

# 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 podślowo 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 podśłowem słowa  $w[1..i + |m| - 1]$ , a jego długość będzie na pozycji  $i$ -tej tej tablicy.

Gdy posiadamy tablicę LPF, to wyznaczanie faktoryzacji nie jest trudne, ponieważ “najdłuższy poprzedni czynnik”, to dokładnie taki czynnik jakiego potrzebujemy do faktoryzacji. Czyli, wystarczy przejść po tablicy LPF zwracając kolejne czynniki, pomijając przy tym czynniki pośrednie, występujące pomiędzy tymi z faktoryzacji. Algorytm zatem wygląda następująco:

---

**Algorithm 1** lempel\_ziv\_factorization

---

**Require:** LPF,  $n$

**Ensure:** LZ

```
pos ← 1
while pos ≤ n do
  push(max(1, LPF[pos]), LZ)
  pos ← pos + max(1, LPF[pos])
end while
```

---