

# 1. Ábécék, nyelvek, műveletek (jegyzet, 9-10. oldal)

## 2.7. példa - Nyelv műveletek - Gyakorló feladat

Legyen  $V = \{a, b, c\}$ ,  $L_1 = \{a, c, bb, aba\}$ ,  $L_2 = \{a, abba, baba, caba, abbaba, babaabba\}$ . Adjuk meg az  $L_1 \cup L_2$ ,  $L_1 \cap L_2$ ,  $L_1 L_2$ ,  $L_1 L_1$  halmazokat. ★

Megoldás:

$$L_1 \cup L_2 = \{a, c, bb, aba, abba, baba, caba, abbaba, babaabba\}$$

$$L_1 \cap L_2 = \{a\}$$

$$L_1 \cdot L_2 = \{aa, aabba, ababa, acaba, abbaba, abababba, ca, cabba, cbaba, ccaba, cabbaba, cbbababba, bba, bbaabba, bbbaba, bbaba, bbaabba, bbbababba, abaa, abaaabba, abababa, abacaba, abaaabba, abababba, abababba\}$$

$$L_1 \cdot L_1 = \{aa, ac, asb, aasa, ca, cc, cbb, caba, bba, bbc, bbbb, bbaba, abaa, abac, ababba, ababba\}$$

## 2.8. példa - Nyelvek konkatenációja

Adjunk példát olyan  $L_1$  és  $L_2$   $V$  ábécé feletti nyelvekre, amelyekre  $L_1 L_2 = L_2 L_1$ . Keressünk nem triviális megoldást is.

Triviális megoldások:

- $L_1 = \emptyset, L_1 = \{ \lambda \}$  vagy a szimmetria miatt  $L_2$ -re teljesül az előző esetek egyike.
- $L_1 = L_2$ .
- $V$  ábécé egyelemű.
- Az egyik nyelvben benne szerepel  $\lambda$ , a másik nyelv pedig a  $V^*$  (univerzális nyelv).

Egy nem triviális megoldás:

legyen  $V = \{ a, b \}$ ,  $L_1 = \{ \lambda, a \}$ ,  $L_2$  pedig legyen azon  $V$  feletti szavak halmaza, amelyekben pontosan egy  $b$  szerepel. Ekkor  $L_1 L_2 = L_2 L_1 = L_2$ . ★

## 2.9. példa - Nyelvek számossága - Gyakorló feladat

Adottak  $L_1$  és  $L_2$  véges nyelvek  $V$  ábécé felett, hogy  $|L_1| = n, |L_2| = m$ . Mennyi lehet a számossága az  $L_1 \cup L_2, L_1 \cap L_2, L_1 L_2$  nyelvműveletekkel előálló nyelveknek? Adjunk meg alsó felső korlátot, és példákat. ★

Megoldás:

- $\max(m, n) \leq |L_1 \cup L_2| \leq m + n$   
pl.:  $|\{a, b, c\} \cup \{a, b\}| = 3$   
 $|\{a, b, c\} \cup \{d, e\}| = 5$
- $0 \leq |L_1 \cap L_2| \leq \min(m, n)$   
pl.:  $|\{a, b, c\} \cap \{d\}| = 0$   
 $|\{a, b, c\} \cap \{a, b\}| = 2$



$$\bullet \max(m, n) \leq |L_1 \cdot L_2| \leq m \cdot n$$

$$\text{pl.: } |\{\lambda\} \cdot \{a, b, c\}| = 3$$

$$|\{a, b, c\} \cdot \{d, e\}| = 6$$

De pl.:

$$|\{\lambda, a\} \cdot \{a, aa, aaa\}| =$$

$$= |\{a, aa, aaa, aaaa\}| = 4 = \max(m, n) + 1$$

## 2.10. példa - Formális nyelvek, nyelvműveletek 1.feladat

Igazoljuk vagy cáfoljuk, hogy  $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$  !

Megoldás: Az állítás hamis. Vegyük a következő ellenpéldát: Legyen  $L_1 = \{a\}$  és  $L_2 = \{b\}$ , ekkor  $(L_1 \cup L_2)^*$  az akárhány  $a$ -t és  $b$ -t tartalmazó szavak halmaza, még  $L_1^* \cup L_2^*$  a csupa  $a$ -t és csupa  $b$ -t tartalmazó szavak nyelve lesz. ★

## 2.11. példa - Formális nyelvek, nyelvműveletek 2.feladat

Mivel egyenlő  $L^2$ , ha

$$L = \{a^n b^n \mid n > 0\} ?$$

Megoldás:  $L^2 = \{a^n b^n a^m b^m \mid n, m > 0\}$ . ★

## 2. Nyelvek → reguláris kifejezések (jegyzet, 80. oldal)

### 5.2. példa - Nyelv megadása reguláris kifejezéssel 1. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon szavakból áll, amelyek tartalmazzák részszerként a 010 szót!

Megoldás:  $L=(0+1)^*010(0+1)^*$  ★

### 5.3. példa - Nyelv megadása reguláris kifejezéssel 2. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon szavakból áll, amelyek tartalmazzák részszerként a 000 vagy az 111 szót!

Megoldás:  $L=(0+1)^*(000+111)(0+1)^*$  ★

### 5.4. példa - Nyelv megadása reguláris kifejezéssel 3. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon 1-esre végződő szavakból áll, amelyek nem tartalmazzák részszerként a 00 szót!

Megoldás:  $L=(1+01)^*$  ★

### 5.5. példa - Nyelv megadása reguláris kifejezéssel 4. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon szavakból áll, melynek 3. betűje 0!

Megoldás:  $L=(00+01+10+11)0(0+1)^*$  ★

## 5.6. példa - Nyelv megadása reguláris kifejezéssel 5. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon szavakból áll, melyek tartalmazznak legalább három 1-et!

Megoldás:  $L=(0+1)^*1(0+1)^*1(0+1)^*1(0+1)^*$  ★

## 5.7. példa - Nyelv megadása reguláris kifejezéssel 6. feladat

Adjuk meg reguláris kifejezéssel azt a nyelvet a  $\{0,1\}$  ábécé felett, amely azon szavakból áll, melyek 5-tel osztható 1-et tartalmazznak!

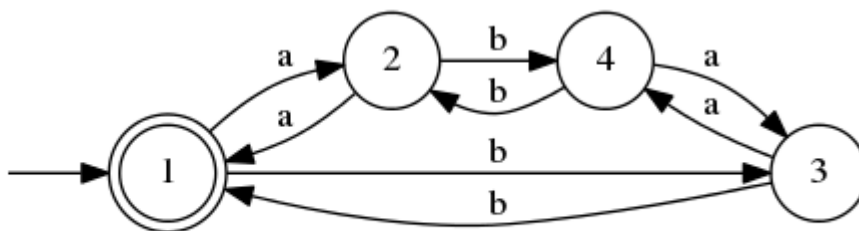
Megoldás:  $L=(0^*10^*10^*10^*10^*)^*$  ★

## 3. Nyelvek → véges automaták

Adjunk olyan determinisztikus véges automatát, ami az alábbi nyelvet fogadja el.

- a) Az összes olyan  $\{a, b\}$  feletti szót, melyben az "a" betűk száma és a "b" betűk száma is páros.

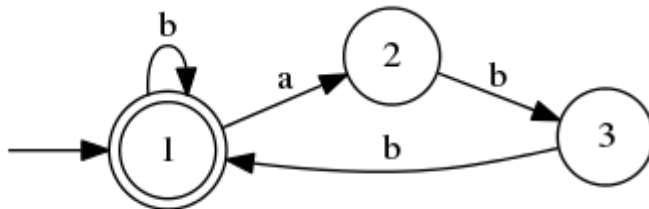
Megoldás:



- Állapot 1: "a" betűk száma páros, "b" betűk száma páros.
- Állapot 2: "a" betűk száma páratlan, "b" betűk száma páros.
- Állapot 3: "a" betűk száma páros, "b" betűk száma páratlan.
- Állapot 4: "a" betűk száma páratlan, "b" betűk száma páratlan.

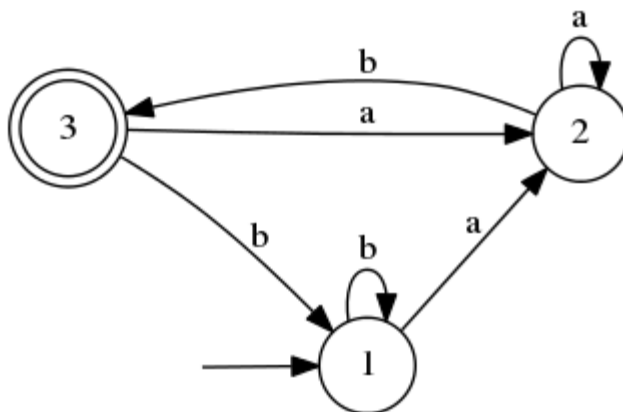
- b) Az összes olyan  $\{a, b\}$  feletti szót, melyben minden "a" betűt (ha van) "bb" követ.

Megoldás:



- b) Az összes olyan  $\{a, b\}$  feletti szót mely "ab"-re végződik.

Megoldás:



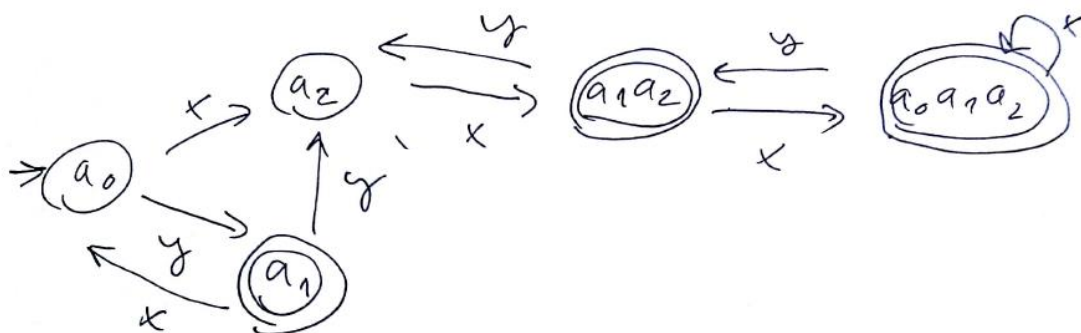
## 4. Végtes automata determinizálása (jegyzet 88. oldal)

### 5.19. példa - Automata determinizálása 1. feladat

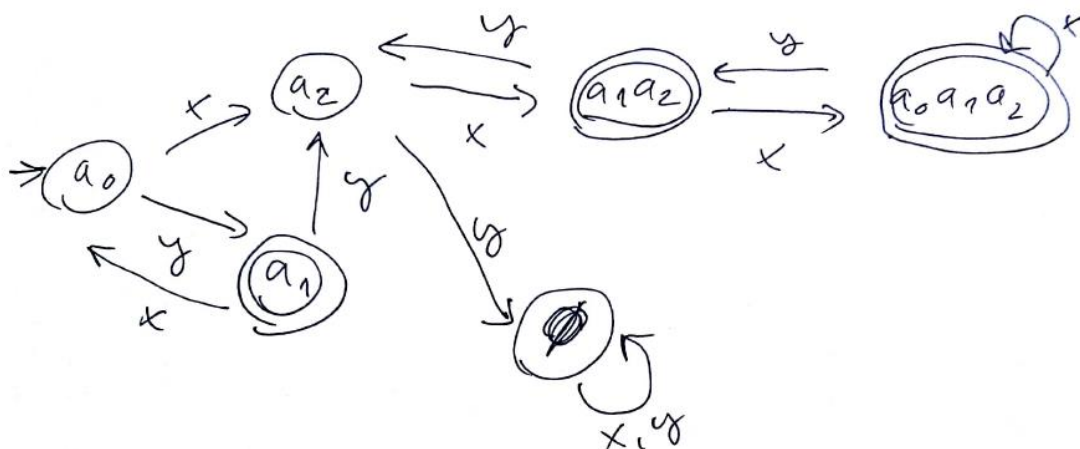
Adjunk meg az  $A = (\{ a_0, a_1, a_2 \}, \{ x, y \}, a_0, \delta, \{ a_1 \})$  nemdeterminisztikus, parciálisan definiált, véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$	$a_2$
$x$	$\{ a_2 \}$	$\{ a_0 \}$	$\{ a_1, a_2 \}$
$y$	$\{ a_1 \}$	$\{ a_2 \}$	-

Megoldás



Illetve ha teljesen specifikált automatát akarunk, akkor:



Ha az állapotokat az alábbiak szerint átnevezzük:

$$b_0 = \{ a_0 \}, b_1 = \{ a_2 \}, b_2 = \{ a_1 \}.$$

$$b_3 = \{ a_1, a_2 \}, b_4 = \emptyset.$$

$$b_5 = \{ a_0, a_1, a_2 \}.$$

akkor az állapot-átmenet reláció táblázatban:

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$
$x$	$b_1$	$b_3$	$b_0$	$b_5$	$b_4$	$b_5$
$y$	$b_2$	$b_4$	$b_1$	$b_1$	$b_4$	$b_3$

A véges automata tehát:

- Az  $A_d$  automata bemenő jeleinek halmaza megegyezik az  $A$  automata bemenő jeleinek a halmazával.
- Az  $A_d$  automata kezdőállapota az  $A$  automata kezdőállapotához rendelt  $b_i$  lesz.
- Az  $A_d$  automata végállapotainak a halmaza pedig tartalmazni fog minden olyan  $b_j$  állapotot, melynek mint halmaznak eleme az  $A$  automata bármely végállapota.

Jelen esetben:

$$A_d = (\{ b_0, b_1, b_2, b_3, b_4, b_5 \}, \{ x, y \}, b_0, \delta', \{ b_2, b_3, b_5 \}).$$



5.20. példa - Automata determinizálása 2. feladat

Adjunk meg az  $A=(\{ a_0, a_1 \}, \{ x, y \}, a_0, \delta, \{ a_1 \})$   
nemdeterminisztikus, parciálisan definiált,  
véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$
$x$	$\{ a_1 \}$	$\{ a_0, a_1 \}$
$y$	$\{ a_0 \}$	-

Megoldás:

I.  $b_0=\{ a_0 \}, b_1=\{ a_1 \}, b_2=\{ a_0, a_1 \}, b_3 = \varnothing.$

II.

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_2$	$b_2$	$b_3$
$y$	$b_0$	$b_3$	$b_0$	$b_3$

III. $A_d=(\{ b_0, b_1, b_2, b_3 \}, \{ x, y \}, b_0, \delta', \{ b_1, b_2 \}).$



5.21. példa - Automata determinizálása 3. feladat

Adjunk meg az  $A=(\{ a_0, a_1 \}, \{ x, y, z \}, a_0, \delta, \{ a_0 \})$   
nemdeterminisztikus, parciálisan definiált,  
véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$
$x$	$\{ a_1 \}$	-
$y$	-	$\{ a_0, a_1 \}$
$z$	$\{ a_0, a_1 \}$	$\{ a_0 \}$

Megoldás:

I.  $b_0=\{ a_0 \}, b_1=\{ a_1 \}, b_2 = \varnothing, b_3=\{ a_0, a_1 \}.$

II.

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_2$	$b_2$	$b_1$
$y$	$b_2$	$b_3$	$b_2$	
$z$	$b_3$	$b_0$	$b_2$	

III. $A_d=(\{ b_0, b_1, b_2, b_3 \}, \{ x, y, z \}, b_0, \delta', \{ b_0, b_3 \}).$

## 5.22. példa - Automata determinizálása 4. feladat

Adjunk meg az  $A = (\{ a_0, a_1 \}, \{ x, y \}, a_0, \delta, \{ a_1 \})$

nemdeterminisztikus, parciálisan definiált, kimenő jel nélküli, iniciális, végállapotokkal bővített véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$
$x$	$\{ a_0, a_1 \}$	$\{ a_0, a_1 \}$
$y$	$\{ a_0, a_1 \}$	-

Megoldás:

I.  $b_0 = \{ a_0 \}, b_1 = \{ a_0, a_1 \}$ .

II.

$\delta'$	$b_0$	$b_1$
$x$	$b_0$	$b_1$
$y$	$b_1$	$b_1$

III.  $A_d = (\{ b_0, b_1 \}, \{ x, y \}, b_0, \delta', \{ b_1 \})$ .

★

## 5.23. példa - Automata determinizálása 5. feladat

Adjunk meg az  $A = (\{ a_0, a_1, a_2 \}, \{ x, y \}, a_0, \delta, \{ a_0, a_1 \})$

nemdeterminisztikus, parciálisan definiált,

véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$	$a_2$
$x$	$\{ a_2 \}$	$\{ a_1 \}$	$\{ a_1, a_2 \}$
$y$	$\{ a_0 \}$	$\{ a_2 \}$	$\{ a_0 \}$

Megoldás:

I.  $b_0 = \{ a_0 \}, b_1 = \{ a_2 \}, b_2 = \{ a_1, a_2 \}, b_3 = \{ a_0, a_2 \}$ .

II.

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_2$	$b_2$	$b_2$
$y$	$b_0$	$b_0$	$b_3$	$b_0$

III.  $A_d = (\{ b_0, b_1, b_2, b_3 \}, \{ x, y \}, b_0, \delta', \{ b_0, b_2, b_3 \})$ .

## 5.24. példa - Automata determinizálása 6. feladat

Adjunk meg az  $A=(\{ a_0, a_1, a_2 \}, \{ x, y \}, a_0, \delta, \{ a_1 \})$

nemdeterminisztikus, parciálisan definiált,

véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$	$a_2$
$x$	$\{ a_1 \}$	$\{ a_1, a_2 \}$	$\{ a_2 \}$
$y$	-	-	$\{ a_1 \}$

Megoldás:

I.  $b_0=\{ a_0 \}, b_1=\{ a_1 \}, b_2 = \emptyset, b_3=\{ a_1, a_2 \}.$

II.

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_3$	$b_2$	$b_3$
$y$	$b_2$	$b_2$	$b_2$	$b_1$

III.  $A_d=(\{ b_0, b_1, b_2, b_3 \}, \{ x, y \}, b_0, \delta', \{ b_1, b_3 \}).$

★

## 5.25. példa - Automata determinizálása 7. feladat

Adjunk meg az  $A=(\{ a_0, a_1, a_2 \}, \{ x, y, z \}, a_0, \delta, \{ a_1, a_2 \})$

nemdeterminisztikus, parciálisan definiált,

véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$	$a_2$
$x$	$\{ a_0 \}$	-	$\{ a_0, a_2 \}$
$y$	$\{ a_1, a_2 \}$	$\{ a_0 \}$	-
$z$	-	$\{ a_0, a_1 \}$	-

Megoldás:

I.  $b_0=\{ a_0 \}, b_1=\{ a_1, a_2 \}, b_2 = \emptyset, b_3=\{ a_0, a_2 \}, b_4=\{ a_0, a_1 \}, b_5=\{ a_0, a_1, a_2 \}.$

II.

$\delta'$	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$
$x$	$b_0$	$b_3$	$b_2$	$b_3$	$b_0$	$b_3$
$y$	$b_1$	$b_0$	$b_2$	$b_1$	$b_5$	$b_5$
$z$	$b_2$	$b_4$	$b_2$	$b_2$	$b_4$	$b_4$

III.  $A_d=(\{ b_0, b_1, b_2, b_3, b_4, b_5 \}, \{ x, y, z \}, b_0, \delta', \{ b_1, b_3, b_4, b_5 \}).$

## 5.26. példa - Automata determinizálása 8. feladat

Adjunk meg az  $A = (\{ a_0, a_1, a_2, a_3 \}, \{ x, y, z \}, a_0, \delta, \{ a_1, a_3 \})$  nemdeterminisztikus, parciálisan definiált, kimenő jel nélküli, iniciális, végállapotokkal bővített véges automatával ekvivalens  $A_d$  determinisztikus, teljesen definiált automatát!

$\delta$	$a_0$	$a_1$	$a_2$	$a_3$
$x$	$\{ a_0 \}$	-	$\{ a_2, a_3 \}$	$\{ a_1, a_2 \}$
$y$	$\{ a_1, a_2, a_3 \}$	$\{ a_0 \}$	-	$\{ a_0 \}$
$z$	-	$\{ a_1, a_2 \}$	$\{ a_1, a_3 \}$	$\{ a_1 \}$

Megoldás:

I.  $b_0 = \{ a_0 \}, b_1 = \{ a_1, a_2, a_3 \}, b_2 = \emptyset$ .

II.

$\delta'$	$b_0$	$b_1$	$b_2$
$x$	$b_0$	$b_1$	$b_2$
$y$	$b_1$	$b_0$	$b_2$
$z$	$b_2$	$b_1$	$b_2$

III.  $A_d = (\{ b_0, b_1, b_2 \}, \{ x, y, z \}, b_0, \delta', \{ b_1 \})$ .

## 5. Determinisztikus véges automata minimalizálása (jegyzet 92. oldal)

### 5.27. példa - Véges elfogadó automata minimalizálása 1. feladat

Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$A = (\{ a_0, a_1, a_2, a_3, a_4, a_5, a_6 \}, \{ x, y \}, a_0, \delta, \{ a_2, a_4, a_5, a_6 \})$

véges automatával ekvivalens  $A_0$  minimális állapotszámú automatát!

$\delta$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$x$	$a_2$	$a_5$	$a_1$	$a_1$	$a_2$	$a_1$	$a_0$
$y$	$a_1$	$a_0$	$a_3$	$a_4$	$a_5$	$a_3$	$a_2$



Megoldás:

(O.) Mielőtt az érdemi munkához hozzáfognánk, meg kell vizsgálni, hogy mely állapotok érhetőek el az  $A$  automata kezdőállapotából. Azokat az állapotokat, melyek nem érhetőek el, egyszerűen töröljük, mivel nem fognak előfordulni semelyik számítás során sem.

Jelen esetben az  $a_0$  állapotból elérhető állapotok:

$$\{a_0, a_2, a_1, a_3, a_5, a_4\}.$$

Látható, hogy semmilyen input szó esetén sem kerülhet az  $A$  automata  $a_6$  állapotba, ezért ezt az állapotot töröljük. Az így kapott

$A'$  automatát kell a továbbiakban minimalizálnunk az

Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével:

$$A' = (\{a_0, a_1, a_2, a_3, a_4, a_5\}, \{x, y\}, a_0, \delta', \{a_2, a_4, a_5\}).$$

$\delta'$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$x$	$a_2$	$a_5$	$a_1$	$a_1$	$a_2$	$a_1$
$y$	$a_1$	$a_0$	$a_3$	$a_4$	$a_5$	$a_3$

(I.) A feladat megoldásának első lépéseként különböző osztályokba fogjuk sorolni az  $A'$  automata belső állapotait. A kezdeti osztályozáskor mindig két osztályba kerülnek az állapotok, attól függően, hogy végállapotok, vagy pedig nem végállapotok.

Jelen esetben:

$$C_1 = \{a_0, a_1, a_3\}, \{a_2, a_4, a_5\}.$$

Ezek után a  $C_{i+1}$ -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a  $C_i$ -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos  $C_i$ -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben  $C_i = C_{i+1}$  valamely  $i \geq 1$  esetén.

Jelen esetben:

$$C_2 = \{a_0, a_1\}, \{a_3\}, \{a_2, a_5\}, \{a_4\}.$$

$$C_3 = \{a_0, a_1\}, \{a_3\}, \{a_2, a_5\}, \{a_4\}.$$

Mivel  $C_2 = C_3$ , ezért az osztályozás véget ért. A  $C_2$  osztályait jelöljük valamely új betűvel, például  $b$ -vel:

$$b_0 = \{a_0, a_1\}, b_1 = \{a_3\}, b_2 = \{a_2, a_5\}, b_3 = \{a_4\}.$$

(II.) Készítsük el az  $A'$  automatával ekvivalens, minimális állapotszámú

$A_0$  automatát, mely állapothalmazát az osztályozás és az új jelölés

bevezetése után kapott  $b_i$  betűk alkotják, a bemenő jeleinek halmaza megegyezik

az  $A$  automata bemenő jeleinek halmazával, az átmenetfüggvényét megkapjuk úgy, hogy megnézzük, hogy az adott  $b_i$  osztálybeli állapotok az adott bemenő jel hatására mely  $b_j$  osztálybeli állapotokba mentek át az  $A'$  automata esetén, az új kezdőállapot az a  $b_i$  lesz, melynek eleme  $a_0$ , és végül az  $A_0$  automata végállapotait azon  $b_{k_1}, \dots, b_{k_m}$  osztályok alkotják, mely osztályok elemei az  $A'$  automata végállapotaiból állnak.

Jelen esetben:

$$A_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, b_0, \delta_0, \{b_2, b_3\}).$$

$\delta_0$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_2$	$b_0$	$b_0$	$b_2$
$y$	$b_0$	$b_3$	$b_1$	$b_2$

★

## 5.28. példa - Véges elfogadó automata minimalizálása 2. feladat

Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$A = (\{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}, \{x, y\}, a_0, \delta, \{a_0, a_1, a_2, a_3\})$$

véges automatával ekvivalens  $A_0$  minimális állapotszámú automatát!

$\delta$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$x$	$a_6$	$a_6$	$a_6$	$a_6$	$a_5$	$a_0$	$a_2$
$y$	$a_6$	$a_3$	$a_0$	$a_3$	$a_1$	$a_4$	$a_1$

Megoldás:

(O.) Az  $a_0$  kezdőállapotból elérhető belső állapotok:  
 $\{a_0, a_6, a_2, a_1, a_3\}$ .

$$A' = (\{a_0, a_1, a_2, a_3, a_6\}, \{x, y\}, a_0, \delta', \{a_0, a_1, a_2, a_3\}).$$

$\delta'$	$a_0$	$a_1$	$a_2$	$a_3$	$a_6$
$x$	$a_6$	$a_6$	$a_6$	$a_6$	$a_2$
$y$	$a_6$	$a_3$	$a_0$	$a_3$	$a_1$

(I.)

$$C_1 = \{a_0, a_1, a_2, a_3\}, \{a_6\}.$$

$$C_2 = \{a_0\}, \{a_1, a_2, a_3\}, \{a_6\}.$$

$$C_3 = \{a_0\}, \{a_1, a_3\}, \{a_2\}, \{a_6\}.$$

$$C_4 = \{a_0\}, \{a_1, a_3\}, \{a_2\}, \{a_6\}.$$

$$b_0 = \{a_0\}, b_1 = \{a_1, a_3\}, b_2 = \{a_2\}, b_3 = \{a_6\}.$$

(II.)

$$A_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, b_0, \delta_0, \{b_0, b_1, b_2\}).$$

$\delta_0$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_3$	$b_3$	$b_3$	$b_2$

$\delta_0$	$b_0$	$b_1$	$b_2$	$b_3$
$y$	$b_3$	$b_1$	$b_0$	$b_1$



### 5.29. példa - Végés elfogadó automata minimalizálása 3. feladat

Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$A = (\{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}, \{x, y\}, a_0, \delta, \{a_2, a_7\})$$

végés automatával ekvivalens  $A_0$  minimális állapotszámú automatát!

$\delta$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
$x$	$a_3$	$a_5$	$a_7$	$a_0$	$a_2$	$a_0$	$a_4$	$a_2$	$a_5$
$y$	$a_8$	$a_1$	$a_3$	$a_2$	$a_6$	$a_7$	$a_3$	$a_5$	$a_2$

Megoldás:

(O.) Az  $a_0$  kezdőállapotból elérhető belső állapotok:

$$\{a_0, a_3, a_8, a_2, a_5, a_7\}.$$

$$A' = (\{a_0, a_2, a_3, a_5, a_7, a_8\}, \{x, y\}, a_0, \delta', \{a_2, a_7, a_8\}).$$

$\delta'$	$a_0$	$a_2$	$a_3$	$a_5$	$a_7$	$a_8$
$x$	$a_3$	$a_7$	$a_0$	$a_0$	$a_2$	$a_5$
$y$	$a_8$	$a_3$	$a_2$	$a_7$	$a_5$	$a_2$

(I.)

$$C_1 = \{a_0, a_3, a_5, a_8\}, \{a_2, a_7\}.$$

$$C_2 = \{a_0\}, \{a_3, a_5, a_8\}, \{a_2, a_7\}.$$

$$C_3 = \{a_0\}, \{a_3, a_5\}, \{a_8\}, \{a_2, a_7\}.$$

$$C_4 = \{a_0\}, \{a_3, a_5\}, \{a_8\}, \{a_2, a_7\}.$$

$$b_0 = \{a_0\}, b_1 = \{a_3, a_5\}, b_2 = \{a_8\}, b_3 = \{a_2, a_7\}.$$

(II.)

$$A_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, b_0, \delta_0, \{b_3\}).$$

$\delta_0$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_0$	$b_1$	$b_3$
$y$	$b_2$	$b_3$	$b_3$	$b_1$

### 5.30. példa - Véges elfogadó automata minimalizálása 4. feladat

Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$A = (\{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\}, \{x, y, z\}, a_0, \delta, \{a_2, a_4, a_5, a_7\})$$

kimenő jel nélküli, iniciális, végállapotokkal bővített

véges automatával ekvivalens  $A_0$  minimális állapotszámú automatát!

$\delta$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
$x$	$a_6$	$a_3$	$a_1$	$a_7$	$a_1$	$a_0$	$a_4$	$a_1$
$y$	$a_4$	$a_7$	$a_0$	$a_2$	$a_3$	$a_1$	$a_5$	$a_6$
$z$	$a_5$	$a_2$	$a_6$	$a_1$	$a_3$	$a_6$	$a_0$	$a_6$

Megoldás:

(O.) Az  $a_0$  kezdőállapotból elérhető belső állapotok:

$$\{a_0, a_6, a_4, a_5, a_1, a_3, a_7, a_2\}.$$

Mivel a kezdőállapotból minden állapot elérhető, ezért nem törölünk egyetlen állapotot sem, tehát  $A' = A$ .

(I.)

$$C_1 = \{a_0, a_1, a_3, a_6\}, \{a_2, a_4, a_5, a_7\}.$$

$$C_2 = \{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_4, a_5, a_7\}.$$

$$C_3 = \{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_5\}, \{a_4, a_7\}.$$

$$C_4 = \{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_5\}, \{a_4, a_7\}.$$

$$b_0 = \{a_0, a_1\}, b_1 = \{a_3, a_6\}, b_2 = \{a_2, a_5\}, b_3 = \{a_4, a_7\}.$$

(II.)

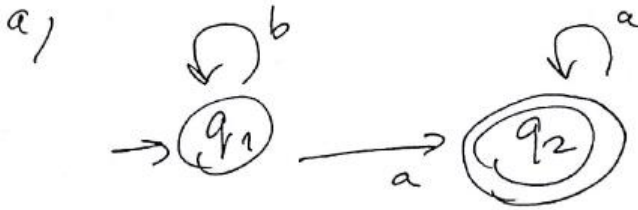
$$A_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y, z\}, b_0, \delta_0, \{b_2, b_3\}).$$



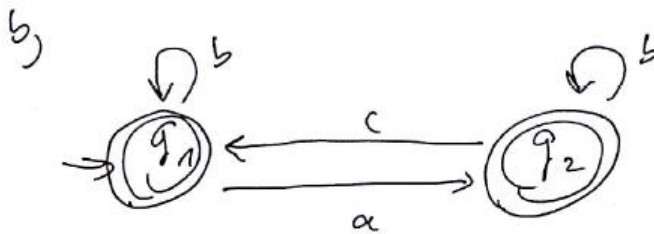
$\delta_0$	$b_0$	$b_1$	$b_2$	$b_3$
$x$	$b_1$	$b_3$	$b_0$	$b_0$
$y$	$b_3$	$b_2$	$b_0$	$b_1$
$z$	$b_2$	$b_0$	$b_1$	$b_1$

## 6. Végtes automata $\rightarrow$ reguláris kifejezés

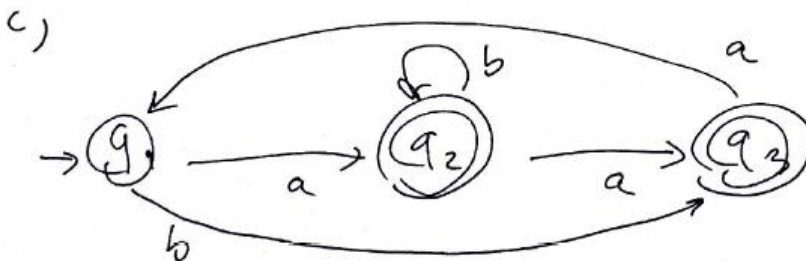
Konstruáljunk reguláris kifejezést az alábbi automaták által elfogadott nyelvekhez.



Reg. kif.:  $b^* a a^*$

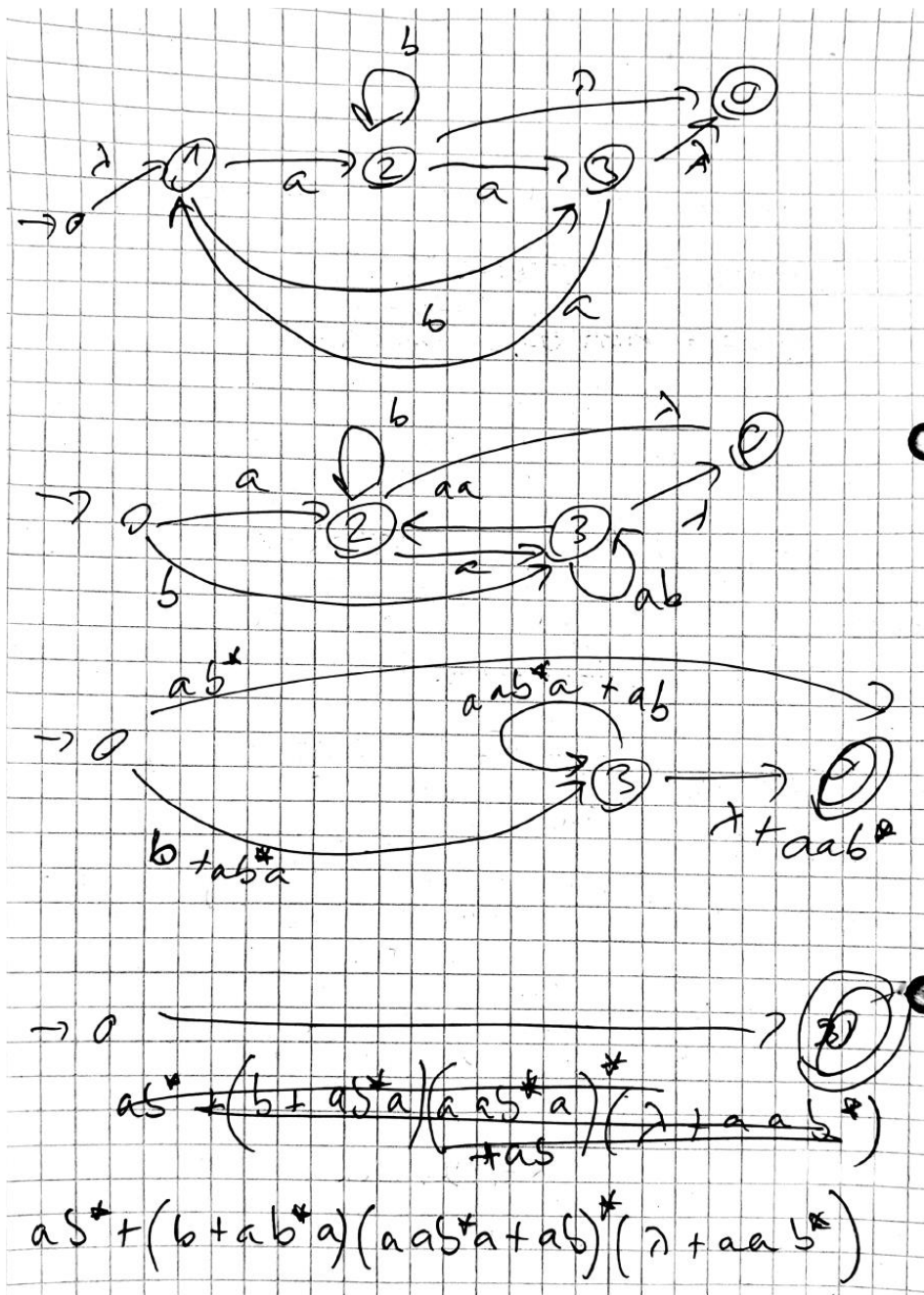


Reg. kif.:  $b^* a (c b^* a + b)^* (c b^* + \lambda) + b^*$

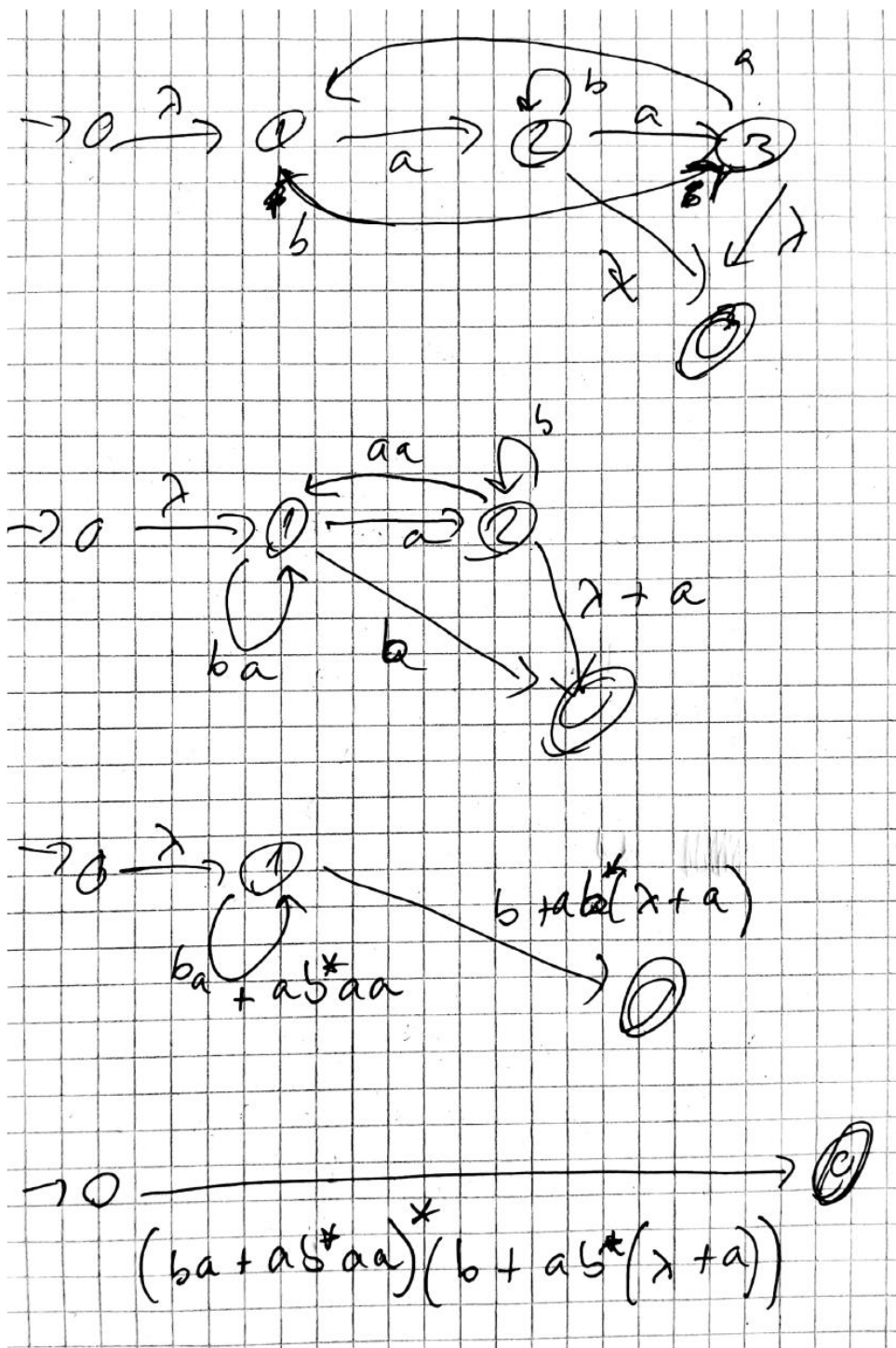


Reg. kif.:  $a b^* + (b + a b^* a) (a a b^* a + a b)^* (\lambda + a a b^*)$

Ha az előző oldalon szereplő 6/c. feladatban q1, q2, q3 sorrendben hagyjuk el az állapotokat, akkor ilyen „átmeneti” általánosított automata rajzokat kapunk:



Ha a 6/c. feladatban q3, q2, q1 sorrendben hagyjuk el az állapotokat, akkor ilyen „átmeneti” általánosított automata rajzokat, és ilyen (az előzővel ekvivalens, de más alakú) reguláris kifejezést kapunk:

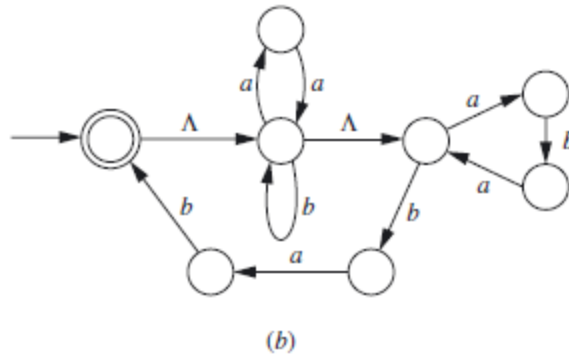
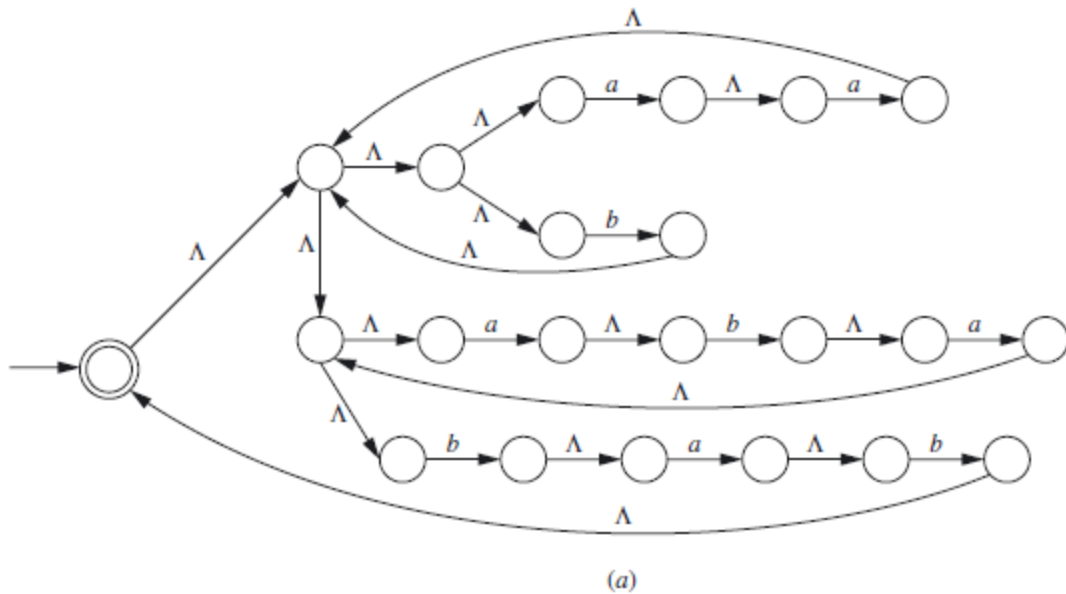




## 7. reguláris kifejezés $\rightarrow$ véges automata

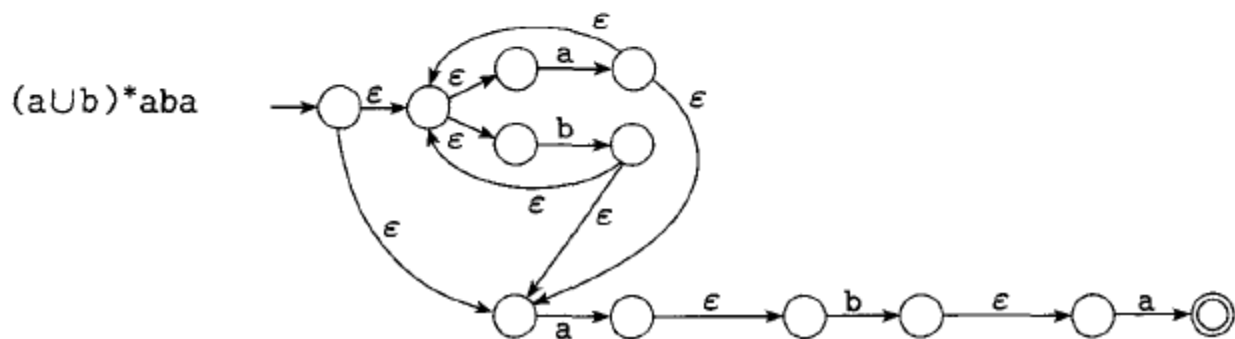
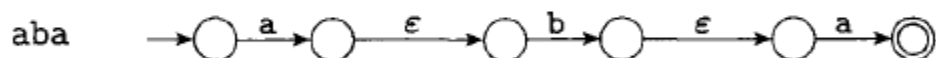
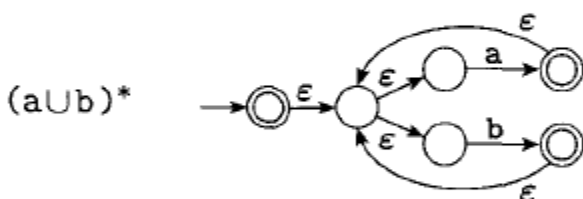
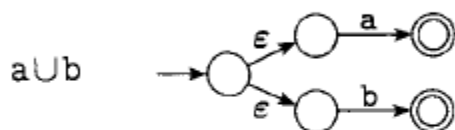
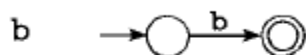
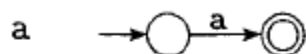
(J. Martin 113. oldal, M. Sipser 69. oldal)

Konstruáljunk véges automatát ami az  $((aa + b)^*(aba)^*bab)^*$  reguláris kifejezés által leírt nyelvet fogadja el.



**EXAMPLE 1.58**

In Figure 1.59, we convert the regular expression  $(a \cup b)^*aba$  to an NFA. A few of the minor steps are not shown.



## 8. Pumpálási lemma reguláris nyelvekre (M. Sipser 81. oldal)

### EXAMPLE 1.75

Let  $F = \{ww \mid w \in \{0,1\}^*\}$ . We show that  $F$  is nonregular, using the pumping lemma.

Assume to the contrary that  $F$  is regular. Let  $p$  be the pumping length given by the pumping lemma. Let  $s$  be the string  $0^p 1 0^p$ . Because  $s$  is a member of  $F$  and  $s$  has length more than  $p$ , the pumping lemma guarantees that  $s$  can be split into three pieces,  $s = xyz$ , satisfying the three conditions of the lemma. We show that this outcome is impossible.

Condition 3 is once again crucial, because without it we could pump  $s$  if we let  $x$  and  $z$  be the empty string. With condition 3 the proof follows because  $y$  must consist only of 0s, so  $xyyz \notin F$ .

Observe that we chose  $s = 0^p 1 0^p$  to be a string that exhibits the “essence” of the nonregularity of  $F$ , as opposed to, say, the string  $0^p 0^p$ . Even though  $0^p 0^p$  is a member of  $F$ , it fails to demonstrate a contradiction because it can be pumped.

✱

---

Magyarázt képpen:

A fenti könyv a következő módon mondja ki a pumpálási lemmát:

**Pumping lemma** If  $A$  is a regular language, then there is a number  $p$  (the pumping length) where, if  $s$  is any string in  $A$  of length at least  $p$ , then  $s$  may be divided into three pieces,  $s = xyz$ , satisfying the following conditions:

1. for each  $i \geq 0$ ,  $xy^i z \in A$ ,
2.  $|y| > 0$ , and
3.  $|xy| \leq p$ .

azaz a „ $p$  pumping length” jelenti azt a számot, aminél hosszabb egy „élég hosszú” szó. A „condition 3” pedig a 3. feltétel, vagyis az, hogy az  $xyz$  felosztás  $xy$  részének hossza kisebb mint  $p$ .

A „ $p$  pumping length” és a „condition 3” a következő magyarázatokban is a fentiekre vonatkozik, „condition 2” pedig azt mondja ki, hogy az  $xyz$  felosztás  $y$  része nem üres.

**EXAMPLE 1.77**

Sometimes “pumping down” is useful when we apply the pumping lemma. We use the pumping lemma to show that  $E = \{0^i 1^j \mid i > j\}$  is not regular. The proof is by contradiction.

Assume that  $E$  is regular. Let  $p$  be the pumping length for  $E$  given by the pumping lemma. Let  $s = 0^{p+1} 1^p$ . Then  $s$  can be split into  $xyz$ , satisfying the conditions of the pumping lemma. By condition 3,  $y$  consists only of 0s. Let's examine the string  $xyyz$  to see whether it can be in  $E$ . Adding an extra copy of  $y$  increases the number of 0s. But,  $E$  contains all strings in  $0^* 1^*$  that have more 0s than 1s, so increasing the number of 0s will still give a string in  $E$ . No contradiction occurs. We need to try something else.

The pumping lemma states that  $xy^i z \in E$  even when  $i = 0$ , so let's consider the string  $xy^0 z = xz$ . Removing string  $y$  decreases the number of 0s in  $s$ . Recall that  $s$  has just one more 0 than 1. Therefore  $xz$  cannot have more 0s than 1s, so it cannot be a member of  $E$ . Thus we obtain a contradiction.  $\square$

**EXAMPLE 1.76**

Here we demonstrate a nonregular unary language. Let  $D = \{1^{n^2} \mid n \geq 0\}$ . In other words,  $D$  contains all strings of 1s whose length is a perfect square. We use the pumping lemma to prove that  $D$  is not regular. The proof is by contradiction.

Assume to the contrary that  $D$  is regular. Let  $p$  be the pumping length given by the pumping lemma. Let  $s$  be the string  $1^{p^2}$ . Because  $s$  is a member of  $D$  and  $s$  has length at least  $p$ , the pumping lemma guarantees that  $s$  can be split into three pieces,  $s = xyz$ , where for any  $i \geq 0$  the string  $xy^i z$  is in  $D$ . As in the preceding examples, we show that this outcome is impossible. Doing so in this case requires a little thought about the sequence of perfect squares:

$$0, 1, 4, 9, 16, 25, 36, 49, \dots$$

Note the growing gap between successive members of this sequence. Large members of this sequence cannot be near each other.

Now consider the two strings  $xyz$  and  $xy^2 z$ . These strings differ from each other by a single repetition of  $y$ , and consequently their lengths differ by the length of  $y$ . By condition 3 of the pumping lemma,  $|xy| \leq p$  and thus  $|y| \leq p$ . We have  $|xyz| = p^2$  and so  $|xy^2 z| \leq p^2 + p$ . But  $p^2 + p < p^2 + 2p + 1 = (p+1)^2$ . Moreover, condition 2 implies that  $y$  is not the empty string and so  $|xy^2 z| > p^2$ . Therefore the length of  $xy^2 z$  lies strictly between the consecutive perfect squares  $p^2$  and  $(p+1)^2$ . Hence this length cannot be a perfect square itself. So we arrive at the contradiction  $xy^2 z \notin D$  and conclude that  $D$  is not regular.  $\square$



## 5.48. példa - Iterációs lemma alkalmazása

Legyen  $L = \{a^m b^m \mid m > 0\}$ . Felhasználva a reguláris pumpáló lemmát, kimutatjuk, hogy ez a nyelv nem reguláris. Tegyük fel hogy az,  $s$  jelöljön  $n$  egy konstanst, mely mellett  $L$  kielégíti a reguláris pumpáló lemma tulajdonságait. Ilyen  $n$ -t viszont nem fogunk találni, hiszen az  $a^n b^n$  szó minden olyan  $uvw$  felbontására, melyre  $|uv| \leq n$  és  $|v| > 0$ , azt kapjuk, hogy  $uw = a^{n-|v|} b^n$ , azaz  $uw \notin L$ . (Hasonlóan, minden  $i > 1$ -re  $uv^i w = a^{n+(i-1)|v|} b^n \notin L$ .) Ezt az ellentmondást csak az indirekt feltevésünk hamissága okozhatja, tehát  $L$  nem reguláris. ★

### The Language $\{x \in \{a, b\}^* \mid n_a(x) > n_b(x)\}$

---

Let  $L$  be the language

$$L = \{x \in \{a, b\}^* \mid n_a(x) > n_b(x)\}$$

The first sentence of a proof using the pumping lemma is always the same: Suppose for the sake of contradiction that there is an FA  $M$  that accepts  $L$  and has  $n$  states. There are more possibilities for  $x$  than in the previous example; we will suggest several choices, all of which satisfy  $|x| \geq n$  but some of which work better than others in the proof.

First we try  $x = b^n a^{2n}$ . Then certainly  $x \in L$  and  $|x| \geq n$ . By the pumping lemma,  $x = uvw$  for some strings  $u$ ,  $v$ , and  $w$  satisfying conditions 1–3. Just as in Example 2.30, it follows from conditions 1 and 2 that  $v = b^k$  for some  $k > 0$ . We can get a contradiction from condition 3 by considering  $uv^i w$ , where  $i$  is large enough that  $n_b(uv^i w) \geq n_a(uv^i w)$ . Since  $|v| \geq 1$ ,  $i = n + 1$  is guaranteed to be large enough. The string  $uv^{n+1}w$  has at least  $n$  more  $b$ 's than  $x$  does, and therefore at least  $2n$   $b$ 's, but it still has exactly  $2n$   $a$ 's.

Suppose that instead of  $b^n a^{2n}$  we choose  $x = a^{2n} b^n$ . This time  $x = uvw$ , where  $v$  is a string of one or more  $a$ 's and  $uv^i w \in L$  for every  $i \geq 0$ . The way to get a contradiction now is to consider  $uv^0 w$ , which has fewer  $a$ 's than  $x$  does. Unfortunately, this produces a contradiction only if  $|v| = n$ . Since we don't know what  $|v|$  is, the proof will not work for this choice of  $x$ .

The problem is not that  $x$  contains  $a$ 's before  $b$ 's; rather, it is that the original numbers of  $a$ 's and  $b$ 's are too far apart to guarantee a contradiction. Getting a contradiction in this just barely satisfied, then ideally any change in the right direction will cause it to fail. A better choice, for example, is  $x = a^{n+1} b^n$ . (If we had used  $x = b^n a^{n+1}$  instead of  $b^n a^{2n}$  for our first choice, we could have used  $i = 2$  instead of  $i = n$  to get a contradiction.)

Letting  $x = (ab)^n a$  is also a bad choice, but for a different reason. We know that  $x = uvw$  for some strings  $u$ ,  $v$ , and  $w$  satisfying conditions 1–3, but now we don't have enough information about the string  $v$ . It might be  $(ab)^k a$  for some  $k$ , so that  $uv^0 w$  produces a contradiction; it might be  $(ba)^k b$ , so that  $uv^2 w$  produces a contradiction; or it might be either  $(ab)^k$  or  $(ba)^k$ , so that changing the number of copies of  $v$  doesn't change the relationship between  $n_a$  and  $n_b$  and doesn't give us a contradiction.

## The Language $L = \{a^{i^2} \mid i \geq 0\}$

---

Whether a string of  $a$ 's is an element of  $L$  depends only on its length; in this sense, our proof will be more about numbers than about strings.

Suppose  $L$  can be accepted by an FA  $M$  with  $n$  states. Let us choose  $x$  to be the string  $a^{n^2}$ . Then according to the pumping lemma,  $x = uvw$  for some strings  $u$ ,  $v$ , and  $w$  satisfying conditions 1–3. Conditions 1 and 2 tell us that  $0 < |v| \leq n$ . Therefore,

$$n^2 = |uvw| < |uv^2w| = n^2 + |v| \leq n^2 + n < n^2 + 2n + 1 = (n + 1)^2$$

This is a contradiction, because condition 3 says that  $|uv^2w|$  must be  $i^2$  for some integer  $i$ , but there is no integer  $i$  whose square is strictly between  $n^2$  and  $(n + 1)^2$ .