The Separating Words Problem

Dorneanu Diana-Delia

April 4, 2025

1 Context

În informatica teoretică, problema separării cuvintelor presupune găsirea celui mai mic automat finit determinist care se comportă diferit pe două șiruri date, adică acceptă unul dintre cele două șiruri și respinge celălalt șir.

1.1 Exemplu

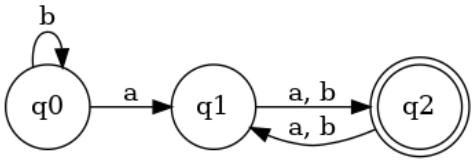


Figure 1: Acest DFA separă șirul aaba de baba.

2 Cazuri speciale

2.1 Cele două cuvinte au lungimi diferite

Propozitie: Dacă două cuvinte au lungimi diferite, ambele $\leq n$, le putem separa cu un DFA de lungime O(log n).

Demonstrație:

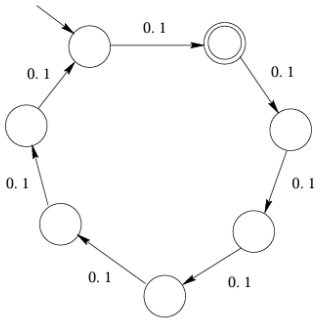
Vom folosi următoarea ***LEMA***:

Dacă $0 \leq i, j < n$ *și* $i = j$*, atunci există un număr prim* $p \leq \log n$ *astfel încât* $i \equiv j \pmod{p}$.

Considerăm două cuvinte w_1 și w_2 cu $|w_1| < |w_2| \leq n$. Putem accepta un cuvânt și respinge celălalt folosind un ciclu modulo p, unde p este numărul prim din lemă și clasa de rest corespunzătoare.

Un exemplu sugestiv:

Să presupunem că $|w_1| = 22$ și $|w_2| = 52$. Atunci $w_1 \equiv 1 \pmod 7$ și $w_2 \equiv 3 \pmod 7$. Putem accepta w_1 și respinge w_2 folosind un DFA care utilizează un ciclu de dimensiune 7, după cum urmează:

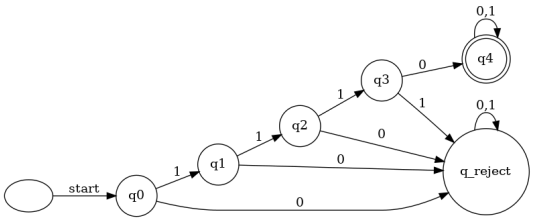


2.2 Cele două cuvinte au aceeași lungime

2.2.1 Cuvintele au prefix/sufix comun până la un caracter

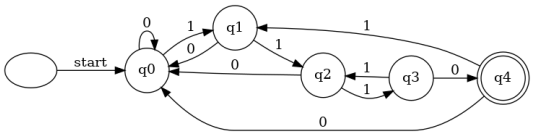
Propoziție: Considerăm două cuvinte w_1 și w_2 care au un prefix comun, până pe o poziție d, de unde încep să difere. Atunci, lungimea minima a DFA-ului este $\leq d + 2$.

De exemplu, putem separa 11010011 de 11100010, construind un DFA care accepta cuvinetele ce incep cu 1110:



Propoziție: Considerăm două cuvinte w_1 și w_2 care au un sufix comun, până pe o poziție d, de unde încep să difere. Atunci, lungimea minima a DFA-ului este $\leq d + 1$.

Analog, construim un DFA care accepta cuvintele ce se termina in 1110, separând astfel cuvintele 01001110, de 10101010.



2.2.2 Cuvintele au un simbol ce apare de un număr diferit de ori

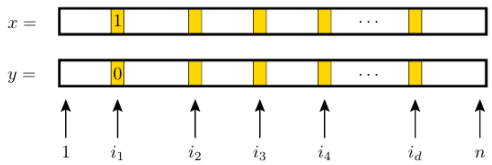
Folosind LEMA din cazul *2.1*, dacă $|w| = |x| = n$ și $|w|_a = |x|_a$ pentru un simbol a, atunci lungimea minimă a unui DFA este $O(\log n)$.

2.2.3 Cuvintele au o distanță Hamming mică

Distanța Hamming $H(w, x)$ dintre două cuvinte de lungime egală w și x este definită ca fiind numărul de poziții în care ele diferă.

Propoziție: Fie w_1 și w_2 două cuvinte de lungime n. Dacă $H(w, x) \leq$ d, atunci lungimea minima a unui DFA este O(d logn).

Demonstrație: Considerăm două cuvinte w și x formate din 0 și 1, iar w are un 1 pe o poziție diferită față de x.



Fie $i_1, i_2, \ldots, i_d$ pozițiile în care x și y diferă.

Considerăm acum $N = (i_2 - i_1)(i_3 - i_1) \cdots (i_d - i_1)$. Atunci $N < n^{d-1}$.

Deci N nu este divizibil cu un număr prim $p = O(\log N) = O(d \log n)$.

Deci $i_j \not\equiv i_1 \pmod{p}$ pentru $2 \leq j \leq d$.

Numărăm acum, modulo 2, câti de 1 apar în poziții congruente cu $i_1 \pmod{p}$.

Aceste poziții nu includ niciuna dintre $i_2, i_3, \ldots, i_d$, deoarece am ales p astfel, iar cele două cuvinte sunt identice în toate celelalte poziții.

Deci x conține exact un 1 în aceste poziții, spre deosebire de w, și astfel putem separa cele două cuvinte folosind $O(d \log n)$ stări.

3 Estimări găsite până acum

Problema estimării dimensiunii unui automat care poate distinge între două șiruri date a fost formulată pentru prima dată de Goralčík și Koubek (1986), care au arătat că dimensiunea automatului este întotdeauna subliniară.

Ulterior, Robson (1989) a demonstrat o limită superioară de $O(n^{2/5}(\log n)^{3/5})$ pentru dimensiunea necesară a automatului.

Această limită a fost îmbunătățită de Chase (2020), care a obținut o limită superioară de

$$\boldsymbol{O(n^{1/3}(\log n)^7)}$$

3.1 Schiță de demonstrație a estimării lui Chase

Scopul este să construim un DFA de dimensiune mică care acceptă un cuvânt x și respinge un alt cuvânt y, cu $|x| = |y| = n$ și $x \neq y$.

Fie $i_1, i_2, \ldots, i_d$ pozițiile în care x și y diferă. Construim produsul:

$$N = (i_2 - i_1)(i_3 - i_1) \cdots (i_d - i_1),$$

care satisface $N < n^{d-1}$.

Conform teoriei numerelor, există un număr prim $p = O(d \log n)$ care nu divide N. Astfel,

$$i_j \not\equiv i_1 \pmod{p} \quad \text{pentru } 2 \leq j \leq d.$$

Considerăm acum pozițiile din x și y care sunt congruente cu $i_1 \mod p$. Numărăm, modulo 2, câți de 1 apar în aceste poziții. Deoarece doar i_1 este congruent cu sine modulo p, iar celelalte poziții de diferență sunt excluse, diferența între x și y se reflectă clar în această numărătoare.

Astfel, putem construi un DFA care ține evidența (modulo 2) a numărului de apariții ale lui 1 în pozițiile congruente cu $i_1 \mod p$. Automatul va accepta un cuvânt și va respinge celălalt, utilizând cel mult $O(d \log n)$ stări.

Optimizând alegerea lui d, se obține o limită superioară generală de:

$$O(n^{1/3} \log^7 n)$$

pentru separarea oricăror două șiruri binare distincte de lungime n.

4 Idei personale

Când am început să mă documentez despre această problemă, primul gând a fost să construiesc un DFA care accepta primul cuvânt. Automat, al doilea este respins, dar lungimea acestui automat ar fi lungimea primului cuvânt.

Apoi, m-am gândit la un caz particular "Ce s-ar întâmpla dacă cele doua string-uri ar fi identice până într-un punct?". Puteam să formez un DFA care conține stările comune, iar în starea finală se putea ajunge doar prin următorul caracter din primul cuvânt. Totuși, acesta nu respecta generalitatea. Mă bucur că în timp ce scriam acest document, am găsit soluție și pentru ipoteza mea.

References

[1] Remarks on Separating Words - MIT

[2] Chase's paper