Compression Methods

neděle 30. listopadu 2025 17:56

How to compress a string? We will look at three measures of compression.

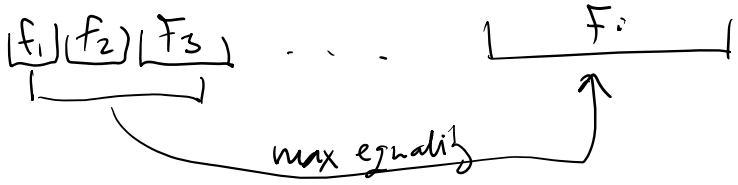
1) Lempel-Ziv (LZ) type compression:

$w \in \Sigma^N \qquad w = f_1 f_2 \dots f_z \qquad f_i \in \Sigma^*$

$f_i = w_{b_i \dots b_i + l_i - 1}$ where l_i is maximal s.t.

$\exists a_i \in \{1, \dots, \sum_{j<i} |f_j| \}$

$w_{a_i \dots a_i + l_i - 1} = w_{b_i \dots b_i + l_i - 1}$



f_i ... Fragment $\qquad z_w = z$ ... # of fragments.

2) grammar representation $\qquad w \in \Sigma^N$

Context-free grammar with rules $\quad A \to BC \quad B \to b$ $\qquad A, B, C \in$ variables, $b \in \Sigma$

(Chomsky normal form)

Consider the smallest grammar (# of rules) that generates w

& nothing else ..... g_w

3) substring diversity : $w \in \Sigma^N$ $\quad d_\ell(w) = |\{ w_{i,\dots,i+\ell-1},\ 1 \le i \le n-\ell+1 \}|$

$$\delta_w = \max_{\ell \in [n]} \frac{d_\ell(w)}{\ell}$$

Claim : $\quad \tilde{O}(z_w) = \tilde{O}(g_w) = \tilde{O}(\delta_w)$

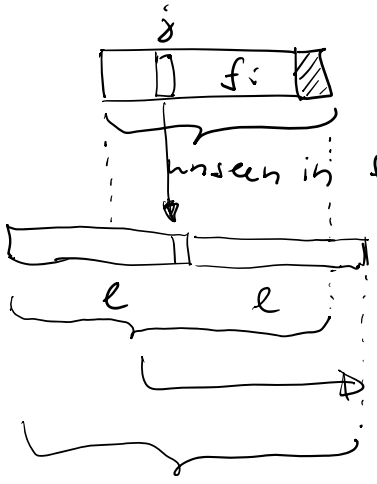
Pf : 1) $z_w \le 4(\delta_w + 1) \lg N$

Claim : $\forall \ell \le n \quad \sum_{k=1}^{\ell} k n_k \le 2\ell(\delta_w + 1)$ where n_k is the # of fragments f_i of length k.

[Raskhodnikova-Ron-Rubinfeld-Smith '13]

Pf : $\quad d_{2\ell} \le \delta_w \cdot 2\ell$

Consider strings of length 2ℓ. We will assign each position $j \in \{\ell, \dots, N-\ell\}$ that falls within a fragment f_i of size $\le \ell$ to a distinct substring of length 2ℓ.



$\ell \le j < N - \ell$

distinct from all previous substrings of length 2ℓ which have their middle point $< j$.

middle point $< j$.

$\Rightarrow$ each fragment of length $k \le \ell$ introduces k distinct substrings of length 2ℓ. (except very end/beginning of w)

$$\Rightarrow \sum_{k=1}^{\ell} k n_k - 2\ell \le d_{2\ell}(w) \le \delta_w \cdot 2\ell$$

$\ell \le j < N - \ell$

$$\Rightarrow \sum_{k=1}^{\ell} k n_k \le (\delta_w + 1) \cdot 2\ell$$

$$\Rightarrow \sum_{k=\frac{\ell}{2}}^{\ell} \frac{\ell}{2} n_k \le \sum_{k=1}^{\ell} k \cdot n_k \le (\delta_w + 1) \cdot 2\ell$$

$$\Rightarrow \sum_{k=\frac{\ell}{2}}^{\ell} n_k \le 4 \cdot (\delta_w + 1)$$

$$\Rightarrow z_w = \sum_{k=1}^{N} n_k \le 4(\delta_w + 1) \cdot \lg N$$

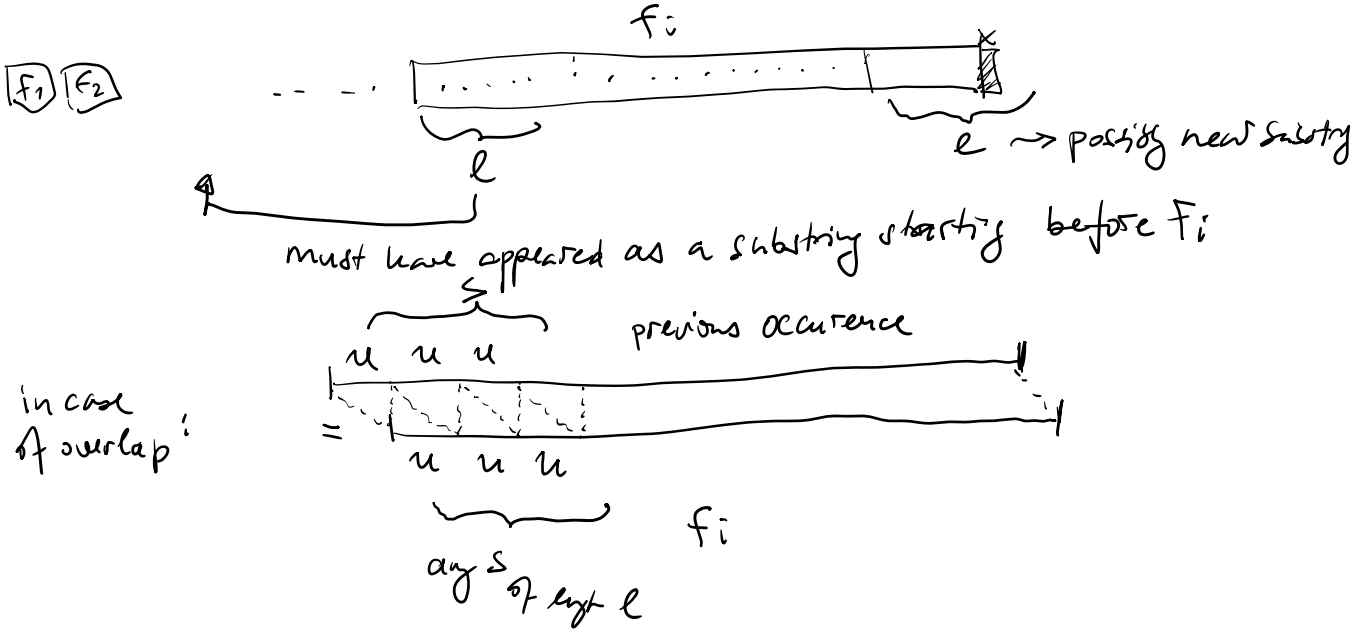
2) $\delta_w \le z_w$

Pf:

For any $\ell \in [N]$ $\quad \frac{d_\ell(w)}{\ell} \le \delta_w$

A fragment f_i of length $k \le \ell$ gives only $k \le \ell$ substrings of length ℓ that start in it.

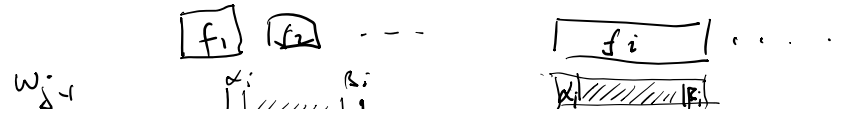
A fragment f_i of length $k > \ell$ gives at most ℓ distinct substrings that start in it:

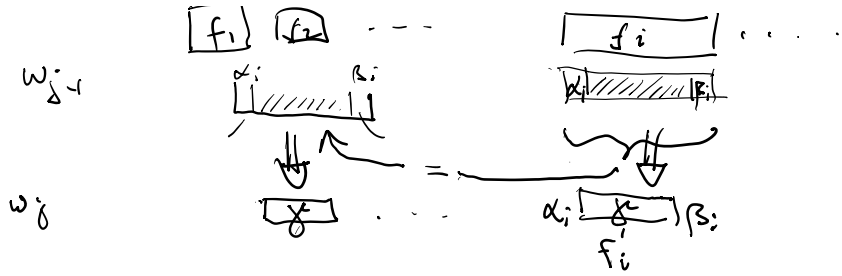


$\Rightarrow \quad d_\ell(w) \leq z_w \cdot \ell \quad \Rightarrow \quad \delta_w \leq z_w$ ∎

3) $g_w \leq O(z_w \cdot \lg N)$ $\quad w \in \Sigma^N$

[Jeż '16] start with factorization $f_1, \ldots f_z$, we will create iteratively a grammar for w. It is created in $O(\lg N)$ stages where in each stage we introduce $O(z)$ new rules. We start with $w_0 = w$ & create $w_1, w_2, \ldots$ where in stage j we create w_j from w_{j-1} by "compressing" some nonoverlapping pairs of w_{j-1}. The compression (news) is done by replacing a consecutive pair of letters by a symbol & adding the appropriate rule.
We compress w_{j-1} factor by factor left to right:





$|\alpha_i| = |\beta_i| = 1$

except for α_i & β_i which might be paired outside of f_i, the inner portion of f_i can be compressed as in its corresponding copy to the left.

(WLOG the corresponding copy starts at least two symbols to the left of f_i o/w remove the first letter of f_i & leave it singleton. This only happens for $f_i = a^k$.)

Whenever a singleton letter α_i or β_i appears next to another singleton letter, compress it using a new rule.

(In particular, if the inner portion of f_i is a single letter, compress it with neighboring singleton if there is one.)

Since #factors $\leq z$, #singletons created at each step is at most $3z \Rightarrow$ at most $\frac{3}{2}z$ new rules are created at each iteration.

Since we greedily compress neighboring singletons whenever they are created, the compression cannot leave two singletons next to each other. So each singleton must be flanked by a pair on either side.

$\Rightarrow$ Out of every 3 consecutive symbols in w_{i-1}, at least two get compressed into one.

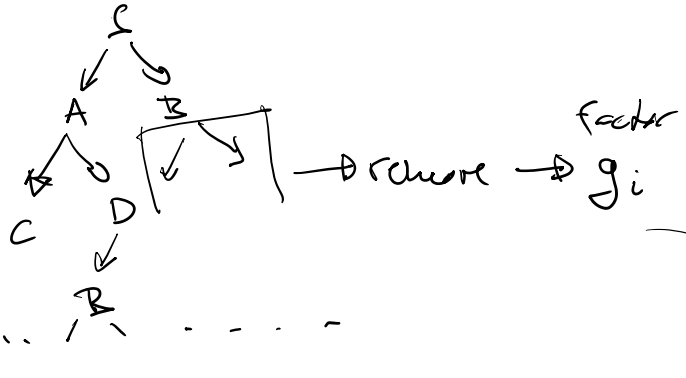
$\Rightarrow$ $|w_i| \leq \frac{2}{3}|w_{i-1}|$.

$\Rightarrow \quad |W_j| \le \frac{2}{3}|W_{j-1}|$.

$\Rightarrow$ After $\lg_{\frac{3}{2}} N$ iterations, W_j consists of only one singleton $\Rightarrow$ done ☐

4) $z_w \le 2g_w$

[Rytter '03] Consider a grammar of size g_w & a derivation tree of w:



going left to right, remove a subtree below a non-terminal if B occurs somewhere in the tree to the left

$\rightarrow$ gives a factorization $g_1 \cdots g_{\le 2g_w}$ of w

where each factor g_i corresponds to some nonterminal or a single letter & g_i appears in $g_1 \cdots g_{i-1}$

first occurrence of A might contribute by two factors $g_i \Rightarrow \#\text{fac.} \le 2g_w$

$A \rightarrow CD$

$\lg$: each inner node corresponds to a distinct nonterminal. # leaves $\le 2 \cdot$ #nonterm.

Let $f_1 f_2 \cdots f_{z_w}$ be LZ-factorization of w

Claim: $\forall i \quad |g_1 \cdots g_i| \le |F_1 \cdots f_i|$.

Pf: Induction on i: $i = \checkmark$

$i \rightarrow i+1$:

if $|g_1 \cdots g_i| = |f_1 \cdots f_i|$

then $|g_1 \cdots g_{i+1}| \leq |f_1 \cdots f_{i+1}|$

since LZ is greedy & f_{i+1} will contain at least g_{i+1}.

if $|g_1 \cdots g_i| < |f_1 \cdots F_i|$

then either g_{i+1} is already contained in $F_1 \cdots F_i$ or some suffix of g_{i+1} will be covered by f_{i+1}. Again by greediness $|g_1 \cdots g_{i+1}| \leq |f_1 \cdots f_{i+1}|$

$\Rightarrow z_w \leq g_w$