String Attractors: A unifying theory of repetitiveness

Dominik Kempa[1] Nicola Prezza[2]

[1]University of Helsinki

[2]University of Pisa

HALG, Amsterdam,
June 4-6, 2018

Based on **D. Kempa and N. Prezza. *At the roots of dictionary compression: String attractors.* STOC 2018**.

Background: Dictionary compression

Definition

Dictionary compression: Encoding of string that replaces repetitions with pointers to other occurrences.

Example: Lempel-Ziv '77 (LZ77)

LZ77 = Greedy left-to-right partition of text into longest previous factors.

$$T = \texttt{B}\,\texttt{A}\,\texttt{B}\,\texttt{B A B}\,\texttt{A B B}\,\texttt{B A B}$$

Encoding: (b,0),(a,0),(1,1),(1,3),(2,3),(4,3)

Background: Dictionary compression

Example: Run-length Burrows-Wheeler transform (RLBWT)

RLBWT = invertible text transformation defined as follows.

Input: text $T = \text{BANANA\$}$

1. Build a matrix with the text *rotations* as rows

B	A	N	A	N	A	$
A	N	A	N	A	$	B
N	A	N	A	$	B	A
A	N	A	$	B	A	N
N	A	$	B	A	N	A
A	$	B	A	N	A	N
$	B	A	N	A	N	A

2. Sort the rows

						L
$	B	A	N	A	N	A
A	$	B	A	N	A	N
A	N	A	$	B	A	N
A	N	A	N	A	$	B
B	A	N	A	N	A	$
N	A	$	B	A	N	A
N	A	N	A	$	B	A

3. Apply run-length compression to $L = \text{ANNB\$AA}$ (the last column)

Output: RLBWT = (1,A), (2,N), (1,B), (1,$), (2,A)

Background: Dictionary compression

Other (less known) dictionary compressors:

- (run-length) grammars (SLP)
- collage systems
- macro schemes
- word graphs (CDAWG)

Applications

Compression: reducing the size of data before archiving or transfer, e.g., over the network. Examples: 7-zip, gzip = LZ77.

Compressed computation: supporting operations on data structures taking space close to dictionary-compressed text. Example operations:

- random access
- pattern matching queries

String Attractors

New combinatorial object generalizing all known dictionary compressors.

Definition

A set $\Gamma \subseteq [1..n]$ is a *string attractor* of $T \in \Sigma^n$ if every substring of T has an occurrence containing an element of Γ.

Example

$$T = \text{CDA}\underline{\text{B}}\text{CC}\underline{\text{D}}\text{ABC}\underline{\text{CA}} \qquad \Gamma = \{3, 6, 10, 11\}$$

Theorem: "compressors **are** attractors"

Let $T \in \Sigma^n$ and let α be the output size of any the following dictionary compressors on T:

(1) (RL)SLP, (2) collage system, (3) LZ77,
(4) macro scheme, (5) RLBWT, (6) CDAWG.

Claim: T has a string attractor of size $\mathcal{O}(\alpha)$.
Example: $T =$ B A B B A B A B B B A B

String Attractors

Theorem (bad news)

Computing the smallest attractor is NP-complete and APX-hard.

But, the reduction *Compressors* $\rightarrow$ *Attractors* can be reversed!

Theorem:

Given a string $T \in \Sigma^n$ and a string attractor Γ of size γ for T, we can build

- a macro scheme for T of size $\mathcal{O}(\gamma \log(n/\gamma))$,
- a collage system for T of size $\mathcal{O}(\gamma \log(n/\gamma))$,
- an SLP for T of size $\mathcal{O}(\gamma \log^2(n/\gamma))$.

Consequence: many new (and easier proofs of existing) relations between sizes of dictionary compressors, for example,

$$z \in \mathcal{O}(r \log^2(n/r)),$$

where z (resp. r) is the size of LZ77 (resp. RLBWT).

String Attractors

String attractors carry enough information about the string to design data structures.

Theorem

If $T \in \Sigma^n$ has an attractor of size γ, then we can build a data structure of size

- $\mathcal{O}(\gamma \operatorname{polylog} n)$ w-bit words that can extract any length-ℓ substring of T in $\mathcal{O}(\ell \log(\sigma)/w + \log n/\log\log n)$ time.
- $\mathcal{O}(\gamma \log(n/\gamma))$ that, given a pattern $P[1..m]$, outputs all its occurrences in T in $\mathcal{O}(m \log n + \mathrm{occ} \log^{\epsilon} n)$ time.

The resulting data structures are **universal** thanks to reductions *Attractors* $\rightarrow$ *Compressors*, i.e., they translate to concrete data structures working on different compressed representations.

Thank You!