UNIVERSITY OF LIVERPOOL

Department of Computer Science
Sebastian Wild

Date: 2021-04-21
Version: 2021-04-21 15:34

Tutorial 7 for COMP 526 – Applied Algorithmics, Spring 2021

Problem 1 (Move-to-front transform)

Let $T = T[0..9) = \texttt{ABBACBAAA}$ be an input text over alphabet $\Sigma = \{\texttt{A}, \texttt{B}, \texttt{C}\}$. Apply the move-to-front transform to this input with initial queue content $Q = [\texttt{A}, \texttt{B}, \texttt{C}]$ and trace the content of Q throughout the execution.

Problem 2 (Lempel-Ziv-Welch compression)

Given word $w = \texttt{ASNXASNASNA}$ over the ASCII character set (relevant parts of ASCII are provided on the right). Construct, step by step, the Lempel-Ziv-Welch (LZW) factorization of w (i.e., the phrases encoded by one codeword) and provide the compressed representation of w; it suffices to show the encoded text C using integer numbers (no need for binary encodings).

Code	Character
65	A
...	...
78	N
...	...
83	S
...	...
88	X
...	...

Problem 3 (No Free Lunch)

Prove the following *no-free-lunch* theorems for lossless compression.

1. *Weak version:* For every compression algorithm A and $n \in \mathbb{N}$ there is an input $w \in \Sigma^n$ for which $|A(w)| \ge |w|$, i. e. the "compression" result is no shorter than the input.

   Hint: Try a proof by contradiction. There are different ways to prove this.

2. *Strong version:* For every compression algorithm A and $n \in \mathbb{N}$ it holds that

   $$\big|\{w \in \Sigma^{\le n} : |A(w)| < |w|\}\big| \;<\; \tfrac{1}{2} \cdot \big|\Sigma^{\le n}\big| .$$

   In words, less than half of all inputs of length at most n can be compressed below their original size.

   Hint: Start by determining $|\Sigma^{\le n}|$.

The theorems hold for every non-unary alphabet, but you can restrict yourself to the binary case, i.e., $\Sigma = \{\texttt{0}, \texttt{1}\}$.

We denote by $\Sigma^\star$ the set of all (finite) strings over alphabet Σ and by $\Sigma^{\le n}$ the set of all strings with size $\le n$. As domain of (all) compression algorithms, we consider the set of (all) *injective* functions in $\Sigma^\star \to \Sigma^\star$, i.e., functions that map any input string to some output string (encoding), where no two strings map to the same output.