Dawid Pawliczek
Lista 1, Zadanie 4

Treść zadania

Nadsłowem zbioru słów $\{w_1, w_2, \ldots, w_k\}$ nazywamy dowolne słowo W takie, że $\forall_{i=1,\ldots,k}\ w_i$ jest podsłowem W. Należy napisać algorytm, który dla danego zbioru *słów dwuliterowych* (tzn. $|w_i| = 2$ dla każdego i) znajduje *najkrótsze* jego nadsłowo (lub dowolne jedno z najkrótszych, gdy nie jest ono jednoznaczne).

Model grafowy

Każde słowo $w_i = a_i b_i$ interpretujemy jako skierowaną krawędź $a_i \to b_i$ w grafie $G = (V, E)$, gdzie V jest zbiorem wszystkich liter występujących w wejściu. Słowo $W = v_0 v_1 \ldots v_\ell$ jest nadsłowem wtedy i tylko wtedy, gdy przebiegając kolejne krawędzie $v_0 \to v_1,\ v_1 \to v_2, \ldots$ odwiedzamy *każdą* krawędź z E co najmniej raz. Szukamy zatem ścieżki (lub cyklu) Eulera w G, przy czym powtórne przejścia po krawędziach wydłużają wynik – interesuje nas najkrótsza ścieżka obejmująca wszystkie krawędzie.

Redukcja do cyklu Eulera

Spójność. Jeśli G nie jest spójny, konstruujemy rozwiązanie osobno dla każdej składowej, a następnie łączymy otrzymane słowa w dowolnej kolejności; ich konkatenacja wciąż zawiera wszystkie wejściowe krawędzie.

Niespełnienie warunku Eulera. W spójnej składowej istnieje ścieżka Eulera wtedy i tylko wtedy, gdy

$$\big|\{v : \mathrm{outdeg}(v) = \mathrm{indeg}(v) + 1\}\big| = \big|\{v : \mathrm{indeg}(v) = \mathrm{outdeg}(v) + 1\}\big| = 1,$$

a dla cyklu Eulera gdy $\mathrm{indeg}(v) = \mathrm{outdeg}(v)$ dla każdego v. Aby ujednolicić przypadki, dodajemy nowy wierzchołek ε (pusta litera) oraz tyle krawędzi $v \to \varepsilon$ lub $\varepsilon \to v$, ile potrzeba, by zrównoważyć stopnie wszystkich wierzchołków – wtedy cały graf *zawsze* posiada cykl Eulera. Litery ε pomijamy w końcowym słowie.

Algorytm

Algorithm 1 SUPERWORD($\{w_1, \ldots, w_k\}$)

$G = (V, E) \leftarrow (\varnothing, \varnothing)$
for all $w_i = a_i b_i$ **do**
    $V \leftarrow V \cup \{a_i, b_i\}$; $E \leftarrow E \cup \{a_i \to b_i\}$
end for
$W \leftarrow$ “”
for all spójna składowa $G' = (V', E')$ grafu G **do**
    if G' spełnia warunek cyklu Eulera **then**
        $P \leftarrow$ cykl Eulera w G' (np. alg. Hierholzera)
    else
        dodaj nowy wierzchołek ε do V'
        wyrównaj stopnie, dodając krawędzie $\varepsilon \leftrightarrow v$ w miarę potrzeby
        $P \leftarrow$ cykl Eulera w tak rozszerzonym G'
        usuń z P wszystkie wystąpienia ε
    end if
    $W \leftarrow W +$ litery ścieżki P
end for
return W

Złożoność

Niech $n = |V|$, $m = |E| = k$. Konstrukcja grafu i obliczenie stopni trwa $O(n + m)$. Dla każdej składowej wywołujemy algorytm Hierholzera $O(n' + m')$, co łącznie daje $O(n + m)$. Pamięć pomocnicza: $O(n + m)$.

Poprawność – szkic

1. **Pokrycie krawędzi.** Każde słowo powstaje ze ścieżki/cyklu Eulera, więc zawiera wszystkie krawędzie wejścia, zatem jest nadsłowem.

2. **Minimalność.** Algorytm Hierholzera przechodzi każdą krawędź dokładnie raz; ewentualne krawędzie do ε odpowiadają dokładnie niezbilansowanym przejściom, które i tak musiałyby zostać pokryte przez dodatkowe litery w dowolnym nadsłowie. Po pominięciu ε otrzymujemy zatem najkrótsze możliwe słowo.