

Proiecte laborator 1 - LFA - Mate-Info

1 Simulare AFD

Dându-se un *automat finit determinist* și m cuvinte formate din alfabetul acestuia, testați acceptanța automatului la fiecare din ele.

Input (afd.in)

- Pe prima linie a fișierului se găsește un număr n - indicele maxim al unei stări din automat ($\{q_0, q_1, \dots, q_n\}$, $n + 1$ stări)
- Pe următoarea linie se află un vector de caracteristică a stărilor finale (de exemplu, pentru un automat cu $n = 5$ și q_3 singura stare finală avem - $\{0, 0, 0, 1, 0, 0\}$)
- Pe următoarea linie se află un șir de caractere diferite reprezentând alfabetul automatului (de exemplu *qwertyuiopasdfghjklzxcvbnm*)
- Pe următoarea linie se află mulțimea de stări finale ale automatului, separate printr-un spațiu
- Pe următoarea linie se află un număr natural k
- Pe următoarele k linii se află câte o relație de adiacență între stările din automat reprezentate printr-un simbol α și două numere naturale $\leq n$ astfel încât să se respecte condițiile de AFD (de exemplu “ $a \ 0 \ 3$ ” reprezintă o muchie $q_0 \xrightarrow{a} q_3$)
- Pe următoarea linie a fișierului se află un număr natural m
- Pe următoarele m linii se află câte un cuvânt format din caractere ale alfabetului

Output (afd.out)

- m linii, linia i răspunde la întrebarea: “este cuvântul de la poziția i acceptat de automatul dat?” (0 - Nu, 1 - Da).

2 Simulare AFN

Dându-se un *automat finit nedeterminist* și m cuvinte formate din alfabetul acestuia, testați acceptanța automatului la fiecare din ele.

Input (afn.in)

- Pe prima linie a fișierului se găsește un număr n - indicele maxim al unei stări din automat ($\{q_0, q_1, \dots, q_n\}$, $n + 1$ stări)
- Pe următoarea linie se află un vector de caracteristică a stărilor finale (de exemplu, pentru un automat cu $n = 5$ și q_3 singura stare finală avem - $\{0, 0, 0, 1, 0, 0\}$)
- Pe următoarea linie se află un șir de caractere diferite reprezentând alfabetul automatului (de exemplu *qwertyuiopasdfghjklzxcvbnm*)
- Pe următoarea linie se află mulțimea de stări finale ale automatului, separate printr-un spațiu
- Pe următoarea linie se află un număr natural k
- Pe următoarele k linii se află câte o relație de adiacență între stările din automat reprezentate printr-un simbol α și două numere naturale $\leq n$ (de exemplu “ $a \ 1 \ 3$ ” reprezintă o muchie $q_1 \xrightarrow{a} q_3$)
- Pe următoarea linie a fișierului se află un număr natural m
- Pe următoarele m linii se află câte un cuvânt format din caractere ale alfabetului

Output (afn.out)

- m linii, linia i răspunde la întrebarea: “este cuvântul de la poziția i acceptat de automatul dat?” (0 - Nu, 1 - Da).

3 Transformare AFN în AFD

Dându-se un AFN și m cuvinte ca la proiectul anterior, aplicați transformarea acestuia în AFD, și testați acceptanța celor m cuvinte de către AFD-ul obținut.

Input (transform.in)

- Același format de input ca la proiectul 2

Output (transform.out)

- Pe prima linie se află un număr natural n' - indicele maxim al unei stări în AFD-ul obținut
- Pe următoarea linie se află un număr natural k' - numărul de relații de adiacență din AFD-ul obținut
- Pe următoarea linie se află mulțimea de stări finale ale automatului determinat, separate printr-un spațiu
- Pe următoarele k' linii se află câte o relație de adiacență între stările din AFD-ul obținut
- Pe următoarele m linii se află, pentru fiecare $1 \leq i \leq m$, răspunsul la întrebarea “este cuvântul de la poziția i acceptat de AFD-ul obținut?” (0 - Nu, 1 - Da).

Observații

- Se consideră prin convenție că starea inițială este mereu q_0 pentru toate proiectele.
- Dacă abordați ambele proiecte 1 și 2, diferențiați soluțiile între ele; deși algoritmul pentru AFN este suficient de general cât să funcționeze și la AFD, de exemplu pentru AFD puteți folosiți ideea de a optimiza respingerea cuvintelor pe baza faptului că mereu putem ajunge în maxim o singură altă stare dintr-o stare precedentă.
- Pentru proiectul 3 va trebui aplicată metoda de acceptanță/respingere **pe AFD-ul obținut și nu pe AFN-ul inițial**.