

Haladó informatikai algoritmusok

Mintaillesztés

Véges determinisztikus automata használata

Fogalmak és jelölések

- Legyen Σ véges jelkészlet, ábécé $\Rightarrow$ **betűk**
- Jelölje Σ^* a Σ elemeiből képezhető összes véges jelsorozat halmazát, beleértve az üres sortozatot is, amit λ -val jelölünk.
- $\Sigma^* = \{x_1 \dots x_n : x_i \in \Sigma, i = 1, \dots, n\} \cup \{\lambda\} \Rightarrow$ **szavak**
- $X \in \Sigma^*$ szó hossza az $X = x_1 \dots x_n$ sorozat elemeinek a száma, jele $|X| = n$.
- A továbbiakban feltételezzük, hogy a szavakat tömb ábrázolja, tehát az X szó i -edik elemére az $X[i]$ tömbelem-kiválasztással hivatkozunk.

Fogalmak és jelölések

- $x = x_1 \dots x_m$, $y = y_1 \dots y_n \in \Sigma^*$ szavak **konkatenációja**:
 $xy = x_1 \dots x_m \cdot y_1 \dots y_n$
- Az u szó kezdő szelete, vagy **prefixe** v -nek, jele $u \sqsubset v$, ha $\exists w \in \Sigma^*$, hogy $uw = v$
- Az u szó végződése, vagy **szuffixe** v -nek, jele $u \sqsupset v$, ha $\exists w \in \Sigma^*$, hogy $wu = v$
- Az $X \in \Sigma^*$ szó első i betűjéből álló prefixére az $X_i = X[1..i]$ jelölést használjuk.
- Általában, egy $X = x_1 \dots x_n \in \Sigma^*$ szó és $1 \leq i \leq j \leq |X|$ esetén $X_{i,j} = X[i..j] = x_i \dots x_j$ jelölést használjuk.
- Azt mondjuk, hogy az S szó **előfordul** i **eltolással** az A szóban S i eltolással **illeszkedik** az A -ra), ha $A[i + 1..i + m] = S$, ahol $m = |S|$.

Mintaillesztési probléma

Bemenet: $A, S \subseteq \Sigma^*$, A a szöveg, S minta

Kimenet: A legkisebb (összes) olyan i , amelyre $A[i + 1..i + m] = S$, ahol $m = |S|$ a minta hossza

Naiv algoritmus:

1. $n := |A|;$
2. $m := |S|;$
3. **for** $i := 0$ **to** $n - m$ **do**
4. **if** $A[i + 1..i + m] = S[1..m]$ **then begin**
5. $WriteLn(i + 1);$ **Exit**
6. **End;**

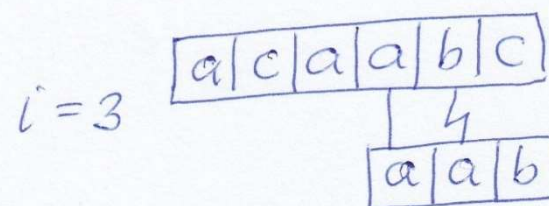
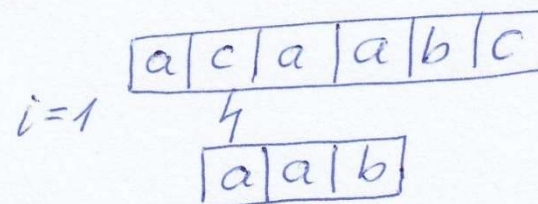
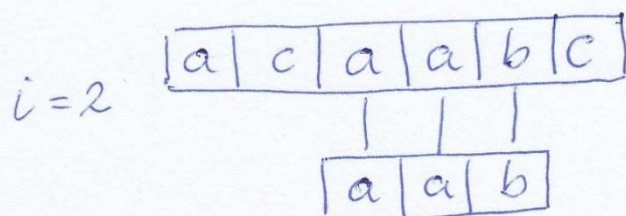
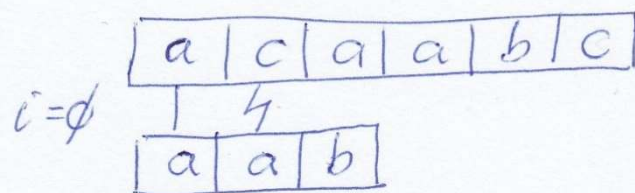
A naiv algoritmus futási ideje: $T_{lr}(n, m) = O(n \cdot m)$

Naiv algoritmus működése

Naiv algoritmus:

$A = acaabc$ $n = 6$
 $S = aab$ $m = 3$

$i = 0, 1, 2, \overset{n-m}{\textcircled{3}}$



Mintaillesztés véges determinisztikus automatával

Véges determinisztikus automata olyan $M = (Q, q_0, F, \Sigma, \delta)$ rendezett ötös, ahol

- Q véges halmaz, az **állapotok halmaza**,
- $q_0 \in Q$ a **kezdőállapot**,
- $F \subseteq Q$ a **végállapotok** (elfogadó állapotok) halmaza,
- Σ véges halmaz, a **bemeneti jelek** halmaza,
- $\delta : Q \times \Sigma \rightarrow Q$, az automata **átmenetfüggvénye**

Jelölje $\delta^* : Q \times \Sigma^* \rightarrow Q$ a δ átmentfüggvény kiterjesztését szavakra:

$$\delta^*(q, w) = \begin{cases} q, & \text{ha } w = \lambda, \\ \delta(\delta^*(q, x_1 \dots x_{n-1}), x_n), & \text{ha } w = x_1 \dots x_n \end{cases}$$

Egy egyszerű 2 állapotú véges automata:

Állapotok halmaza: $Q = \{0, 1\}$

Kezdő állapot: $q_0 = 0$

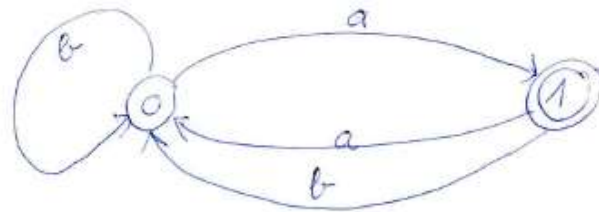
Bemeneti jelek halmaza: $\Sigma = \{a, b\}$

Átmenet függvény:

δ	a	b
0	1	0
1	0	0

Végállapotok halmaza: $F = \{1\}$

Ekvivalens állapot-átmenet diagram:



Ez az automata azokat a sorozatokat fogadja el, amelyek végén páratlan számú „a” szerepel.

pl: 1. bemenet: abaaa

állapotok: $\langle 0, 1, 0, 1, 0, 1 \rangle$

↑
végállapot

2. bemenet: babaabaa

állapotok: $\langle 0, 0, 1, 0, 1, 0, 0, 1, 0 \rangle$

↑
nem végállapot

Automata működése - algoritmus

Legyen $M = (Q, q_0, F, \Sigma, \delta)$ olyan véges determinisztikus automata, amely a $\Sigma^* S$ nyelvet ismeri fel, azaz $L(M) = \Sigma^* S$. Mivel $\Sigma^* S$ az összes olyan szavak halmaza, amelyek az S mintára végződnek, ezért $A_i \in L(M)$ akkor és csak akkor, ha az S minta $i - m$ eltolással előfordul az A szövegben. $\Rightarrow$

Ha van olyan $M = (Q, q_0, F, \Sigma, \delta)$ véges determinisztikus automatánk, amely a $\Sigma^* S$ nyelvet ismeri fel, azaz $L(M) = \Sigma^* S$, akkor az algoritmus mintaillesztést valósít meg:

1. $q := 0$;
2. $n := |A|$;
3. $m := |S|$;
4. **for** $i := 1$ **to** n **do begin**
5. $q := \delta(q, A[i])$;
6. **if** $q \in F$ **then begin** $\{S = A[i - m + 1..i]\}$
7. $WriteLn(i - m + 1)$;
8. Exit;
9. **end**;
10. **end**;

Az algoritmus futási ideje, eltekintve az automata előállítás költségétől $O(n)$.

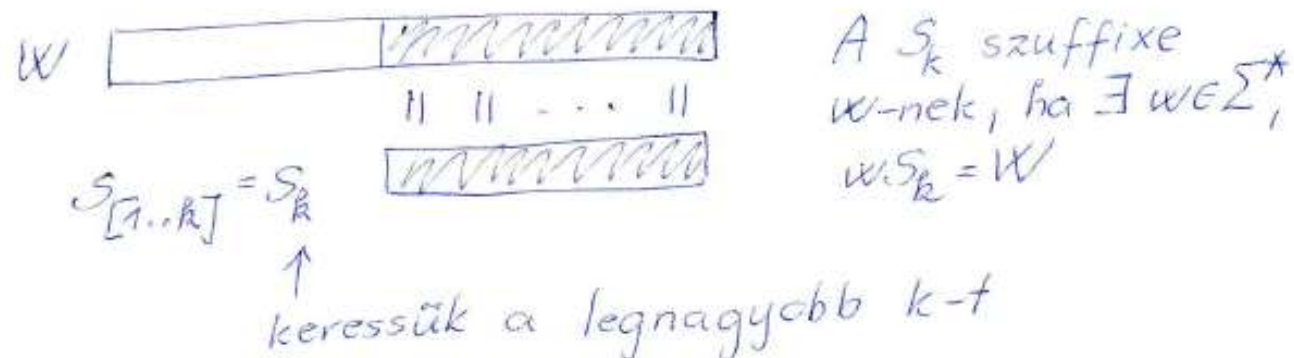
Átmenetfüggvény

Hogyan és mekkora költséggel lehet előállítani az M automatát (átmenetfüggvényét)?

Legyen $\sigma : \Sigma^* \rightarrow \{0, 1, \dots, m\}$ az S minta **suffix-függvénye**

$$\sigma(W) = \max\{k : S_k \sqsupseteq W\}$$

Az S minta akkor és csak akkor illeszkedik az A szövegre i eltolással, ha $\sigma(A_{i+m}) = m$.



pl: $W = abaacbabaa$
 $\underbrace{abaa} c \dots$
 $S_k \Rightarrow k = 4$

Lemma 1

Ha $q = \sigma(W)$, akkor $\sigma(Wx) = \sigma(S_q x)$ minden $W \in \Sigma^*$ és $x \in \Sigma$ -ra.

Bizonyítás:

Legyen $u = \sigma(Wx)$ és $r = \sigma(S_q x) \Rightarrow$
a legnagyobb olyan u , amelyre $S_u \sqsubset Wx \Rightarrow$

$S_{u-1} \sqsubset W \Rightarrow$

σ definíciója miatt $u - 1 \leq q \Rightarrow$

$S_{u-1} \sqsubset S_q \Rightarrow$

$S_u = S_{u-1}x \sqsubset S_q x$

de r a legnagyobb olyan index, hogy $S_r \sqsubset S_q x$, tehát $u \leq r$.

Másrészt $S_r \sqsubset S_q x \sqsubset Wx \Rightarrow r \leq u$.



$u = r$

Mire van szükségünk

Olyan automatát akarunk szerkeszteni, amelynek δ átmenetfüggvénye a σ függvényt számítja:

$$\delta^*(0, W) = \sigma(W)$$

Legyen $Q = \{0, 1, \dots, m\}$, ahol $m = |S|$.

Minden $0 \leq q \leq m$ állapotra és $x \in \Sigma$ jelre:

$$\delta(q, x) = \begin{cases} q + 1, & \text{ha } x = S[q + 1], \\ \max\{k : S_k \sqsupseteq S_q x\}, & \text{ha } x \neq S[q + 1] \end{cases}$$

Nyilvánvaló, hogy $\delta(q, x) = \sigma(S_q x)$.

Példa:

$$S = ababc \quad \Sigma = \{a, b, c\}$$

$$\begin{aligned} \delta(0, a) &= 1 \quad \left\{ \begin{array}{l} a \\ a \dots \end{array} \right. \\ \delta(0, b) &= 0 \quad \left\{ \begin{array}{l} b \\ a \end{array} \right. \neq \\ \delta(0, c) &= 0 \quad \left\{ \begin{array}{l} c \\ a \end{array} \right. \neq \end{aligned}$$

$$\begin{aligned} \delta(1, a) &= 1 \quad \left\{ \begin{array}{l} aa \\ a \dots \end{array} \right. \\ \delta(1, b) &= 2 \quad \left\{ \begin{array}{l} ab \\ ab \dots \end{array} \right. \\ \delta(1, c) &= 0 \quad \left\{ \begin{array}{l} ac \\ -a \dots \end{array} \right. \end{aligned}$$

$$\begin{aligned} \delta(2, a) &= 1 \quad \left\{ \begin{array}{l} aba \\ a \dots \end{array} \right. \\ \delta(2, b) &= 0 \quad \left\{ \begin{array}{l} abb \\ a \dots \end{array} \right. \\ \delta(2, c) &= 0 \quad \left\{ \begin{array}{l} abc \\ a \dots \end{array} \right. \end{aligned}$$

$$\begin{aligned} \delta(3, a) &= 1 \quad \left\{ \begin{array}{l} abaa \\ a \dots \end{array} \right. \\ \delta(3, b) &= 4 \quad \left\{ \begin{array}{l} abab \\ abab \dots \end{array} \right. \\ \delta(3, c) &= 0 \quad \left\{ \begin{array}{l} abac \\ a \dots \end{array} \right. \end{aligned}$$

$$\begin{aligned} \delta(4, a) &= 3 \quad \left\{ \begin{array}{l} ababa \\ ab \dots \end{array} \right. \\ \delta(4, b) &= 0 \quad \left\{ \begin{array}{l} ababb \\ a \dots \end{array} \right. \\ \delta(4, c) &= 5 \quad \left\{ \begin{array}{l} ababc \\ ababc \dots \end{array} \right. \end{aligned}$$

Lemma 2

Minden $W \in \Sigma^*$ szóra $\delta^*(0, W) = \sigma(W)$.

Bizonyítás:

Bizonyítás W hossza szerinti indukcióval.

$W = \lambda$ esetén $\delta^*(0, W) = 0 = \sigma(W)$.

Legyen $W = Vx$, $V \in \Sigma^*$, $x \in \Sigma$ és tegyük fel, hogy $\delta^*(0, V) = \sigma(V) = q$.

$$\begin{aligned}\delta^*(0, Vx) &= \delta(\delta^*(0, V), x) \\ &= \delta(\sigma(V), x) \\ &= \delta(q, x) \\ &= \sigma(S_q x) \\ &= \sigma(Vx)\end{aligned}$$

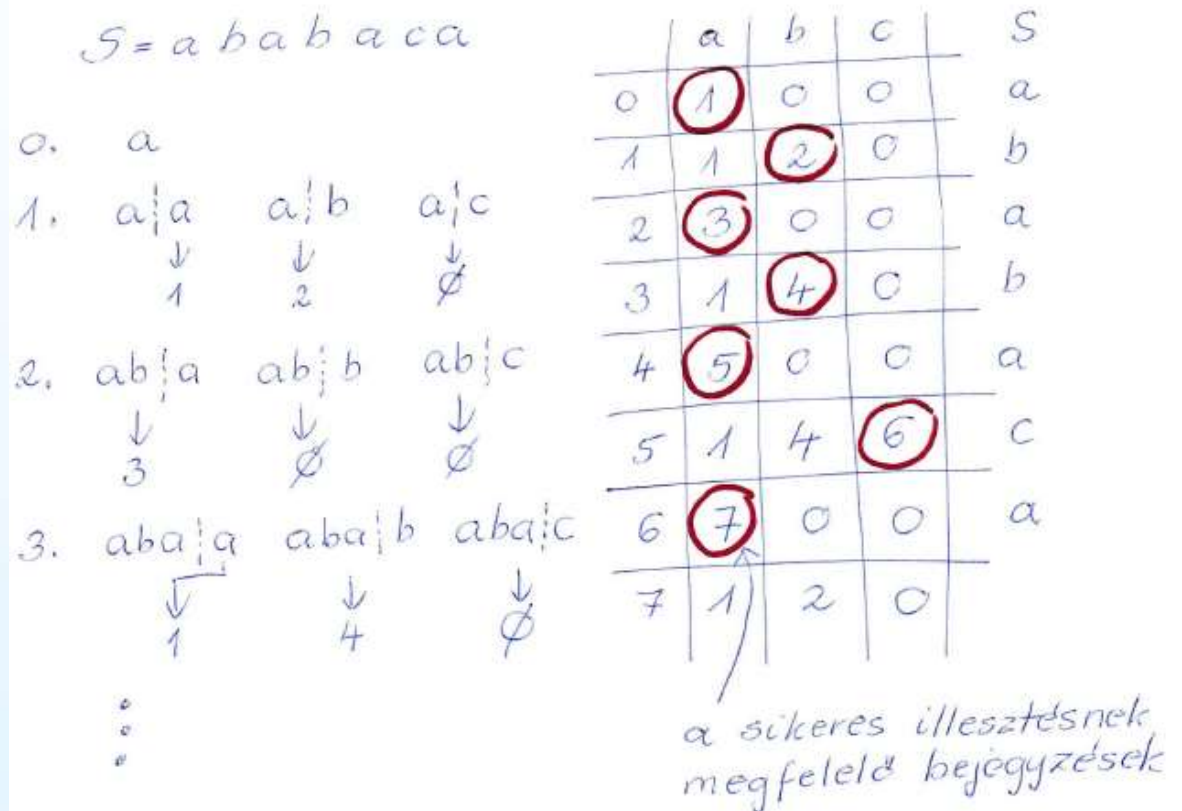
Az átmenetfüggvény kiszámítása

```

1.  $m := |S|$ ;
2. for  $q := 0$  to  $m$  do
3.   for  $x \in \Sigma$  do begin
4.      $k := \min(m + 1, q + 2)$ ;
5.     repeat
6.        $k := k - 1$ ;
7.     until  $S_k \supseteq S_q x$ ;
8.      $\delta(q, x) := k$ ;
9.   end;
10. end;

```

Az algoritmus futási ideje $O(m^3|\Sigma|)$.



Az automata működése:

A szöveg összes $A[i]$ jele alatt feltüntettük az automata $\delta(A_i)$ állapotát, amelybe az A_i prefix feldolgozása után kerül.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A[i]$	a	b	a	b	a	b	a	c	a	b	a	b	a	c	a

$\delta(A_i)$	0	1	2	3	4	5	4	5	6	7	2	3	4	5	6	7
$\uparrow$																

$\max\{k : S_k \supset A_i\}$

Itt minta 2-szer fordul elő a szövegben.

Az átmenetfüggvény kiszámítására szolgáló eljárás működése:

$S = ababaca$

$\Sigma = \{a, b, c\} \quad m = |S| = 7$

1. $q_i = \phi$
 $x := a \quad k := \min(\overset{m+1}{8}, \overset{q+2}{2}) = 2$
 $k := 1$

$S_1 \supset S_0 x ?$
 $a \supset a \checkmark \quad \boxed{\delta(0, a) := 1}$

$x := b \quad k := \min(8, 2) = 2$
 $k := 1$
 $S_1 \supset S_0 b$
 $a \supset b \times \quad S_0 \supset b \checkmark$
 $\boxed{\delta(0, b) := 0}$

$x := c \quad k := 0$
 $\boxed{\delta(0, c) := 0}$

2. $q_i = 1$
 $x := a \quad k := \min(8, 3) = 3$
 $k := 2$

$S_2 \supset S_1 a$
 $ab \supset aa \times$
 $k := 1$
 $S_1 \supset S_1 a \quad \boxed{\delta(1, a) := 1}$
 $a \supset aa$

$x := b \quad k := 3$
 $k := 2$
 $S_2 \supset S_1 b$
 $ab \supset ab \checkmark \quad \boxed{\delta(1, b) := 2}$

$x := c \quad k := 3$
 $k := 2$
 $S_2 \supset S_1 c$
 $ab \supset ac \times$
 $k := 1$
 $S_1 \supset S_1 c$
 $a \supset ac \times \quad \boxed{\delta(1, c) := 0}$

3. $q_i = 2$
 $x := a \quad k := \min(8, 4) = 4$
 $k := 3$

$S_3 \supset S_2 a$
 $aba \supset aba \checkmark \quad \boxed{\delta(2, a) := 3}$

$x := b \quad k := 4$
 $k := 3$

$S_3 \supset S_2 b$
 $aba \supset abb \times$

$k := 2$
 $S_2 \supset S_2 b$
 $ab \supset abb \times$

$k := 1$
 $S_1 \supset S_2 b \quad \boxed{\delta(2, b) := 0}$
 $a \supset abb \times$

$x := c \quad \boxed{\delta(2, c) := 0}$

4. $q_i = 3$
 $x := a \quad k := \min(8, 5) = 5$
 $k := 4$

$S_4 \supset S_3 a$
 $abab \supset abaa \times$

$k := 3$
 $S_3 \supset S_3 a$

$aba \supset abaa \times$

$k := 2$
 $ab \supset abaa \times$

$k := 1$
 $a \supset abaa \checkmark$

$\boxed{\delta(3, a) := 1}$

⋮