

A simple algorithm for Lempel-Ziv factorization

*

19 maja 2022

Faktoryzacja Lempel-Ziv'a dla słowa w jest takim rozkładem $u_0u_1\dots u_k = w$, że każde u_i , za wyjątkiem możliwie ostatniego, jest albo najdłuższym prefiksem $u_iu_{i+1}\dots u_k$ i występuje jako podsłowo w $u_0u_1\dots u_i$, ale nie tylko jako sufix, albo jest pojedynczym symbolem, gdy takiego prefiksu nie ma.

Algorytm korzysta z tablicy Longest Previous Factor. Aby zrozumieć co to jest, weźmy taki najdłuższy czynnik słowa $w[1..i]$, równy m . Wtedy m musi być najdłuższym podsłowem słowa $w[1..i + |m| - 1]$, i to jego długość będzie występować w tej tablicy, na pozycji i -tej.

Gdy posiadamy tablicę LPF, wyznaczanie faktoryzacji nie jest trudne. Łatwo zauważyć, że “najdłuższy poprzedni czynnik”, to dokładnie taki czynnik jakiego potrzebujemy do faktoryzacji. Wystarczy zatem przejść po tablicy LPF zwracając kolejne czynniki, pomijając przy tym czynniki pośrednie, występujące pomiędzy tymi z faktoryzacji, oraz zamieniając wszystkie zera na jedynki w tablicy LPF, ponieważ faktoryzacja nie zawiera słów pustych. Algorytm wygląda następująco:

Algorithm 1 lempel_ziv_factorization

Require: LPF, n

Ensure: LZ

LZ $\leftarrow []$

pos $\leftarrow 1$

while pos $\leq n$ **do**

 push(max(1, LPF[pos]), LZ)

 pos $\leftarrow pos + \max(1, \text{LPF}[\text{pos}])$

end while
