

## 1 Nejkratší jehla končící na dané pozici

Pro vyřešení úlohy je třeba nejprve modifikovat algoritmus vytváření prohledávacího automatu. Když budeme vytvářet zkratkové hrany, bude se spolu s nimi předávat ještě jedna další hodnota. Tato hodnota bude obsahovat ukazatel na *nejkratší* suffix stavu, který je zároveň jehlou. Jinými slovy ukazuje na konec posloupnosti stavů vytvořené průchodem po zkratkových hranách (pro jednoduchost neexistují zkratkové hrany do prázdného stavu).

Tento přídavek nám zjevně asymptoticky nijak nezhorší časovou složitost stavby automatu, neboť jsme jen o konstantní množství operací rozšířili operaci přidání zkratkové hrany. Zároveň si můžeme uvědomit, že pokud se v nějaké pozici dostaneme do stavu, ze kterého vede takovýto ukazatel, určitě je na pozici ukazatele jehla, která na této pozici končí. Toto vyplývá z definice zkratkových hran a faktu, že k cíli ukazatele se vždy můžeme dostat po nich.

Zbývá pak jen rozmyslet, že tato jehla je nejkratší možná. Kdyby existovala nějaká kratší jehla končící na stejném místě jako ta naše, musela by tato kratší jehla nutně být suffixem naší jehly. Z toho ovšem vyplývá, že by z naší jehly do té kratší vedla zkratková hrana, což je v rozporu s definicí námi zavedeného ukazatele.

Vlastní algoritmus je pak již zřejmý. Načteme písmeno ze vstupu, přesuneme se do vhodného stavu, a podíváme se, zda tento stav obsahuje ukazatel. Pokud ano, nejkratší jehla končící na této pozici je reprezentována právě stavem, na který se ukazuje.

V každé iteraci provádíme jen konstantní množství operací, tedy samotný průchod senem má složitost lineární vůči jeho velikosti. Spolu se sestavením automatu je tedy výsledná časová složitost  $O(N + M)$ , kde  $N$  je velikost sena a  $M$  součet velikostí jehel.

## 2 Nejdelší jehla končící na dané pozici

Nejdelší jehlu získáme snadno přímo z algoritmu Aho-Corasickové. Sestavíme automat běžným způsobem a pak procházíme seno po znacích a při přechodu do stavu děláme vždy následující. Pokud je v daném stavu označen konec jehly, je tato jehla určitě nejdelší končící na dané pozici. Pokud tam označen není, podíváme se, kam z něj vede zkratková hrana. Existuje-li, nejdelší jehla je reprezentována stavem na konci této hrany; jinak na této pozici žádná jehla nekončí.

Správnost tohoto postupu vyplývá z pozorování, že všechny jehly končící na dané pozici se dají dosáhnout po zkratkových hranách a že průchodem po zkratkové hraně mohou vždy jen snížit velikost jehly.

Stejným argumentem jako v prvním případě má tento algoritmus asymptotickou časovou složitost  $O(N + M)$ .

### 3 Jehly začínající na dané pozici

Abychom zjistili, které jsou nejkratší (resp. nejdelší) jehly začínající na dané pozici, nejprve všechny jehly přepíšeme pozpátku (v  $O(M)$ ). Na těchto převrácených jehlách pak postavíme vyhledávací automat. Poté začneme prohledávat od konce sena a aplikujeme algoritmy popsané v sekcích **1** a **2**.

Snadno pak rozmyslíme, že nejkratší (resp. nejdelší) převrácená jehla končící na nějaké pozici ve zpětném vyhledávání je ekvivalentní nejkratší (resp. nejdelší) obyčejné jehle začínající na též pozici v původním seně.

Jelikož jsme využili předchozí algoritmus a pouze ho rozšířili o část trvající  $O(M)$  (potažmo  $O(N + M)$ , pokud neumíme prohledávat seno pozpátku), výsledná časová složitost asymptoticky zůstává  $O(N + M)$ .