00101	1×01000	1×01010	1×10000	1×10100
6	1×000001	1×000010	1×000101	1×001010	1×010000	1×010100	1×100000	1×101000
7	1×0000010	1×0000101	1×0001010	1×0010100	1×0100000	1×0101000	1×1000001	1×1010000
8	1×00000101	1×00001010	1×00010100	1×00101000	1×01000001	1×01010000	1×10000010	1×10100000

About N^2 counters to send!

CSE informally (2)

The substring enumeration, from the decompressor's point of view:

Length	Substrings
0	? $\times \epsilon$
1	
$\vdots$	$\vdots$
?	

$N =$?

CSE informally (2)

The substring enumeration, from the decompressor's point of view:

Length	Substrings
0	?$\times\epsilon$
1	
$\vdots$	$\vdots$
?	

$N =$?

Knowing: $N \in Naturals$

CSE informally (2)

The substring enumeration, from the decompressor's point of view:

Length	Substrings
0	? $\times \epsilon$
1	
$\vdots$	$\vdots$
?	

$N =$?

Knowing: $N \in Naturals$

Receive: $N = 8$.

CSE informally (3)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8\times\epsilon$	
1	$?\times 0$	$?\times 1$
2		
3		
$\vdots$	$\vdots$	
8		

$C_0 = ?$

CSE informally (3)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8\times\epsilon$	
1	?$\times$0	?$\times$1
2		
3		
⋮	⋮	
8		

$C_0 =$?

Knowing: $0 \leq C_0 \leq 8$

CSE informally (3)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8 \times \epsilon$	
1	?×0	?×1
2		
3		
⋮	⋮	
8		

$C_0 =$?

Knowing: $0 \leq C_0 \leq 8$

Receive: $C_0 = 6$.

CSE informally (4)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8{\times}\epsilon$	
1	$6{\times}0$	$?{\times}1$
2		
3		
⋮	⋮	
8		

$C_1 = ?$

CSE informally (4)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8 \times \epsilon$	
1	6×0	$? \times 1$
2		
3		
$\vdots$	$\vdots$	
8		

$C_1 = ?$

Knowing: $C_0 + C_1 = N = 8$

CSE informally (4)

The substring enumeration, from the decompressor's point of view:

Length	Substrings	
0	$8{\times}\epsilon$	
1	$6{\times}0$	?$\times 1$
2		
3		
⋮	⋮	
8		

$C_1 =$?

Knowing: $C_0 + C_1 = N = 8$

Deduce: $C_1 = 2$.

CSE informally (5)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	?$\times$00	?$\times$01	?$\times$10	?$\times$11
3				
$\vdots$	$\vdots$			
8				

$C_{00} =$?

CSE informally (5)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	?×00	?×01	?×10	?×11
3				
⋮	⋮			
8				

$C_{00} =$?

Naïvely knowing: $0 \leq C_{00} \leq 6$ (but $C_{00} = 0 \Rightarrow C_{01} = 6; \ldots; C_{00} = 3 \Rightarrow C_{01} = 3$)

CSE informally (5)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	?$\times$00	?$\times$01	?$\times$10	?$\times$11
3				
⋮	⋮			
8				

$C_{00} =$?

Naïvely knowing: $0 \leq C_{00} \leq 6$ (but $C_{00} = 0 \Rightarrow C_{01} = 6; \ldots; C_{00} = 3 \Rightarrow C_{01} = 3$)

So, knowing: $4 \leq C_{00} \leq 6$

CSE informally (5)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	$6\times$0		$2\times$1	
2	?×00	?×01	?×10	?×11
3				
⋮	⋮			
8				

$C_{00} = ?$

Naïvely knowing: $0 \leq C_{00} \leq 6$ (but $C_{00} = 0 \Rightarrow C_{01} = 6; \ldots; C_{00} = 3 \Rightarrow C_{01} = 3$)

So, knowing: $4 \leq C_{00} \leq 6 \ldots$ receive: $C_{00} = 4$.

CSE informally (6)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	$?\times 01$	$?\times 10$	$?\times 11$
3				
⋮		⋮		
8				

$C_{01} = ?$

CSE informally (6)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	$?\times 01$	$?\times 10$	$?\times 11$
3				
⋮	⋮			
8				

$C_{01} = ?$

Knowing: $C_{00} + C_{01} = C_0$

CSE informally (6)

The substring enumeration, from the decompressor's point of view:

Length	**Substrings**			
0	$8{\times}\epsilon$			
1	$6{\times}0$		$2{\times}1$	
2	$4{\times}00$	?$\times$01	?$\times$10	?$\times$11
3				
⋮	⋮			
8				

$C_{01} =$?

Knowing: $C_{00} + C_{01} = C_0$

Deduce: $C_{01} = 2$.

CSE informally (7)

The substring enumeration, from the decompressor's point of view:

Length	**Substrings**			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	2×01	$?\times 10$	$?\times 11$
3				
⋮	⋮			
8				

$C_{10} = ?$

CSE informally (7)

The substring enumeration, from the decompressor's point of view:

Length	**Substrings**			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	2×01	$?\times 10$	$?\times 11$
3				
⋮	⋮			
8				

$C_{10} = ?$

Knowing: $C_{00} + C_{10} = C_0$

CSE informally (7)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	2×01	$?\times 10$	$?\times 11$
3				
⋮	⋮			
8				

$C_{10} = ?$

Knowing: $C_{00} + C_{10} = C_0$

Deduce: $C_{10} = 2$.

CSE informally (8)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	2×01	2×10	$?\times 11$
3				
⋮		⋮		
8				

$C_{11} = ?$

CSE informally (8)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8\times\epsilon$			
1	6×0		2×1	
2	4×00	2×01	2×10	$?\times 11$
3				
$\vdots$		$\vdots$		
8				

$C_{11} = ?$

Knowing: $C_{10} + C_{11} = C_1$

CSE informally (8)

The substring enumeration, from the decompressor's point of view:

Length	Substrings			
0	$8{\times}\epsilon$			
1	$6{\times}0$		$2{\times}1$	
2	$4{\times}00$	$2{\times}01$	$2{\times}10$	$?{\times}11$
3				
⋮	⋮			
8				

$C_{11} = ?$

Knowing: $C_{10} + C_{11} = C_1$

Deduce: $C_{11} = 0$.

CSE informally (9)

The substring enumeration, from the decompressor's point of view:

Length	Substrings					
0	$8{\times}\epsilon$					
1	$6{\times}0$				$2{\times}1$	
2	$4{\times}00$		$2{\times}01$		$2{\times}10$	
3	?×000	?×001	?×010	?×011	?×100	?×101
⋮	⋮					
8						

$C_{000} = ?$

And so on. . .

Definition of substring

Substring w *occurs* at *position* p in $\mathbf{D}$, denoted by $w \in_p \mathbf{D}$, if:

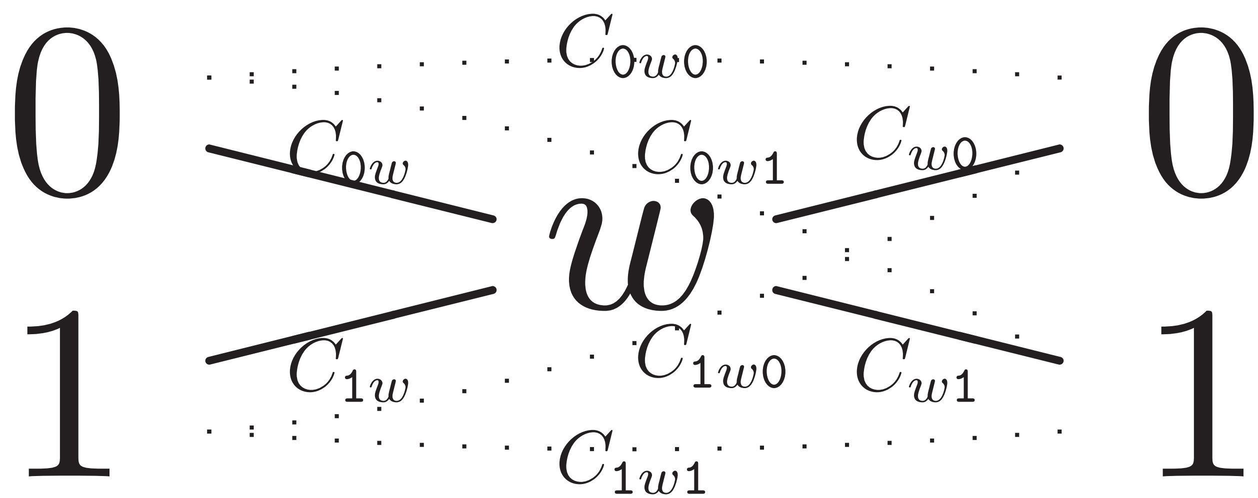
$$\exists\, u \in \{0,1\}^*,\ v \in \{0,1\}^\infty.\ |u| = p < N \ \text{ and } \ u\, w\, v = \mathbf{D}^\infty.$$

Notation: C_w is the *number* of occurrences of w in $\mathbf{D}$.

Formally, C_w is the number of positions where w occurs.

Butterfly (1)

The four extensions of the *core* w and the associated counters:



Butterfly (2)

Counter	C_w	C_{0w}	C_{1w}	C_{w0}	C_{w1}	C_{0w0}	C_{0w1}	C_{1w0}	C_{1w1}

Equations relating the counters together:

$$C_{0w} + C_{1w} \quad = \quad C_w \quad = \quad C_{w0} + C_{w1}$$

$$\begin{array}{lcl} C_{0w} & = & C_{0w0} + C_{0w1} \\ C_{1w} & = & C_{1w0} + C_{1w1} \end{array} \qquad \begin{array}{lcl} C_{w0} & = & C_{0w0} + C_{1w0} \\ C_{w1} & = & C_{0w1} + C_{1w1} \end{array}$$

Main algorithm

Compression of $\mathbf{D}$ using substring enumeration,
where $\mathbf{D} \in \{0,1\}^+$ and $N = |\mathbf{D}|$:

Send N
Send C_0
For l **:=** 2 **to** N **do**
  For every core w in the CST such that $|w| = l - 2$ **do**
    Send $0w0$ (and deduce $0w1$, $1w0$, and $1w1$)
Send rank of $\mathbf{D}$

Butterfly (3)

Each unknown counter has to be non-negative:

$$\begin{array}{rcl} C_{0w0} & \geq & 0 \\ C_{0w1} & \geq & 0 \\ C_{1w0} & \geq & 0 \\ C_{1w1} & \geq & 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} C_{0w0} & \geq & 0 \\ C_{0w0} & \leq & C_{0w} \\ C_{0w0} & \leq & C_{w0} \\ C_{0w0} & \geq & C_{0w} - C_{w1} \end{array}$$

which results in the following bounds:

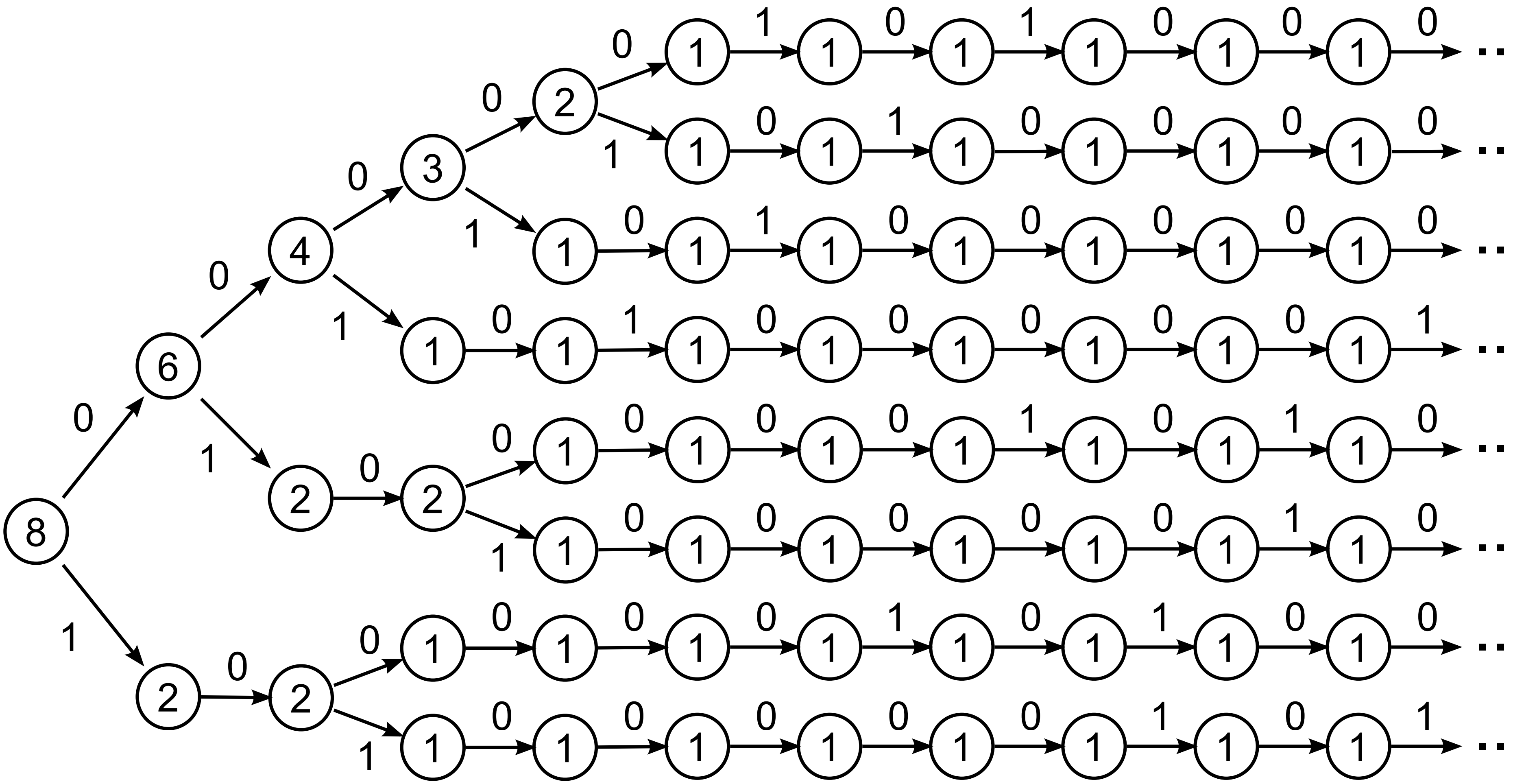
$$\max(0, C_{0w} - C_{w1}) \quad \leq \quad C_{0w0} \quad \leq \quad \min(C_{0w}, C_{w0}).$$

Reasons behind the actual compression

- The set of l-bit strings is a good summary for the set of $(l + 1)$-bit strings
- Fewer counters on higher levels
- Butterflies take all the available local information into account
- A majority of the butterflies (almost 78%) are trivial
- For almost all butterflies ($>$ 99%), the min–max range is narrow (at most 23 possibilities)

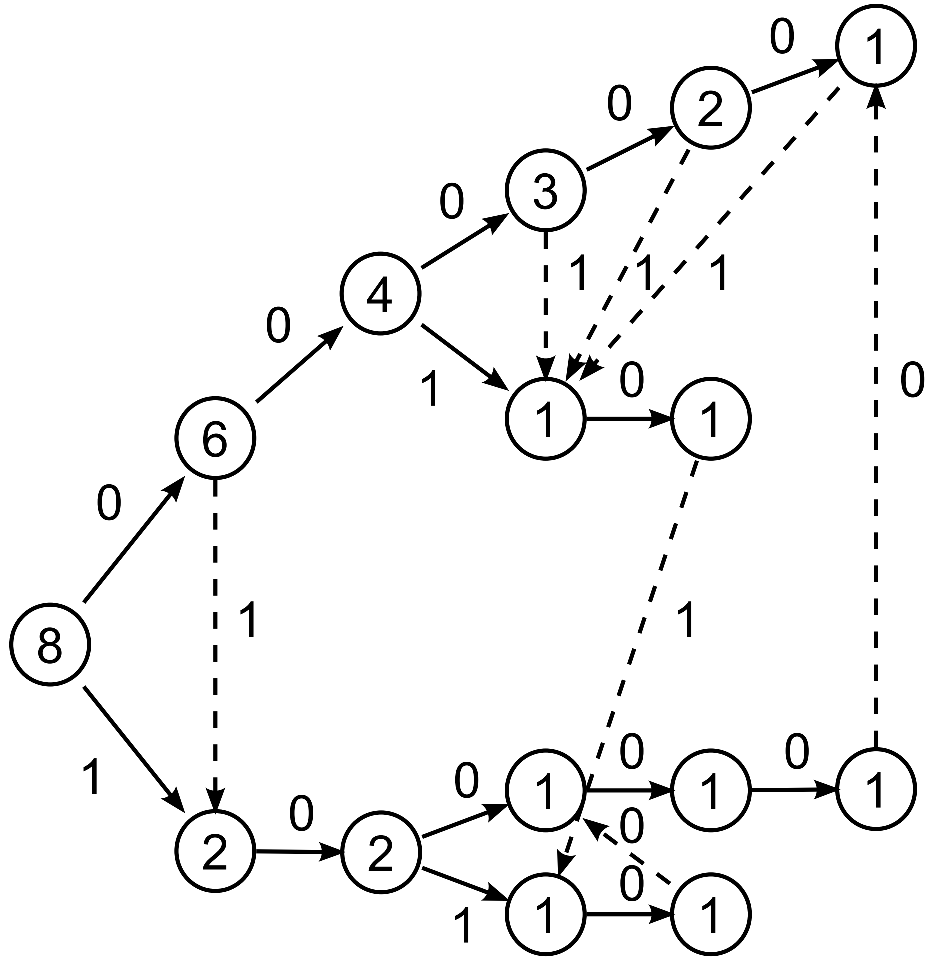
IST: infinite substring tree

The IST for $\mathbf{D} = \texttt{01000001}$.



CST: compact substring tree

The CST for $\mathbf{D} = $ 01000001, which is isomorphic to the IST.



A CST has $2N - 1$ nodes, if $\mathbf{D}$ is non-repetitive [DY11]. Fewer if repetitive.

Main Algorithm

Compression of $\mathbf{D} \in \{0,1\}^+$ using substring enumeration:

Send N
Send C_0
For l **:=** 2 **to** N **do**
  For every core w in the CST such that $|w| = l - 2$ **do**
    Send $0w0$ (and deduce $0w1$, $1w0$, and $1w1$)
Send rank of $\mathbf{D}$

At most $2N - 1$ numbers to send!

Experimental results (1)

Techniques being compared:

- **Gzip**: `gzip` set at maximal compression
- **BWT**: the Burrows-Wheeler transform (from [2])
- **PPM**: PPM*C (from [2])
- $\overline{\textbf{Btf}}$: prototype, 32 kB blocks, flat predictions
- **Btf** : prototype, 32 kB blocks, adaptive predictions
- **BTF**: prototype, 1 MB blocks, adaptive predictions

[2] J. G. Cleary and W. J. Teahan. Unbounded length contexts for PPM. *The Computer Journal*, 40(2/3):67-75, 1997.

Experimental results (2)

File	Gzip	BWT	PPM	$\overline{\text{Btf}}$	Btf	BTF
bib	2.51	2.07	**1.91**	2.54	2.56	1.98
book1	3.25	2.49	2.40	3.14	3.06	**2.27**
book2	2.70	2.13	2.02	2.74	2.72	**1.98**
geo	5.34	**4.45**	4.83	6.03	5.52	5.35
news	3.06	2.59	**2.42**	3.33	3.32	2.52
obj1	**3.84**	3.98	4.00	5.10	4.46	4.46
obj2	2.63	2.64	**2.43**	3.03	3.02	2.71
paper1	2.79	2.55	**2.37**	2.79	2.80	2.54
paper2	2.89	2.51	**2.36**	2.77	2.77	2.41
paper3	3.11	—	—	2.95	2.96	**2.73**
paper4	3.33	—	—	**3.17**	3.20	3.20
paper5	3.34	—	—	**3.29**	3.33	3.33
paper6	2.77	—	—	2.75	2.76	**2.65**
pic	0.82	0.83	0.85	2.05	0.79	**0.77**
progc	2.68	2.58	**2.40**	2.76	2.77	2.60
progl	1.80	1.80	**1.67**	1.90	1.89	1.71
progp	1.81	1.79	**1.62**	1.99	1.96	1.78
trans	1.61	1.57	**1.45**	2.16	2.07	1.60

Links of CSE with other techniques

- With prediction by partial matching (PPM)
  $\Rightarrow$ knowing C_{w0} and C_{w1} makes order-$|w|$ predictions possible
  $\Rightarrow$ predicts order-$(|w|+1)$ models, not individual bits

- With anti-dictionaries
  $\Rightarrow$ if w is an anti-word, then $C_w = 0$

- With LZ77 and LZ78
  $\Rightarrow$ recurring words lead to highly reliable / certain predictions

- With the Burrows-Wheeler transform (BWT)
  $\Rightarrow$ block-based
  $\Rightarrow$ prediction contexts grow up to full block

Subsequent work by me et al

CSE in linear time and space [DY11]

We show that the CST is made of three different kinds of nodes:

- the root n_ϵ,
- nodes n_{0w}, where $w \in V(\mathbf{D})$, and
- nodes n_{1w}, where $w \in V(\mathbf{D})$.

$w \in V(\mathbf{D})$ iff both $0w$ and $1w$ occur in $\mathbf{D}$.

We also show that $|V(\mathbf{D})| = N - 1$.

This implies that the CST has $2N - 1$ nodes.

Subsequent work by me et al

Universality for Markovian sources [DY11]

Adapted pseudo-code for CSE:

Send N; **Send** C_0; **Send** $\mathrm{rank}(\mathbf{D})$;
For $l := 2$ **to** N **do**
    For all $w \in I(\mathbf{D})$ such that $|w| = l - 2$ **do**
        If $|w| < l_{\mathrm{t}}$ **then**
            Predict and **Send** C_{0w0} **uniformly**
        Else
            Predict and **Send** C_{0w0} **combinatorially**;

Subsequent work by me et al

Universality for Markovian sources [DY11]

Selecting l_{t} based on encoding costs:

$$\begin{aligned} \mathcal{K}_{\mathrm{CSE}|_{l_{\mathrm{t}}}}(\mathbf{D}) &= \text{length of compressed file using threshold } l_{\mathrm{t}}, \\ l_{\mathrm{opt}} &= \arg\min_{l_{\mathrm{t}}} \quad \mathcal{K}_{\mathrm{CSE}|_{l_{\mathrm{t}}}}(\mathbf{D}), \\ \mathcal{K}_{\mathrm{CSE}}(\mathbf{D}) &= \mathcal{K}_{\mathrm{CSE}|_{l_{\mathrm{opt}}}}(\mathbf{D}). \end{aligned}$$

Subsequent work by me et al

Universality for Markovian sources [DY11]

Uniform prediction

Given that

$$\max(0, C_{0w} - C_{w1}) \leq C_{0w0} \leq \min(C_{0w}, C_{w0}),$$

each of the possible values of C_{0w0} is assigned probability

$$\frac{1}{\min(C_{0w}, C_{w0}) - \max(0, C_{0w} - C_{w1}) + 1}.$$

Subsequent work by me et al

Universality for Markovian sources [DY11]

Combinatorial prediction

Given that

$$\max(0, C_{0w} - C_{w1}) \leq C_{0w0} \leq \min(C_{0w}, C_{w0}),$$

each of the possible values of C_{0w0} is assigned probability

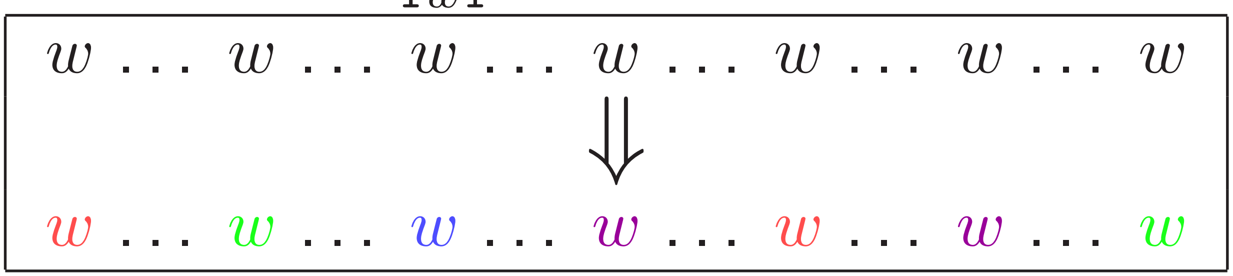
$$\frac{\binom{C_w}{C_{0w0},\ C_{0w1},\ C_{1w0},\ C_{1w1}}}{\displaystyle\sum_{C_{0w0}=\max(0,C_{0w}-C_{w1})}^{\min(C_{0w},C_{w0})} \binom{C_w}{C_{0w0},\ C_{0w1},\ C_{1w0},\ C_{1w1}}}.$$

Subsequent work by me et al

Universality for Markovian sources [DY11]

Rationale behind combinatorial prediction. Given a value of C_{0w0}, we partition the C_w occurrences of w into:

- C_{0w0} occurrences of $0w0$,
- C_{0w1} occurrences of $0w1$,
- C_{1w0} occurrences of $1w0$,
  and
- C_{1w1} occurrences of $1w1$.



Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

Example: compression of executable code

`...!W*...`

$$\ldots\overbrace{\texttt{00}\textcolor{red}{\texttt{100}}\texttt{001}}\overbrace{\texttt{0101011}\textcolor{red}{\texttt{1}}}\overbrace{\textcolor{red}{\texttt{00}}\texttt{101010}}\ldots$$

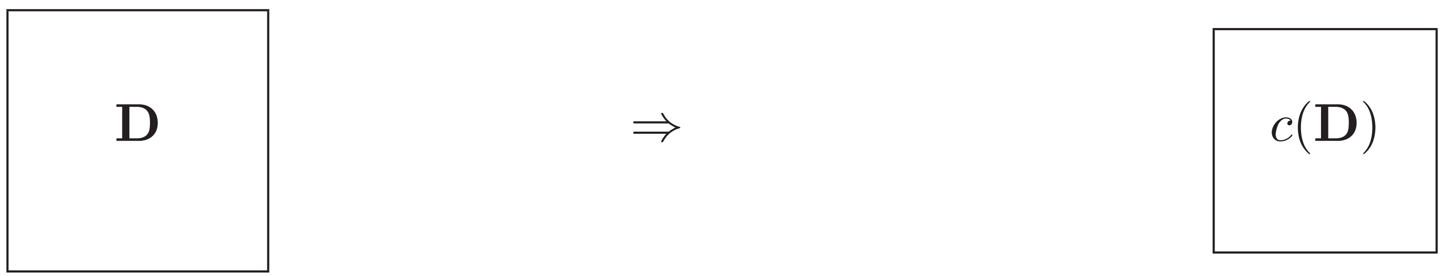
Bits on different phases (i.e. that have different significances) might have different statistics!

Subsequent work by me et al

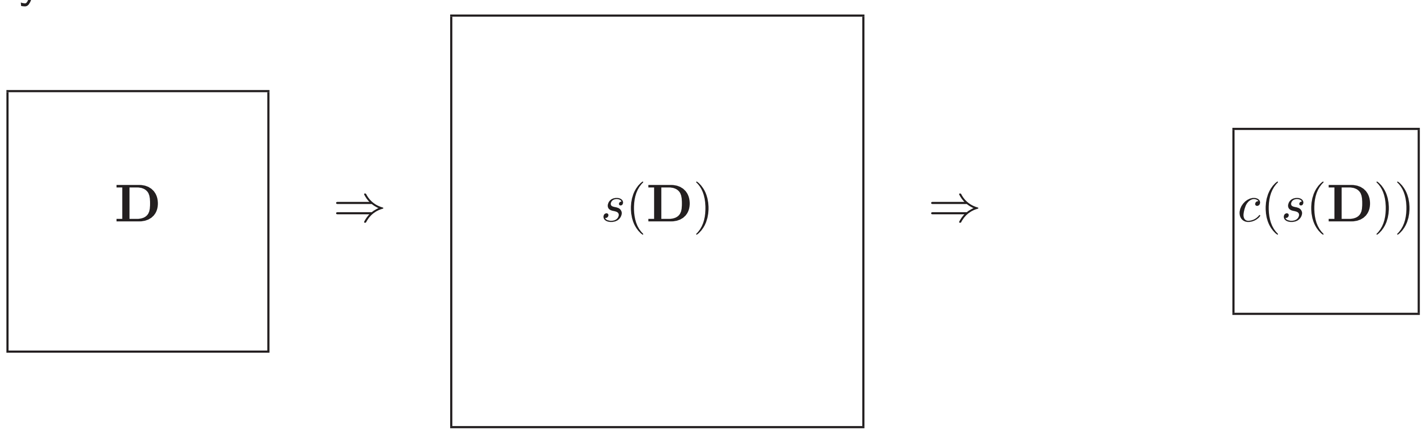
Phase awareness using synchronization codes [D10,D11]

Goal: inserting synchronization bits in the data.

Instead of:



we intend to try:



Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

k	**Synchronization Scheme**
1	_ _ _ _ _ _ _ _ 0
2	_ _ _ _ _ _ _ _ 0 1
3	_ _ _ _ _ _ _ _ 0 1 1
4	_ _ _ _ _ _ _ _ 0 1 1 1
5	_ _ _ _ _ _ 0 _ _ 0 1 1 1

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

Example of a reliable 5-bit synchronization scheme:

```
_ _ _ _ _ _ 0 _ _ 0 1 1 1
1 _ _ _ _ _ _ 0 _ _ 0 1 1
1 1 _ _ _ _ _ _ 0 _ _ 0 1
1 1 1 _ _ _ _ _ _ 0 _ _ 0
0 1 1 1 _ _ _ _ _ _ 0 _ _
_ 0 1 1 1 _ _ _ _ _ _ 0 _
_ _ 0 1 1 1 _ _ _ _ _ _ 0
0 _ _ 0 1 1 1 _ _ _ _ _ _
_ 0 _ _ 0 1 1 1 _ _ _ _ _
_ _ 0 _ _ 0 1 1 1 _ _ _ _
_ _ _ 0 _ _ 0 1 1 1 _ _ _
_ _ _ _ 0 _ _ 0 1 1 1 _ _
_ _ _ _ _ 0 _ _ 0 1 1 1 _
```

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

File	BWT	PPM	Anti	CSE	S-1	S-2	S-3	S-4	S-5
bib	2.07	1.91	2.56	1.98	1.95	1.92	1.92	1.91	**1.90**
book1	2.49	2.40	3.08	2.39	2.38	**2.37**	2.39	2.42	2.43
book2	2.13	**2.02**	2.81	2.07	2.06	2.06	2.06	2.05	2.04
geo	**4.45**	4.83	6.22	5.35	5.21	4.98	4.81	4.70	4.63
news	2.59	**2.42**	3.42	2.52	2.49	2.46	2.45	2.51	2.55
obj1	**3.98**	4.00	4.87	4.46	4.53	4.43	4.32	4.24	4.17
obj2	2.64	**2.43**	3.61	2.71	2.69	2.59	2.53	2.49	2.47
paper1	2.55	**2.37**	3.17	2.54	2.51	2.48	2.47	2.46	2.44
paper2	2.51	**2.36**	3.14	2.41	2.39	2.38	2.38	2.37	**2.36**
paper3	—	—	—	2.73	2.70	2.69	2.68	2.67	**2.65**
paper4	—	—	—	3.20	3.16	3.13	3.13	3.10	**3.07**
paper5	—	—	—	3.33	3.29	3.27	3.24	3.22	**3.19**
paper6	—	—	—	2.65	2.61	2.58	2.56	2.55	**2.52**
pic	0.83	0.85	1.09	**0.77**	0.84	0.83	0.83	0.84	0.83
progc	2.58	**2.40**	3.18	2.60	2.58	2.54	2.52	2.50	2.48
progl	1.80	1.67	2.24	1.71	1.70	1.69	1.68	1.67	**1.66**
progp	1.79	**1.62**	2.27	1.78	1.76	1.73	1.71	1.70	1.68
trans	1.57	**1.45**	1.94	1.60	1.58	1.53	1.52	1.50	1.48

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

Observations:

- The more numerous the synchronization bits, the better the compression (usually).
- Even non-reliable synchronization helps.
- No dramatic improvement when reliable synchronization is reached.
- Even text-like data is better compressed, but to a lesser extent than binary data.
- `pic` is already organized at the bit level; adding synchronization bits does not help.

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

	n	k	**Synchronization scheme**
ISITA'10	—	0	_ _ _ _ _ _ _ _
	—	1	_ _ _ _ _ _ _ _ 0
	—	2	_ _ _ _ _ _ _ _ 0 1
	—	3	_ _ _ _ _ _ _ _ 0 1 1
	—	4	_ _ _ _ _ _ _ _ 0 1 1 1
	13	5	_ _ _ _ _ _ 0 _ _ 0 1 1 1
DCC'11	12	8	_ _ _ 0 _ _ 1 0 0 _ _ 1 _ 1 1 0
	11	8	_ _ _ 0 _ 0 _ _ 1 1 0 _ _ 1 1 0
	10	10	_ _ _ _ 0 _ 0 1 1 _ _ _ 0 1 0 0 1 1
	9	10	_ _ _ _ 0 0 0 1 1 _ _ _ _ 0 1 0 1 1
	8	15	_ _ _ 0 0 0 1 0 _ _ _ 1 1 1 0 1 _ _ 1 1 0 0 1
	7	20	1 1 0 1 _ 1 _ 1 1 0 0 _ 0 _ 0 1 0 0 _ 0 _ 1 1 0 0 _ 1 _

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

```
_ _ _ 0 0 0 1 0 |_ _ _ 1 1 1 0 1 _ _ 1 1 0 0 1
_ _ 0 0 0 1 0 _ |_ _ 1 1 1 0 1 _ _ 1 1 0 0 1 _
_ 0 0 0 1 0 _ _ |_ 1 1 1 0 1 _ _ 1 1 0 0 1 _ _
0 0 0 1 0 _ _ _ |1 1 1 0 1 _ _ 1 1 0 0 1 _ _ _
0 0 1 0 _ _ _ 1 |1 1 0 1 _ _ 1 1 0 0 1 _ _ _ 0
0 1 0 _ _ _ 1 1 |1 0 1 _ _ 1 1 0 0 1 _ _ _ 0 0
1 0 _ _ _ 1 1 1 |0 1 _ _ 1 1 0 0 1 _ _ _ 0 0 0
0 _ _ _ 1 1 1 0 |1 _ _ 1 1 0 0 1 _ _ _ 0 0 0 1
_ _ _ 1 1 1 0 1 |_ _ 1 1 0 0 1 _ _ _ 0 0 0 1 0
_ _ 1 1 1 0 1 _ |_ 1 1 0 0 1 _ _ _ 0 0 0 1 0 _
_ 1 1 1 0 1 _ _ |1 1 0 0 1 _ _ _ 0 0 0 1 0 _ _
1 1 1 0 1 _ _ 1 |1 0 0 1 _ _ _ 0 0 0 1 0 _ _ _
1 1 0 1 _ _ 1 1 |0 0 1 _ _ _ 0 0 0 1 0 _ _ _ 1
1 0 1 _ _ 1 1 0 |0 1 _ _ _ 0 0 0 1 0 _ _ _ 1 1
0 1 _ _ 1 1 0 0 |1 _ _ _ 0 0 0 1 0 _ _ _ 1 1 1
1 _ _ 1 1 0 0 1 |_ _ _ 0 0 0 1 0 _ _ _ 1 1 1 0
_ _ 1 1 0 0 1 _ |_ _ 0 0 0 1 0 _ _ _ 1 1 1 0 1
_ 1 1 0 0 1 _ _ |_ 0 0 0 1 0 _ _ _ 1 1 1 0 1 _
1 1 0 0 1 _ _ _ |0 0 0 1 0 _ _ _ 1 1 1 0 1 _ _
1 0 0 1 _ _ _ 0 |0 0 1 0 _ _ _ 1 1 1 0 1 _ _ 1
0 0 1 _ _ _ 0 0 |0 1 0 _ _ _ 1 1 1 0 1 _ _ 1 1
0 1 _ _ _ 0 0 0 |1 0 _ _ _ 1 1 1 0 1 _ _ 1 1 0
1 _ _ _ 0 0 0 1 |0 _ _ _ 1 1 1 0 1 _ _ 1 1 0 0
```

Subsequent work by me et al

Phase awareness using synchronization codes [D10,D11]

	ISITA'10									DCC'11					
Bits/Car.	**BWT**	**PPM**	**Anti**	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$n=13$	$n=12$	$n=11$	$n=10$	$n=9$	$n=8$	$n=7$
bib	2.07	1.91	2.56	1.98	1.95	1.92	1.92	1.91	1.90	1.89	1.89	1.89	1.89	1.88	1.88
book1	2.49	2.40	3.08	2.27	2.26	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.29	2.33
book2	2.13	2.02	2.81	1.98	1.96	1.95	1.95	1.94	1.94	1.93	1.93	1.93	1.93	1.93	1.95
geo	4.45	4.83	6.22	5.35	5.21	4.98	4.81	4.70	4.63	4.58	4.59	4.58	4.58	4.58	4.57
news	2.59	2.42	3.42	2.52	2.49	2.46	2.45	2.44	2.43	2.43	2.43	2.43	2.43	2.42	2.42
obj1	3.98	4.00	4.87	4.46	4.53	4.43	4.32	4.24	4.17	4.03	4.05	4.02	4.01	4.00	3.99
obj2	2.64	2.43	3.61	2.71	2.69	2.59	2.53	2.49	2.47	2.45	2.46	2.45	2.45	2.45	2.44
paper1	2.55	2.37	3.17	2.54	2.51	2.48	2.47	2.46	2.44	2.41	2.41	2.41	2.41	2.41	2.41
paper2	2.51	2.36	3.14	2.41	2.39	2.38	2.38	2.37	2.36	2.35	2.35	2.34	2.35	2.34	2.34
paper3	—	—	—	2.73	2.70	2.69	2.68	2.67	2.65	2.63	2.63	2.63	2.63	2.63	2.63
paper4	—	—	—	3.20	3.16	3.13	3.13	3.10	3.07	3.02	3.02	3.02	3.02	3.01	3.01
paper5	—	—	—	3.33	3.29	3.27	3.24	3.22	3.19	3.12	3.13	3.12	3.12	3.11	3.10
paper6	—	—	—	2.65	2.61	2.58	2.56	2.55	2.52	2.50	2.50	2.49	2.50	2.49	2.49
pic	0.83	0.85	1.09	0.77	0.84	0.82	0.82	0.82	0.81	0.81	0.81	0.81	0.81	0.81	0.81
progc	2.58	2.40	3.18	2.60	2.58	2.54	2.52	2.50	2.48	2.44	2.44	2.44	2.44	2.44	2.43
progl	1.80	1.67	2.24	1.71	1.70	1.69	1.68	1.67	1.66	1.65	1.65	1.65	1.65	1.64	1.64
progp	1.79	1.62	2.27	1.78	1.76	1.73	1.71	1.70	1.68	1.66	1.66	1.66	1.66	1.66	1.65
trans	1.57	1.45	1.94	1.60	1.58	1.53	1.52	1.50	1.48	1.47	1.47	1.47	1.47	1.47	1.46

Using synchronization schemes stronger than those used in ISITA'10 does not achieve significant improvements.

Subsequent work by me et al

Explicit phase awareness [DB14]

`...!W*...` $\longmapsto$

$$\ldots\overbrace{00100001}\overbrace{01010111}\overbrace{00101010}\ldots$$

Subsequent work by me et al

Explicit phase awareness [DB14]

Experimental results (in bpc)

File	Gzip	BWT	PPM	CSE	+SC	+EPA
bib	2.51	2.07	1.91	1.98	1.88	**1.87**
book1	3.25	2.49	2.40	2.27	2.33	**2.24**
book2	2.70	2.13	2.02	1.98	**1.93**	**1.93**
geo	5.34	**4.45**	4.83	5.35	4.57	4.56
news	3.06	2.59	**2.42**	2.52	**2.42**	**2.42**
obj1	**3.84**	3.98	4.00	4.46	3.99	3.95
obj2	2.63	2.64	**2.43**	2.71	2.44	2.44
paper1	2.79	2.55	**2.37**	2.54	2.41	2.39
paper2	2.89	2.51	2.36	2.41	2.34	**2.33**

File	Gzip	BWT	PPM	CSE	+SC	+EPA
paper3	3.11	—	—	2.73	2.63	**2.61**
paper4	3.33	—	—	3.20	3.01	**2.96**
paper5	3.34	—	—	3.33	3.10	**3.05**
paper6	2.77	—	—	2.65	2.49	**2.47**
pic	0.82	0.83	0.85	**0.77**	0.81	0.81
progc	2.68	2.58	**2.40**	2.60	2.44	2.42
progl	1.80	1.80	1.67	1.71	1.64	**1.63**
progp	1.81	1.79	**1.62**	1.78	1.66	1.64
trans	1.61	1.57	**1.45**	1.60	1.47	**1.45**

Work by the community

Universality for stationary and Ergodic sources

Hidetoshi Yokoo. "Asymptotic optimal lossless compression via the CSE technique." In Proceedings of the International Conference on Data Compression, Communication and Processing, Palinuro, Italy, June 2011.

Work by the community

Improved bounds for special cores

Ken-ichi Iwata, Mitsuharu Arimura, and Yuki Shima. “An improvement in lossless data compression via substring enumeration.” In Proceedings of the IEEE/ACIS International Conference on Computer and Information Science, pages 219-223, Sanya, Hainan Island, China, May 2011.

Work by the community

Analysis of CSE's redundancy

Ken-ichi Iwata, Mitsuharu Arimura, and Yuki Shima. "On the maximum redundancy of CSE for i.i.d. sources." In Proceedings of the International Symposium on Information Theory and Applications, pages 489-492, Honolulu, Hawaii, USA, October 2012.

Ken-ichi Iwata and Mitsuharu Arimura. "Maximum Redundancy of Lossless Data Compression via Substring Enumeration with a Finite Alphabet." IEICE Technical Report, IT2013-55, 2014.

Ken-ichi Iwata, Mitsuharu Arimura, and Yuki Shima. "Evaluation of Maximum Redundancy of Data Compression via Substring Enumeration for k-th Order Markov Sources." In IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E97-A, no. 8, pages 1754-1760, 2014.

Ken-ichi Iwata and Mitsuharu Arimura. "Lossless data compression via substring enumeration for k-th order Markov sources with a finite alphabet." In Proceedings of the Data Compression Conference, pp. 452–452, Snowbird, Utah, USA, April 2015.

Work by the community

Link to anti-dictionary compression

Takahiro Ota and Hiroyoshi Morita. “On Antidictionary Coding Based on Compacted Substring Automaton.” In Proceedings of the International Symposium on Information Theory, pages 1754-1758, 2013.

Takahiro Ota and Hiroyoshi Morita. “On a Universal Antidictionary Coding for Stationary Ergodic Sources with Finite Alphabet.” In Proceedings of the International Symposium on Information Theory and Applications, pages 294-298, 2014.

Work by the community

Faster and more lightweight implementations

Kosumo Yamazaki, Hideaki Kaneyasu, and Hidetoshi Yokoo. "Efficient implementation of compression by substring enumeration." In IEICE Technical Report, IT2013-51, Jan. 2014. (in Japanese)

Sho Kanai, Hidetoshi Yokoo, Kosumo Yamazaki, and Hideaki Kaneyasu. "Efficient Implementation and Empirical Evaluation of Compression by Substring Enumeration." In IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E99-A, no. 2, pages 601-611, February 2016.

Shumpei Sakuma, Kazuyuki Narisawa, and Ayumi Shinohara. "Generalization of Efficient Implementation of Compression by Substring Enumeration." In Proceedings of the Data Compression Conference, page 630, Snowbird, Utah, USA, March 2016.

Work by the community

Direct handling of non-binary alphabets

Ken-ichi Iwata and Mitsuharu Arimura. "Lossless data compression via substring enumeration for k-th order Markov sources with a finite alphabet." In IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E99-A, no. 12, pages 2130-2135, 2016.

Shumpei Sakuma, Kazuyuki Narisawa, and Ayumi Shinohara. "Generalization of Efficient Implementation of Compression by Substring Enumeration." In Proceedings of the Data Compression Conference, page 630, Snowbird, Utah, USA, March 2016.

Work by the community

Two-dimensional CSE

Takahiro Ota and Hiroyoshi Morita. "Two-Dimensional Source Coding by Means of Subblock Enumeration." In Proceedings of the IEEE Symposium on Information Theory, pages 311-315, Aachen, Germany, July 2017.

Future work

- Showing the compression capacity when input data has a certain LZ or grammar compactness.
- Avoiding the forgetfulness of blockwise compression
- Exploitation of the existence of a Hamiltonian circuit
- Simplifying two-dimensional CSE

Questions?

Web page:
`http://www.ift.ulaval.ca/~dadub100/`