

Vypracoval(a):

UČO:

Skupina:

Bonus [5 bodů]

Uvažme abecedu Σ a slovo $u \in \Sigma^*$. Definujme množinu slov $\text{nafoukni}(u)$ následovně:

$$\begin{aligned} \text{nafoukni}(u) &\stackrel{\text{def}}{=} \{u\} && \text{pokud } u = \epsilon \\ \text{nafoukni}(u) &\stackrel{\text{def}}{=} \{\sigma \cdot \sigma' \mid \sigma' \in \Sigma\} \cdot \text{nafoukni}(v) && \text{pokud } u = \sigma \cdot v, \text{ kde } \sigma \in \Sigma \text{ a } v \in \Sigma^*. \end{aligned}$$

Neformálně řečeno, množina $\text{nafoukni}(u)$ obsahuje slova, která jsou dvakrát delší než slovo u a mají k -tý znak ze slova u vždy na $(2k - 1)$ -té pozici. Například pro $u = aab$ platí

$$\text{nafoukni}(u) = \{aaaaba, aaaabb, aaabba, aaabbb, abaaba, abaabb, ababba, ababbb\}.$$

Uvažme jazyk $L \subseteq \Sigma^*$. Definujme jazyk \ddot{L} jakožto sjednocení všech množin $\text{nafoukni}(u)$, kde $u \in L$.

$$\ddot{L} \stackrel{\text{def}}{=} \bigcup_{u \in L} \text{nafoukni}(u)$$

Mějme zadaný konečný, deterministický, automat \mathcal{A} s totální přechodovou funkcí, který akceptuje jazyk $L(\mathcal{A}) = L$. Najděte algoritmus, který zkonstruuje konečný, deterministický, totální, automat $\ddot{\mathcal{A}}$, který akceptuje jazyk $L(\ddot{\mathcal{A}}) = \ddot{L}$. Algoritmus naprogramujte. Na výběr máte jazyky **Python**, **C**, **C++**, **Java**, **Perl**, **Pascal** a **Haskell**.

Vstupem pro váš program je textový soubor obsahující několik automatů oddělených od sebe prázdným řádkem. Každý automat je zadán následujícím způsobem. Na prvním řádku je n , počet stavů automatu. Množina stavů automatu je $\{1, \dots, n\}$. Na druhém řádku je m , počet znaků abecedy. Abeceda je dána prvními m znaky anglické abecedy, například pro $m = 4$ je abeceda $\{a, b, c, d\}$. Na třetím řádku je číslo iniciálního stavu. Na čtvrtém jsou čísla koncových stavů oddělená mezerou. Řádky 5 až $n + 4$ jsou řádky tabulkového zápisu automatu. Každý řádek tudíž obsahuje přesně m čísel z rozmezí 1 až n , která jsou od sebe oddělena mezerou.

Příklad vstupu:

```
5
3
1
5 1
2 1 3
3 2 4
4 3 5
5 4 1
1 5 2
```

Odpovídající automat:

	a	b	c
$\leftrightarrow 1$	2	1	3
2	3	2	4
3	4	3	5
4	5	4	1
$\leftarrow 5$	1	5	2

Výstupní formát je stejný jako vstupní. Odevzdávejte zdrojové kódy vašich programů. Hodnocení bude probíhat na základě výsledků testu. Je možné dostat částečné body za algoritmus i v případě, že kód neprojde testem – ale to jen pokud svůj kód vybavíte komentáři, ze kterých bude jasná myšlenka algoritmu.

V učebních materiálech naleznete také ukázkový vstupní soubor.